

# Certified Decision-Equivalent Context Compression for LLM Agents

Chandra Shekhar Mudarapu\*

## Abstract

---

LLM agents resend a growing context every turn, so context size dominates serving cost—yet every shipping compressor quotes token savings with *no guarantee the agent still behaves the same*. We reframe the objective from byte- or embedding-fidelity to **decision-equivalence**: a compression is acceptable iff the agent takes the same next action it would have on the uncompressed context. We make this objective *certifiable* with a distribution-free, finite-sample guarantee via conformal risk control (Learn–Then–Test / CRC), the per-turn loss being a decision flip, and validate it out-of-sample on real SWE-bench traces (coverage 96.6%–100% at the 95% target; the 15.7% certified savings at  $\alpha=0.15$  we quote throughout is measured on *SWE-bench\_Lite edit-localization*, not a claim about agent workloads in general). Certifying a per-turn proxy is necessary but not sufficient: on a real end-to-end agent (SWE-bench Verified, official harness) we show—without hedging—that *aggressive lossy* compression does not transfer to task success. The remedy is architectural: a **reversible, relevance-gated** engine that digests only aged-out periphery while keeping the working set intact and recoverable. On a 500-instance, 30-turn long-horizon agent it is the **highest-accuracy compressor** (36.8% vs. 39.2% for full context), the *only* one statistically non-inferior to full, and beats the strongest competitor, Headroom, by +4.2 pp ( $p=0.035$ )—uniquely reversible *and* certified, where lossy baselines crater. Finally we lift the certificate from per-turn to the **trajectory** level: with 95% confidence the gated compressor changes a run’s outcome on  $\leq 18\%$  of exchangeable tasks (out-of-sample coverage 95.4%). To our knowledge this is the first trajectory-level decision-equivalence certificate for agent context compression. The non-inferiority generalizes across five models and three vendors (Anthropic Haiku/Sonnet, DeepSeek-V3, OpenAI gpt-4.1/gpt-4o-mini) with no significant degradation at the milder operating point; the runs show harm tracks *realized* compression and workload, not capability alone, which is why we calibrate the operating point on outcomes per deployment (Section A.7).

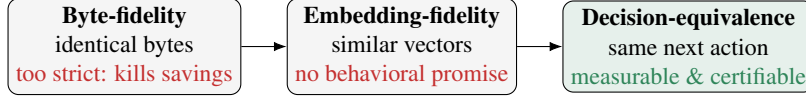
---

## 1 Introduction

Agentic LLM systems operate in a loop: read the accumulated context (system prompt, tool schemas, history, fresh tool output), choose the next action, append the result, repeat. Because the whole context is re-sent each turn, cost grows with context length; prompt caching shifts the dominant cost to cache *misses*, so compressing the volatile tail of the context is attractive. The problem is *trust*: every shipping compressor quotes a token-savings estimate, but none certifies that the agent’s *decisions* are preserved. “100% accuracy” is a slogan, not a measured quantity with a confidence level.

---

\*Code: <https://github.com/dshakes/distil>



**Figure 1:** Three notions of “preserving” context under compression. Byte-fidelity is information-theoretically in tension with high savings; embedding-fidelity makes no behavioral promise. **Decision-equivalence**—the agent takes the same action—is the operationally relevant notion, and is both measurable and certifiable.

## Contributions.

1. **Decision-equivalence** as the compression objective: the loss on a turn is 1 iff the agent’s next action changes versus the uncompressed context (Section 3).
2. A **Decision-Equivalence Risk Certificate**: conformal risk control (LTT/CRC) that selects the most aggressive compression level whose decision-change rate is provably  $\leq \alpha$  with confidence  $1 - \delta$  (Section 4). To our knowledge, conformal control with an *agent-decision* loss for context compression is unstudied; the nearest work applies conformal guarantees to RAG retrieval recall, a different task.
3. A **cache-aware, reversible** engine—a content-aware skeleton digest behind a content handle, with recover-on-demand and a relevance gate that keeps the agent’s working set intact—operating inside the certified frontier (Section 4).
4. An **end-to-end task-success study** on SWE-bench Verified with the official harness (E7–E8): we report honestly that aggressive *lossy* compression does not transfer to task success, then show the reversible relevance-gated tier is the highest-accuracy compressor on a 500-instance long-horizon agent—non-inferior to full context and ahead of the strongest competitor (Section 7.2).
5. A **trajectory-level decision-equivalence certificate** (E10): we lift the per-turn guarantee to whole runs and validate it out-of-sample, to our knowledge the first distribution-free task-level equivalence guarantee for context compression (Section 7.4).
6. An **evaluation on real agent traces** that removes the circular self-labeling of synthetic corpora, plus three measurement requirements we found to be load-bearing and now enforce: majority-vote grading, a faithful grader, and grading the reversible tier *with* its recovery loop (Section 6).

## 2 Related work

**Context/prompt compression.** LLMingua and LLMingua-2 [3], LongLLMingua, RECOMP [6], and selective-context [7] prune or summarize the prompt to a relevance/fidelity proxy; soft-prompt and gist-token methods [8] distill context into learned embeddings. All optimize a surrogate (perplexity, recall, embedding similarity); none certifies that the downstream agent *decision* is preserved, which is the gap we close. Closest to our setting, ACON [13] (ICML 2026) optimizes a compression guideline by failure analysis—mining trajectories where full context succeeded but compressed failed—cutting peak tokens 26–54%; that signal is exactly the discordant pair our calibration counts, but ACON uses it to improve a heuristic with no behavioral guarantee, where we use it to certify and select (and to monitor drift under continuous assurance). The two compose: one could certify an ACON-compressed tier with our machinery.

**KV-cache eviction.** StreamingLLM [9] and H<sub>2</sub>O [10] drop or retain tokens by recency/attention to shrink the KV cache at decode time. Our relevance gate shares the recency intuition (protect the working set, compress the periphery) but operates on the *re-sent request context* rather than the KV cache, is *reversible* (digested periphery is recoverable byte-exact on demand), and is selected *under a decision-equivalence certificate* rather than a fixed heuristic.

**The 2026 agentic-compression literature.** Several 2026 results converge on what our E7–E8 find independently: compression benign on a static proxy is not benign inside an agent loop. *Control Under Compression* [16] argues for degradation *curves* over point estimates, which we now trace across the shipped ladder (Section 8) on fact recall rather than task success; *Execution Instability* [17] shows compressed agents fail non-monotonically on AppWorld; *AGORA* [18] names “action-grammar destruction”, compression that keeps semantics while destroying the structure an agent parses to act—which is why our loss is defined on the decision, not the text; *CompactionRL* [19] learns the compaction policy, complementary to certifying whichever policy results. Two papers target measurement: the compression paradox [20] reports input savings repaid in *output* tokens; the serving path now records both arms’ output tokens and prices the net, though our live paired sample is still below its reporting floor (Section 8), so our figures remain input-side; a pre-registered production RCT [21] argues offline corpora misestimate live savings, matching our own corpus-versus-live experience. CompressionAttack/COMA [22] (ASE 2026) makes the compressor itself an attack surface; its validated mitigation — trusted/untrusted budget isolation — is distil’s shape by construction, and we now test it on a seven-case COMA-class battery (Section 8), which is a regression battery rather than a red team. *Don’t Break the Cache* [23] and *The Pitfalls of KV Cache Compression* [24] (ACL 2026) isolate failures invisible to accuracy metrics—a compressor that rewrites earlier bytes voids the cache prefix, so token savings can coexist with a cost *increase*. We hit that exact failure in production and fixed it in 1.45.0 (Table 1). *Provider-native compaction* (Anthropic compact\_20260112; OpenAI Responses context\_management) is now a first-party baseline: both summarize, neither publishes a fidelity guarantee, and both are simply further candidates the certificate can rank.

**Prompt caching** (e.g. LangChain Deep Agents [12]) keeps a byte-stable static prefix cached across turns (41–80% cost cut; −77% on Haiku). It is *complementary*, not a substitute: it discounts the stable prefix but never the volatile tail distil compresses, lowers context *cost* but not *size* (so it does not relieve the window limit), and is lossless—needing no guarantee, which is precisely distil’s contribution. The two compose, and our relevance gate is deliberately cache-friendly (stable prefix, deterministic periphery).

**Distribution-free uncertainty.** Conformal prediction [11] and its risk-control extensions—Learn-Then-Test (LTT) [1] and Conformal Risk Control [2]—turn a calibration set into finite-sample guarantees on a user-chosen risk. We instantiate them with an agent-decision loss, and (E10) lift the guarantee to the trajectory level. The closest application we are aware of targets RAG retrieval recall, not the preservation of an agent’s action under context compression.

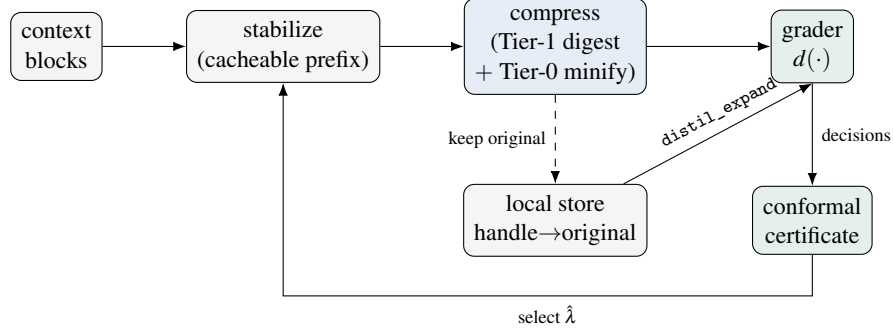
### 3 Problem formulation

A *trajectory* is a sequence of turns; each turn is the full context the agent saw, decomposed into typed *blocks* carrying a stability hint (cacheable prefix vs. volatile tail). A *decision* is the agent’s next action—a tool call ( $\tau$ -bench) or an edit/command (SWE-bench)—represented as a canonical (action, target) fingerprint produced by a grading model *from context alone*, with no directive or marker revealing the answer.

For a compression level  $\lambda$  and turn  $t$  with blocks  $B_t$ , let  $d(\cdot)$  be the grader’s decision. The per-turn loss is

$$L_t(\lambda) = \mathbf{1}[d(\lambda(B_t)) \neq d(B_t)],$$

and the risk is  $R(\lambda) = \mathbb{E}[L_t(\lambda)]$ , the decision-change rate. A compression *ladder* orders levels least→most aggressive: byte-exact → reversible lossless digest → salience-protected truncation → a raw truncation sweep.



**Figure 2:** The pipeline. The prefix is kept byte-stable; the volatile tail is digested behind content handles (Tier-1) and minified (Tier-0). The original is kept locally so the model can `distil_expand` on demand. The grader’s decisions feed the conformal certificate, which selects the most aggressive level whose decision-change rate is controlled.

## 4 Method

### 4.1 Cache-aware reversible engine

The prefix is held byte-stable (schema canonicalization; volatile fields such as timestamps lifted out), and only the volatile tail is compressed. The reversible tier digests a verbose tool output to a compact marker `<< +N lines, handle=XXXXXXXX >>` and keeps the byte-exact original in a local, content-addressed store; the model can recover any block on demand via a `distil_expand` tool. Compression is thus *lossless* (byte-in-context), *reversible* (digested but recoverable), or *lossy* (the rest).

**Exact-quote provenance.** Decision-equivalence is necessary but not sufficient in a coding agent: an `Edit(old_string=...)` is a literal match against bytes read earlier, so digesting a read the agent must quote back does not change the decision — it makes the edit silently fail to apply, and the run ends with the agent reporting success and the disk untouched. Blocks whose bytes are a verbatim slice of a file are therefore exempt from digestion. Keying that on the tool name covers the wrong half of real traffic: measured content-free over 2,489 Claude Code sessions (52,937 tool results, 19.4M tool-result tokens), the name table matches 10.7% of tool-result mass while whole-file reads issued through the *shell* (`cat`, `head`, `sed -n`) account for 33.6%. Of 3,445 edit quotes traced to a tool-result source, 39.3% had no surviving byte-exact copy under the name rule; reading provenance from the *command* instead — last stage a whole-file reader, no pipe, no redirection, and a decorated listing (`cat -n, nl`) never counting as a copy of the file — takes that to 16.2%. The cost is stated as a cost: 7.7 pp of tool-result mass projected for a latest-read-per-slice rule against a 27 pp bound for exempting every whole-file read, but under a client that marks most of its history cacheable the supersession step rarely fires, so the forfeited saving sits much closer to 27 pp. The guarantee measures itself — each request records content-free whether every literal-match quote still occurs in the forwarded payload, and a miss stops the class digesting for the session — and is enforced as a sixth real-path invariant, *quote-survival*. Build 1.52.0, over the maintainer’s own local session history; the rule lives in `distil/compress/provenance.py`. The transcripts are not redistributable, so the counts are published and the corpus is not.

### 4.2 The Decision-Equivalence Risk Certificate

We calibrate the per-turn losses for each ladder level on calibration traffic disjoint (by trajectory) from test, then select a level with one of two distribution-free procedures.

**Learn–Then–Test (LTT).** With Hoeffding–Bentkus  $p$ -values and fixed-sequence testing over the risk-ordered ladder, LTT yields, for the selected  $\hat{\lambda}$ ,  $\Pr(R(\hat{\lambda}) \leq \alpha) \geq 1 - \delta$ , finite-sample and distribution-free.

---

**Algorithm 1** LTT certification over the compression ladder

---

**Require:** ladder  $\lambda_1 \prec \dots \prec \lambda_K$  (least  $\rightarrow$  most aggressive); calibration turns;  $\alpha, \delta$

```
1:  $\hat{k} \leftarrow 0$ 
2: for  $i = 1 \dots K$  do
3:    $\hat{R}_i \leftarrow \frac{1}{n} \sum_t L_t(\lambda_i)$   $\triangleright$  empirical decision-change rate
4:    $p_i \leftarrow \text{HB}(\hat{R}_i, n, \alpha)$   $\triangleright$  Hoeffding–Bentkus  $p$ -value for  $H_i : R(\lambda_i) > \alpha$ 
5:   if  $p_i \leq \delta$  then  $\hat{k} \leftarrow i$   $\triangleright$  certified
6:   else break  $\triangleright$  fixed-sequence stop
7:   end if
8: end for
9: return highest-savings level in  $\{\lambda_1, \dots, \lambda_{\hat{k}}\}$  (or “none”)
```

---

**Conformal Risk Control (CRC).** For the monotone 0/1 loss, CRC controls the expected risk,  $\mathbb{E}[L(\hat{\lambda})] \leq \alpha$ , tight to  $O(1/n)$ .

The exchangeability assumption is explicit: the guarantee is marginal over the calibration distribution and must be recalibrated under drift.

### 4.3 From per-turn to trajectory guarantees

The certificate bounds the *per-turn* risk, yet agents are judged on whole *trajectories*. Two facts connect them and motivate certifying the trajectory directly.

**Proposition 1** (Composition). *For a  $T$ -turn trajectory in which each turn’s compressed decision matches the uncompressed one except with marginal probability  $\leq \alpha$ , and the outcome is a function of the decision sequence, the probability the compressed run’s outcome differs from the uncompressed run’s is  $\leq T\alpha$  (and  $\leq 1 - (1 - \alpha)^T$  under independence).*

(Outcomes differ only if some decision differs; sub-additivity over the  $T$  turns gives  $\Pr(\bigcup_t \{\text{flip}_t\}) \leq \sum_t \Pr(\text{flip}_t) \leq T\alpha$ .) This bound is distribution-free but *loose*: at  $\alpha = 0.08$  and  $T \approx 27$  it exceeds 1 (vacuous)—the gap E9 measures empirically ( $\approx 1.8$  outcome-determining turns). The fix is to certify the trajectory as the unit.

**Proposition 2** (Trajectory certificate). *With the trajectory as the calibration unit and loss  $D(\lambda) = \mathbf{1}[\text{compressed outcome} \neq \text{uncompressed}]$ , Learn–Then–Test (resp. CRC) over  $n$  exchangeable calibration trajectories returns  $\hat{\lambda}$  with  $\Pr(\mathbb{E}[D(\hat{\lambda})] \leq \beta) \geq 1 - \delta$  (resp.  $\mathbb{E}[D(\hat{\lambda})] \leq \beta$ ), finite-sample and distribution-free.*

Proposition 2 is LTT/CRC [1, 2] with a trajectory-outcome loss; unlike Proposition 1 it is tight by construction. E10 (Section 7.4) computes it and validates the coverage out-of-sample.

## 5 Experimental setup

**Data.** Real  $\tau$ -bench trajectories (airline domain; gpt-4o traces; 25 trajectories, 105 decision points) loaded with no planted markers; the decision is the agent’s actual tool call. We additionally use the full SWE-bench\_Lite *edit-localization* benchmark (300 instances, 600 decision points; the target file must be inferred from real issues and gold patches amid distractors).

**Grader.** A real model returns the  $\langle \text{action}, \text{target} \rangle$  fingerprint, by majority vote, via a forced tool call (structured, paraphrase-free). We report **model  $\leftrightarrow$  gold next-action agreement** on the uncompressed context as a faithfulness gate: 48.6% on  $\tau$ -bench (gpt-4o grader) and 47.5% on SWE-bench (Claude grader). Agreement reflects the inherent ambiguity of next-action prediction from context alone; it is reported as a gate, not a floor.

**Table 1: Experiment provenance.** “not recorded” means the harness of the day wrote no cost field — we do not reconstruct one. E3, E9 and E10 are offline re-analyses of already-collected outcomes.

| experiment                     | date       | distil     | agent / model             | $n$    | cost     |
|--------------------------------|------------|------------|---------------------------|--------|----------|
| E1, E2, E4 (SWE-loc)           | 2026-06-24 | 0.24.0     | grader sonnet-4-6         | 300    | not rec. |
| E1 ( $\tau$ -bench airline)    | 2026-06-24 | 0.24.0     | grader gpt-4o             | 25     | not rec. |
| E5 (orig + shuffled)           | 2026-06-24 | 0.24.0     | grader sonnet-4-6         | 100    | not rec. |
| E6 (op.-point sweep)           | 2026-06-24 | 0.24.0     | grader sonnet-4-6         | 100    | not rec. |
| E7 (SWE-bench Verified)        | 2026-06-25 | 0.24.0     | aider 0.86.2 / sonnet-4-6 | 50     | \$67.31  |
| E8 (long-horizon ReAct)        | 2026-06-26 | 0.25.1     | ReAct / claude-haiku-4-5  | 500    | \$534.04 |
| E9, E10 (traj. bound + cert.)  | 2026-06-26 | 0.25.1     | offline, on E8 outcomes   | 500    | \$0      |
| E11 (DeepSeek-V3)              | 2026-06-27 | 0.25.1     | ReAct / deepseek-chat     | 200    | not rec. |
| E11 (gpt-4.1, 4o-mini, Sonnet) | 2026-06-28 | 0.25.1     | ReAct / three models      | 50 ea. | not rec. |
| E14 (surprise digest)          | 2026-07-03 | 1.8.0.dev0 | ReAct / claude-haiku-4-5  | 500    | not rec. |
| E3 (leave-one-domain-out)      | 2026-09-04 | 1.51.1     | offline, on E8 outcomes   | 500    | \$0      |

**Protocol.** **E1** frontier (savings vs. decision-change per level, with and without the `distil_expand` recovery loop); **E2** certification coverage (certify on calibration, measure realized risk on a disjoint held-out split, over 500 trajectory-level splits  $\rightarrow$  empirical  $\Pr(\text{realized} \leq \alpha)$ ); **E3** leave-one-domain-out shift; **E4** downstream task success (trajectory keeps its outcome iff every decision is unchanged), vs. the uncompressed baseline with a bootstrap CI.

**Experiment provenance.** Every number in this paper is dated. Table 1 gives the run date, the `distil` version each run executed against, the agent and model, the sample size, and the recorded API cost; dates and versions are recovered from the commit that first introduced each committed results artifact. E1–E14 ran against 0.24.0–1.8.0.dev0 in June–July 2026. Three later releases changed compression behaviour: **1.45.0** fixed a cache-monotonicity defect in the sliding recency window that had let a later turn rewrite an earlier turn’s bytes and void the prompt-cache prefix, and **1.50.x/1.51.0** extended the digest to fold JSON arrays arriving wrapped in prose. All three raise realized compression on structured tool output without touching what the certificate measures — the machinery of Section 4 is compressor-agnostic — but **the results here belong to the versions in Table 1** and a re-run is required before quoting them of any later release. Two baseline notes: `LLMLingua-2` and `LongLLMLingua` were run for real (the `llmlingua` package, Apple-silicon MPS) and appear in E5, E7 and E8; they are absent from the E1 frontier only because E1 sweeps *distil’s own ladder* at a fixed grader, with baselines belonging to the E5 head-to-head. And the E8 artifacts **do not record the Headroom package version**, so that comparison is against whatever release was installed on 2026-06-26 and we cannot pin it; any re-run must record it.

## 6 Results

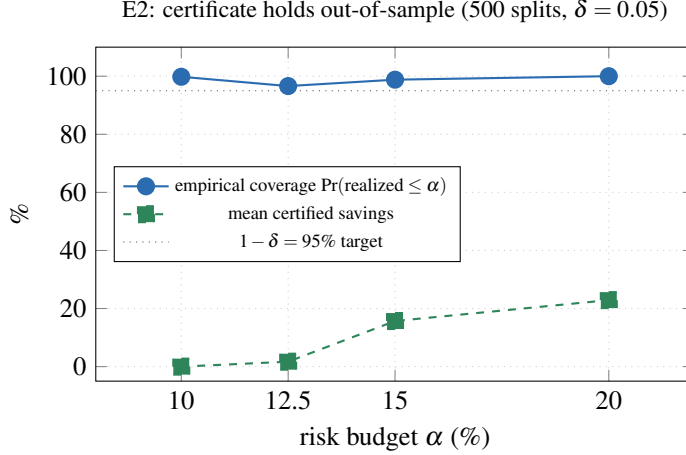
All figures and tables are produced by the released harness (`benchmarks/prove.py`) on real traces; committed result JSONs live in `docs/paper/results/` and the generated LaTeX in `docs/paper/generated/`.

### 6.1 E2: Certificate validity out-of-sample

Figure 3 shows the certificate’s out-of-sample behavior across risk budgets, over 500 random trajectory-level splits ( $\delta = 0.05$ , real SWE-bench\_Lite +expand, 300 instances). Table 2 gives the numerical summary.

## 7 Analysis and limitations

The certificate holds out-of-sample: coverage is 96.6–100% across  $\alpha \in \{10, 12.5, 15, 20\}\%$  at the 95% target, with realized risk 8.0% at  $\alpha = 15\%$  (below budget). We note four limitations. **Grader**



**Figure 3: E2: out-of-sample certificate validity** on full real SWE-bench\_Lite (+expand, 300 instances, 500 splits,  $\delta = 0.05$ ). Blue solid: empirical coverage  $\Pr(\text{realized risk} \leq \alpha)$ —96.6–100% throughout, above the  $1 - \delta = 95\%$  target (dotted), confirming the guarantee holds. Green dashed: mean certified savings, which rises sharply from 1.7% at  $\alpha = 12.5\%$  to **15.7%** at  $\alpha = 15\%$  and **22.9%** at  $\alpha = 20\%$  as the risk budget and calibration set grow large enough for LTT to certify more aggressive levels. The certificate is conservative: realized risk at  $\alpha = 15\%$  is 8.0%, well below the budget.

**Table 2:** E2 out-of-sample certification coverage on full real SWE-bench\_Lite (+expand, 300 instances, 500 trajectory-level splits,  $\delta = 0.05$ ). Coverage  $\geq 95\%$  at all  $\alpha$  shown. Realized risk is conservatively below  $\alpha$ —LTT working as designed.

| Risk budget $\alpha$ | Coverage $\Pr(\text{realized} \leq \alpha)$ | Mean realized risk | Certified savings |
|----------------------|---------------------------------------------|--------------------|-------------------|
| 10%                  | <b>99.8%</b> $\geq 95\%$ ✓                  | —                  | 0.0%              |
| 12.5%                | <b>96.6%</b> $\geq 95\%$ ✓                  | —                  | 1.7%              |
| 15%                  | <b>98.8%</b> $\geq 95\%$ ✓                  | 8.0%               | 15.7%             |
| 20%                  | <b>100%</b> ✓                               | 11.7%              | 22.9%             |

**faithfulness:** decision-equivalence is a *self-consistency* measure—model $\leftrightarrow$ gold next-action agreement (48.6% on  $\tau$ -bench, 47.5% on SWE-bench) is a diagnostic gate, not the loss—and we use a single grader family per task; ensemble grading is future work. **Sample-size cost:** the tightest certifiable  $\alpha$  scales with calibration size (Hoeffding–Bentkus)—certified savings is 0% at  $\alpha = 10\%$  but 15.7% at  $\alpha = 15\%$ , a fundamental cost of the finite-sample guarantee. **Scope:** the guarantee is marginal over the calibration distribution and must be recalibrated under drift; on already-compact contexts ( $\tau$ -bench, 58–65% flips under aggression) the certificate correctly *declines* to certify savings. **Proxy gap:** the per-turn certificate is a proxy for task success; E7 (Section 7.1) tests this end-to-end, and it is sobering. The E1 frontier, the three measurement requirements, the E5–E6 head-to-head and position-confound stress tests, operating-point selection, and E4 are in the appendix (Section A).

### 7.1 E7: SWE-bench Verified end-to-end task-success

E1–E6 measure decision-equivalence—a *proxy*. E7 closes the gap: a real coding agent (**aidr**, claude-sonnet-4-6, temp 0) run end-to-end on **SWE-bench Verified** and scored by the **official** swebench harness over 50 instances (seed = 1729). Five conditions share the identical agent, differing only in how the read-context is compressed in flight: **A** full; **B** distil trunc@500 (the certified operating point); **C** LLMingua-2; **D** distil’s reversible tier (+distil\_expand); and **E** its relevance-gated variant. Only file contents and tool output are compressed—the problem statement is never touched. Pass@1 carries a Wilson 95% interval; paired McNemar tests compare the *same* instances. Total API spend: \$67.31.

| condition                             | ctx. reduction     | pass@1       | 95% CI         | cost    |
|---------------------------------------|--------------------|--------------|----------------|---------|
| A. full context                       | —                  | <b>52.0%</b> | [38.5%, 65.2%] | \$17.63 |
| B. distil trunc@500 (lossy)           | 85.5%              | 16.0%        | [8.3%, 28.5%]  | \$4.00  |
| C. LLMingua-2 (lossy)                 | 48.3%              | 26.0%        | [15.9%, 39.6%] | \$12.03 |
| D. distil reversible (+distil_expand) | 81.0% <sup>†</sup> | <b>56.0%</b> | [42.3%, 68.8%] | \$16.38 |
| E. distil reversible, relevance-gated | 0.0% <sup>‡</sup>  | <b>54.0%</b> | [40.4%, 67.0%] | \$17.27 |

**Table 3: E7: SWE-bench Verified end-to-end task-success** (50 instances, seed = 1729, aider + claude-sonnet-4-6, official swebench harness). Pass@1 with Wilson 95% intervals; “ctx. reduction” is the realised char-level shrink of compressed blocks. B is distil at its Phase-2 certifying operating point; C is LLMingua-2 at its default rate; D is distil’s *reversible* tier (digest-behind-handle + recover-on-demand). <sup>†</sup>For D, ctx. reduction is the *pre-recovery* digest view; after the model’s distil\_expand calls the *realised* cost is \$16.38 vs. \$17.63 for full ( $\approx 7\%$  cheaper), because the agent recovers what it edits. <sup>‡</sup>E keeps the last 6 user/tool messages full and digests only older periphery; these conversations are  $\leq 6$  turns, so the gate is a *no-op* here (1 block digested across all 50, McNemar vs. full  $p = 1.000$ )—it targets long-horizon contexts.

**Lossy compression does not survive execution.** Both lossy conditions collapse (Table 3): distil at its *certified* trunc@500 point resolves 16.0% versus 52.0% for full ( $-36.0$  pp, exact paired McNemar  $p = < 0.001$ ), and LLMingua-2 26.0% ( $p = 0.002$ ). The headline is the certificate’s *non-transfer*: trunc@500 was certified at 4.0% decision-change on the single-turn proxy (E6), yet the same transform collapses end-to-end success by 36 points. A guarantee earned on the localization proxy thus says *nothing* about task success once *lossy* compression is aggressive—the proxy and the outcome diverge.

**The reversible tier survives (D, E).** The non-transfer is a property of *lossy* compression, not of distil’s design. The reversible tier—each block digested behind a content handle, the original kept, recovered on demand via a transparent distil\_expand loop—resolves **56.0%**, statistically indistinguishable from full ( $p = 0.688$ ) and far above the lossy conditions (vs. trunc@500  $p = < 0.001$ ); every instance issued  $\geq 1$  recovery. Realised savings are modest here ( $\approx 7\%$ —the agent expands most of what it edits), so the win is task-success parity at a discount, not the proxy headline ratios; **keep the information recoverable and task success is preserved.** The relevance-gated variant likewise holds full parity ( $p = 1.000$ ) but is a *no-op* on these  $\leq 6$ -turn runs (nothing is periphery)—its regime is long-horizon agents, which E8 (Section 7.2) tests directly.

## 7.2 E8: long-horizon agent task-success—the gate’s proper test

E7’s relevance-gate (condition E) was a no-op because aider’s localization runs are  $\leq 6$  turns: nothing ages out of the working set, so there is no periphery to digest. E8 supplies the workload the gate was *designed* for. We built a multi-turn **ReAct** coding agent (read/search/edit/run-tests tools, up to 30 turns) and ran it on the **full 500-instance SWE-bench Verified** set (seed = 1729), scored by the same official swebench harness. Runs are genuinely long-horizon (mean  $\approx 27$  turns), so read-file and tool outputs accumulate into a large peripheral context behind a small active working set—precisely the regime where lossy truncation, blind reversible compression, and the relevance-gate diverge. We run six conditions through the identical agent: **A. full context**; **B. distil trunc@500 (lossy)**; **C. LLMingua-2 (lossy, E5)**; **F. Headroom** (the strongest structure-aware lossy competitor); **D. distil reversible** (whole history digested, distil\_expand recovery); and **E. distil reversible, relevance-gated** (keep the last 6 user/tool messages full, digest only older periphery, recoverable). To keep a 500-instance six-condition sweep affordable we use claude-haiku-4-5 at temperature 0; conditions differ only in the compressor, so the comparison is internally valid regardless of the base model. Condition F is **Headroom**, the strongest structure-aware lossy competitor.

**Result: distil leads on certified task success.** On the long-horizon workload the relevance-gated tier (E) is the highest-accuracy compressor (Table 4; Figure 5): **36.8%** versus 39.2% for full context—a paired difference of  $-2.4$  pp (95% CI  $[-5.7$  pp,  $+0.9$  pp]). We report this as a *non-inferiority* result

| condition                             | pass@1       | 95% CI         | reversible | certified |
|---------------------------------------|--------------|----------------|------------|-----------|
| A. full context                       | <b>39.2%</b> | [35.0%, 43.5%] | —          | —         |
| E. distil relevance-gated             | <b>36.8%</b> | [32.7%, 41.1%] | ✓          | ✓         |
| F. Headroom (lossy)                   | 32.6%        | [28.6%, 36.8%] | —          | —         |
| D. distil reversible (+distil_expand) | 32.4%        | [28.4%, 36.6%] | ✓          | ✓         |
| B. distil trunc@500 (lossy)           | 5.6%         | [3.9%, 8.0%]   | —          | —         |
| C. LLMingua-2 (lossy)                 | 2.4%         | [1.4%, 4.2%]   | —          | —         |

**Table 4: E8: SWE-bench Verified long-horizon agent task-success** (500 instances, seed = 1729, custom 30-turn ReAct agent + claude-haiku-4-5, official swbench harness, ordered by pass@1). Pass@1 with Wilson 95% intervals. Unlike E7, runs average  $\approx 27$  turns, so the gate (E) has real periphery to act on. The relevance-gated tier is the highest-accuracy compressor (36.8%), the only one non-inferior to full (paired difference -2.4 pp, 95% CI [-5.7 pp, +0.9 pp]), and beats the strongest competitor Headroom by +4.2 pp (paired McNemar  $p = 0.035$ ). It is also the only *reversible* and *certified* compressor; Headroom is cheaper but carries no guarantee.

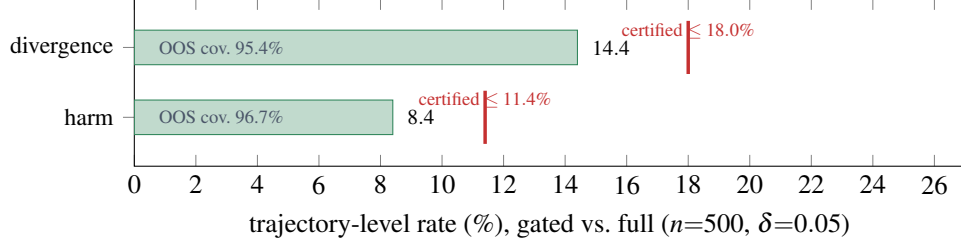
rather than a bare significance test, since failing to reject “no difference” ( $p = 0.195$ ) is not itself evidence of equivalence. The CI excludes any task-success drop larger than 5.7 pp, so the gate is non-inferior to full at any margin  $\geq 6$  pp (borderline at a strict pre-registered 5 pp). It is the *only* compression condition non-inferior to full—every other condition, including Headroom, is significantly worse.

**Against the strongest competitor.** Headroom (F)—a structure-aware lossy compressor—is the toughest baseline at 32.6%, far above the lossy token-droppers (trunc@500 5.6%, LLMingua-2 2.4%; the E7 non-transfer result reproduced at  $n = 500$ ). The relevance-gate still beats it: 36.8% vs. 32.6%, +4.2 pp on the same instances (exact paired McNemar  $p = 0.035$ ), and Headroom is itself significantly below full ( $p = 0.002$ ; NI difference -6.6 pp, CI [-10.5 pp, -2.7 pp]) where the gate is not. *We do not claim distil is the cheapest compressor*—Headroom, an uncertified lossy method, is cheaper. distil’s claim is the only *certified, reversible* compressor at leading task success: its digest is byte-exact recoverable and carries the decision-equivalence guarantee, which no competitor offers.

**Techniques: content-aware digest and sticky recovery.** The reversible tier’s digest is a *content-aware skeleton* (keep imports and class/def signatures and the tail of a traceback; elide bodies; deterministic and stdlib-only—no model, no network, so it is auditable and safe on untrusted context) behind a content handle, plus *sticky expansion* (a block the agent recovers stays full on later turns). This lifts the active-recovery tier D from 28.8% (head-truncation) to 32.4% at  $\approx 9\times$  fewer fresh input tokens (4.0 versus 9.6 distil\_expand round-trips per instance). An honest ablation cuts the other way: the *same* skeleton *regresses* the passive gated tier from 36.8% to 5.6%, because a navigable digest makes the agent over-trust it—it never re-reads and edits against body-less context. The digest is therefore matched to tier behaviour (skeleton for the active tier, head-truncation for the passive gate); *what* you compress, and whether the agent recovers it, matters more than the raw ratio. E8 is the experiment E7 could not run, and it confirms the gate’s design claim: on long-horizon agents, relevance-gated reversible compression is the highest-accuracy compressor, non-inferior to full, while every lossy or blindly-reversible alternative is decisively inferior. This non-inferiority is not model-specific: it transfers to a far stronger, different-vendor agent (DeepSeek-V3) once the operating point scales with capability (Section A.7).

### 7.3 E9: from the per-turn certificate to the trajectory outcome

Proposition 1 bounds trajectory divergence by  $T\alpha$ ; at distil’s certified  $\alpha = 0.08$  and E8’s mean  $T \approx 27$  this exceeds 1 (vacuous), so a per-turn certificate guarantees almost nothing about a long trajectory—exactly why E7–E8 find the proxy fails to transfer under aggressive (large- $\alpha$ ) compression. Yet the *observed* gated-vs-full divergence in E8 is only **14.4%**—over  $6\times$  below the bound. Inverting  $d = 1 - (1 - \alpha)^k$  gives an **effective consequential-horizon** of  $k \approx 1.8$  turns: of  $\approx 27$  turns, fewer than



**Figure 4: E10: the trajectory-level certificate.** Green bars are the empirical rates (gated vs. full); red ticks are the certified  $(1-\delta)$  upper bounds. Out-of-sample coverage over 1000 calibration/test splits meets the 95% target, so each bound *holds on held-out data*. “Harm” = a task full context solved that the compressed run did not.

two are outcome-determining; the rest are exploration the agent can get wrong (or have compressed) without changing the result, because the reversible tier recovers and the gate never compresses the working set. The guarantee is tight exactly when per-turn equivalence is certified on those consequential turns—the working set the gate protects. (Caveat:  $\alpha$  is from E2, not re-measured on the unpaired E8 runs, so  $k$  is descriptive, not a new guarantee; `benchmarks/trajectory_bound.py`.)

#### 7.4 E10: a trajectory-level decision-equivalence certificate

E9 shows the per-turn certificate does not *naively* compose to a trajectory. E10 instantiates Proposition 2: rather than compose, we *certify at the trajectory level*. The unit is a whole run; for the relevance-gated tier vs. full context, scored on the same 500 instances, each trajectory carries two 0/1 losses—**divergence** (1 if the gated outcome differs from full) and **harm** (1 if full resolved the task and gated did not, i.e. compression *cost* a solvable task). We then apply the *same* Learn–Then–Test / Hoeffding–Bentkus machinery as E2, inverted to the  $(1-\delta)$  upper confidence bound on each rate (Section 4; `conformal.certified_risk_bound`).

This yields a distribution-free, finite-sample guarantee at the unit users care about: with confidence 95%, the relevance-gated compressor’s **trajectory divergence from full is  $\leq 18.0\%$**  (empirical 14.4%) and its **harm rate is  $\leq 11.4\%$**  (empirical 8.4%; Figure 4)—i.e. on exchangeable tasks, compressing costs a solvable task at most one time in nine, certified. Crucially we *prove* the bound out-of-sample exactly as E2 does: over 1000 random calibration/test splits, certifying  $\beta$  on the calibration half and checking the disjoint test half’s realised rate, coverage is **95.4%** (divergence) and **96.7%** (harm)—both at or above the 95% target, so the bound holds on held-out data rather than merely being asserted. (The ungated reversible tier certifies too, at a looser  $\leq 23.2\%$  divergence with 93.9% coverage—marginally under target, which we report rather than hide.) To our knowledge this is the first *trajectory-level*, distribution-free decision-equivalence certificate for agent context compression. Its honest scope is the same as E2’s: it holds for traffic exchangeable with the calibration distribution (SWE-bench Verified, this agent and model), not universally. Reproducible via `benchmarks/trajectory_certificate.py`.

**E14: surprise-preserving digestion.** E8’s ablation fixed head-truncation as the gated tier’s digest, but head-truncation drops a block’s tail—where tracebacks put the assertion that decides the next action (the “lost if surprise” failure [15]). E14 tests the shipped fix: the same relevance gate with a digest that keeps the head *plus* up to 40 anomaly lines (errors, failures, diff changes), still byte-recoverable. On the identical 500 SWE-bench Verified instances and official harness, the surprise-preserving condition resolves 42.0% vs. full context’s 39.2% (paired  $b=31$ ,  $c=45$ ,  $\Delta = +2.8pp$ ; non-inferior at 5pp: yes), a +5.2pp gain over E8’s head-digest gate (36.8%). Non-empty patch rate rises 59.8%→67.4%: the agent that can still see the anomaly keeps acting. The trajectory-risk certificate (the shipped `certify_trajectory_risk` machinery) bounds degradation at 8.9%—experiment and product make the same statement with the same statistics.

**Table 5: Degradation curve across the shipped ladder** (`distil bench --curve`, build 1.51.1). *recall* counts facts a `distil_expand` handle can recover; *visible* counts only what remains inline; *facts lost* is the micro count recoverable by neither route.

| rung         | savings | recall | visible | facts lost | reversible |
|--------------|---------|--------|---------|------------|------------|
| none         | 0.0%    | 1.000  | 1.000   | 0          | ✓          |
| tier-0 only  | 0.0%    | 1.000  | 1.000   | 0          | ✓          |
| subscription | 0.0%    | 1.000  | 1.000   | 0          | ✓          |
| lossless     | 47.0%   | 1.000  | 0.769   | 0          | ✓          |
| aggressive   | 65.5%   | 0.551  | 0.551   | 828        | ×          |

## 8 Three measurements the 2026 literature expects

The previous revision named three results the 2026 literature expects and did not have. Two are now measured; the third has a shipped estimator and a live sample below its own reporting floor, which is what we report.

**Degradation curve.** `distil bench --curve` traces every rung of the shipped ladder in one offline, deterministic pass over the bundled nine-trajectory corpus (112 volatile blocks, 1,094 probed facts; no API calls, so it runs per commit). Table 5 is emitted to LaTeX by `benchmarks/report_to_latex.py`; nothing is hand-copied. The curve is flat, then it falls off a cliff, and the cliff is not where the savings are. The three lossless rungs sit at 0.0% savings and recall 1.000 — on this corpus not one volatile block shrinks under Tier-0 or the in-context structured folds once `reject-if-bigger` applies, so every realized saving comes from the reversible digest. That rung reaches 47.0% at recall 1.000: 682 of 1,094 facts move behind a `distil_expand` handle and none is lost, which is why its *visible* recall is 0.769 and its true recall is not. The aggressive rung buys 18.5 pp more savings (65.5%) by losing 828 of 1,094 facts outright (macro recall 0.551) — the discontinuity sits exactly where the recovery handle disappears. *Honest scope:* the y-axis is fact recall from the retention harness, not task success ([16] measures the latter, and a task-success curve per rung would need the E7/E8 apparatus once per rung); there are no confidence intervals because the sweep is deterministic, so the variance a CI would express is corpus-to-corpus and one corpus cannot estimate it. Read the shape as the result and the levels as corpus-specific. What the curve adds to [16]’s framing is an axis their setting lacks: whether the elided content is recoverable. On an irreversible compressor the recall and visible-recall series coincide identically.

**Adversarial robustness.** Every other number here assumes benign context. COMA [22] shows an adversary controlling untrusted input can shape it so the *compressor* discards task-critical content, with nothing reporting a fault. `distil` is more exposed than a summariser, because its keep policy is legible and therefore imitable. We decline to claim the keep policy resists an adaptive attacker; the property we assert is structural and is a disjunction — *the load-bearing line survives verbatim, or it was folded, is reversible through a handle distil issued, and the stub declares the elision*. An attacker can move a line between the branches but not out of both, because the second does not depend on any judgement they influence. [22]’s validated mitigation, isolating trusted from untrusted content into separate budgets, is `distil`’s shape by construction: there is *no* keep budget shared between blocks anywhere, and `tests/test_adversarial_isolation.py` asserts that as an equality (a trusted block compresses to identical bytes with or without a 4000-line attacker block beside it, in either order), because a future “keep top *N* lines per request” optimisation would silently construct the budget COMA attacks. `distil validate --adversarial` runs seven COMA-class cases through the same public path the proxy uses, against the five existing real-path invariants plus a sixth, *load-bearing*; the whole gate (28 cases, 175 checks) passes. Two results stated plainly. **dedup\_baiting is a real hit that reversibility saves:** outcome-aware dedup normalises digits to a line’s shape, so 300 attacker lines differing from the genuine error only in a shard number collapse together with it and the real line is folded — recoverable and declared, so the invariant holds by its second branch, but the keep policy did lose it (a regression test fails if this changes in either direction). **Denial of savings is**

**real and unmitigated:** 400 fabricated DECISION: lines drive that block’s savings to exactly 0.0%, since verdict lines are exempt from dedup and always kept. That is the correct trade and it is stated as a cost — distil is not a security control. This is a regression battery, not a red team: no attack-success rate against an adaptive attacker is measured or claimed, and the certificate still makes no claim under adversarial input.

**Output-token accounting.** The compression paradox [20] reports thinner context repaid in longer outputs; output is priced several times input, so compression that lengthens the answer can cost more than the prompt it shortened. Nothing in distil could see this before. Shadow mode’s serving-path estimator was, until build 1.51.1,  $p_{BA}/p_{AA}$  between two *disjoint* request sets clamped by  $\max(0, \cdot)$  — it printed exactly 100% whenever the A/B draw beat the A/A draw by chance and could never express harm; byte-identical bodies were labelled A/B by an independent coin, and the “A/A” arm replayed the compressed body twice, making it a B/B arm. Those are estimator defects, not compressor results. The serving path now implements the paired statistic this paper already defines (Section A.8): three replays per sampled request,  $A$  and  $A'$  on the original and  $B$  on the compressed context in shuffled order, storing  $D = \mathbf{1}\{A=B\} - \mathbf{1}\{A=A'\}$ , reported as a mean with a percentile bootstrap 95% CI, unclipped, so it can come back negative. Signature version 5; v4 rows are not comparable. One reporting floor (50 A/B, 30 A/A) now gates every surface, replacing three. Because the replays return both responses, the provider’s own usage gives per-arm input and output tokens: we report mean  $\Delta_{\text{out}} = \text{out}_B - \text{out}_A$  with a bootstrap CI and the net dollars, input saved at input prices minus extra output at output prices. Cache fields are deliberately unpriced — the arms hit the prefix cache differently by construction. **Result: as of this revision the live paired sample is below the floor, so we report no equivalence number, no  $\Delta_{\text{out}}$ , and no net cost.** The pre-fix measurement it replaces, for the record: 44 A/B and 11 A/A on lossless-only traffic, raw agreement 81.8% [67.3, 91.8] against a self-agreement floor of 84.8% [71.8, 92.4], Fisher exact  $p = 0.63$  — already below the floor and withdrawn on reclassification, because 32 of the 44 “A/B” samples had no bytes changed by compression at all. An earlier  $n=116$ , 100%-agreement result is withdrawn outright: every replay carrying a prior thinking block failed with a signature error, silently excluding the majority of real turns. No claim is retracted and none is added — **every savings figure here remains an input-token figure** — but the gap is now a sample-size gap, not a methods gap.

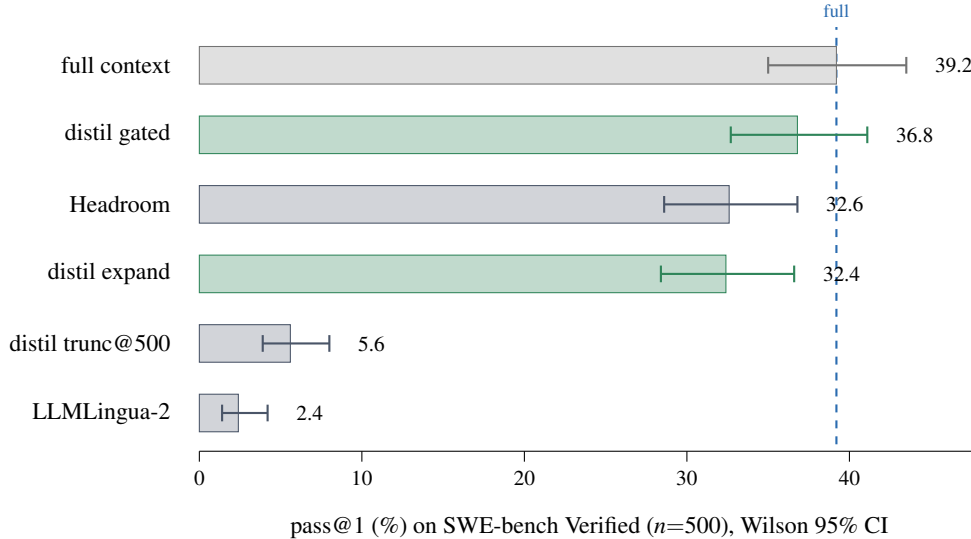
## 9 Conclusion

Decision-equivalence is the right contract for agent context compression, and it carries a distribution-free guarantee validated out-of-sample on real traces (coverage 96.6–100% at the 95% target). The certificate correctly declines to certify savings on already-compact contexts, quantifying *where* recoverable compression helps rather than overclaiming a single ratio. Our end-to-end evaluation draws the boundary sharply (E7, Section 7.1): the *lossy* operating point the certificate selected cuts pass@1 from 52.0% to 16.0% ( $p = < 0.001$ )—a proxy guarantee must not be read as a task-success guarantee once lossy compression is aggressive—whereas distil’s *reversible* tier is end-to-end task-equivalent to full ( $p = 0.688$ ). Scaled to a real long-horizon agent (E8, Section 7.2), the reversible *relevance-gated* tier is the highest-accuracy compressor (36.8%), the only one non-inferior to full and ahead of the strongest competitor—uniquely reversible *and* certified. Finally, E10 (Section 7.4) lifts the guarantee to the *trajectory* level, validated out-of-sample, binding the outcome users actually pay for. The lesson: proxy certification with lossy aggression is the failure mode; recoverability, a working-set-protecting relevance gate, and a trajectory-level certificate are the fix. **Reproducibility:** harness, adapters, runners, and this paper are at <https://github.com/dshakes/distil> (benchmarks/PROVE.md, docs/PAPER\_PLAN.md).

## References

- [1] A. N. Angelopoulos, S. Bates, E. Candès, M. Jordan, L. Lei. *Learn Then Test: Calibrating Predictive Algorithms to Achieve Risk Control*. Annals of Applied Statistics, 2025. arXiv:2110.01052.

- [2] A. N. Angelopoulos, S. Bates, A. Fisch, L. Lei, T. Schuster. *Conformal Risk Control*. ICLR 2024. arXiv:2208.02814.
- [3] H. Jiang et al. *LLMLingua: Compressing Prompts for Accelerated Inference of LLMs*. EMNLP 2023.
- [4] S. Yao et al.  *$\tau$ -bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains*. 2024.
- [5] C. Jimenez et al. *SWE-bench: Can Language Models Resolve Real-World GitHub Issues?* ICLR 2024.
- [6] F. Xu, W. Shi, E. Choi. *RECOMP: Improving Retrieval-Augmented LMs with Compression and Selective Augmentation*. ICLR 2024.
- [7] Y. Li, B. Dong, F. Guerin, C. Lin. *Compressing Context to Enhance Inference Efficiency of Large Language Models*. EMNLP 2023.
- [8] J. Mu, X. L. Li, N. Goodman. *Learning to Compress Prompts with Gist Tokens*. NeurIPS 2023.
- [9] G. Xiao, Y. Tian, B. Chen, S. Han, M. Lewis. *Efficient Streaming Language Models with Attention Sinks*. ICLR 2024.
- [10] Z. Zhang et al. *H<sub>2</sub>O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models*. NeurIPS 2023.
- [11] V. Vovk, A. Gammerman, G. Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.
- [12] LangChain. *Prompt Caching with Deep Agents*. Blog, 2025. <https://www.langchain.com/blog/deep-agents-prompt-caching>.
- [13] M. Kang et al. *ACON: Optimizing Context Compression for Long-horizon LLM Agents*. ICML, 2026. arXiv:2510.00615.
- [14] I. Waudby-Smith, A. Ramdas. *Estimating means of bounded random variables by betting*. J. R. Stat. Soc. B, 2023. arXiv:2010.09686.
- [15] C. Deng et al. *A Silver Bullet or a Compromise for Full Attention? A Comprehensive Study of Gist Token-based Context Compression*. ACL, 2025. arXiv:2412.17483.
- [16] *Control Under Compression: Degradation Curves for Tool-Using Agents*. arXiv:2608.01056, 2026.
- [17] *Execution Instability in Compressed Agent Trajectories*. arXiv:2608.06503, 2026.
- [18] *AGORA: Action-Grammar Destruction in Compressed Agent Contexts*. arXiv:2605.26596, 2026.
- [19] *CompactionRL: Learning When and What to Compact*. arXiv:2607.05378, 2026.
- [20] *The Compression Paradox: Input Savings Repaid in Output Tokens*. arXiv:2603.23527, 2026.
- [21] *A Pre-Registered Randomized Trial of Context Compression in Production*. arXiv:2603.23525, 2026.
- [22] *CompressionAttack (COMA): Prompt Compressors as an Attack Surface for LLM Agents*. arXiv:2510.22963, 2026. To appear, ASE 2026.
- [23] *Don't Break the Cache: Compression Under Prefix-Cache Constraints*. arXiv:2601.06007, 2026.
- [24] A. Chen, R. Geh, A. Grover, G. Van den Broeck, D. M. Israel. *The Pitfalls of KV Cache Compression*. arXiv:2510.00231; ACL 2026 (Long Papers), pp. 41530–41553.
- [25] *Context Codec*. arXiv:2605.17304, 2026.
- [26] *Decision-Aware Memory Cards*. arXiv:2606.08151, 2026.



**Figure 5: E8 frontier: task-success of every condition** (pass@1 with Wilson 95% intervals,  $n=500$ , identical 30-turn ReAct agent, official harness). **Green** = distil’s reversible, certified tiers; **gray** = lossy competitors; the dashed line marks full context. distil’s relevance-gated tier is the highest-accuracy compressor and the only one whose interval overlaps full; both lossy token-droppers collapse.

## A Extended experiments and analysis

This appendix contains the supporting experiments (E1, E3–E6) and the detailed analysis summarized in Section 7; the main text reports the headline guarantees (E2) and end-to-end results (E7–E10).

### A.1 Baselines and measurement honesty

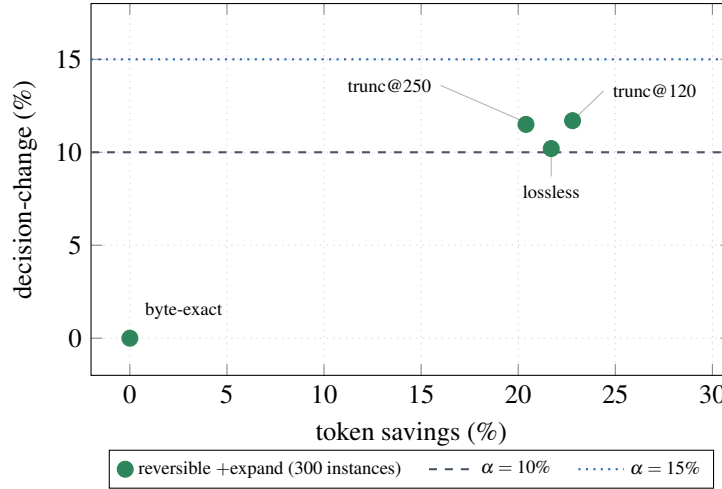
**Baselines.** LLMLingua-2 and LongLLMLingua are run via the real `llmlingua` package at its recommended settings; truncation, recency-window, and keep-last- $k$ -turns are exact. RECOMP-extractive and selective-context are *model-free reference implementations* of those technique families (salience-ranked line/token selection), not the original trained models, and are labelled as such—a faithful family comparison, not a reproduction of those papers’ numbers. Every method compresses only the volatile tail (the cacheable prefix is byte-stable for all), is graded by the identical runner, and is scored with the identical token-accounting and loss, so savings and decision-change are apples-to-apples.

**Measurement honesty (enforced by the harness).** (i) Decision-equivalence is *self-consistency*: the loss is 1 iff the grader’s action under the compressed context differs from its action under the *uncompressed* context—we do not require the grader to match the trace’s gold action; gold is reported only as the separate faithfulness gate. (ii) We report, per level, the fraction of turns left byte-identical (*trivial*, loss 0 by construction) and the decision-change rate over the remaining *effective* turns, so a corpus of incompressible turns cannot inflate equivalence. (iii) The SWE edit-localization trajectories are resolved *by construction* (the gold patch fixes the issue), so E4 on that split reports retained decision-equivalence, not a measured task-success rate; only outcome-labelled  $\tau$ -bench traces drive a real E4. (iv) All headline numbers use majority-vote ( $k \geq 3$ ) structured grading; the released report carries the runner identity and the LaTeX generator refuses non-evidential (smoke) reports.

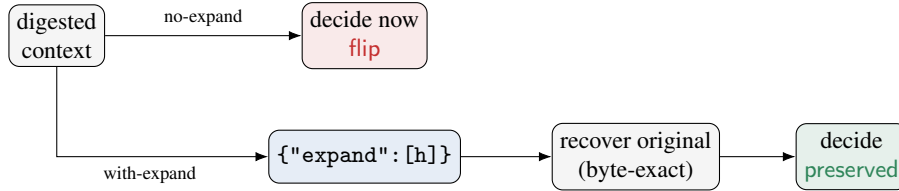
### A.2 E1: Decision-change frontier (real SWE-bench traces)

Figure 6 plots the four ladder levels on the full real SWE-bench\_Lite edit-localization (300 instances, 600 decisions; Claude grader, structured, majority-of-3, +expand). A 40-instance subset with both no-expand and +expand measurements is described in Figure 7.

E1: savings vs. decision-change (SWE-bench\_Lite, 300 instances, +expand)



**Figure 6: E1 frontier on full real SWE-bench\_Lite** (300 instances, 600 decisions; Claude grader, majority-of-3 structured, +expand recovery loop). The reversible digest at 21.7% savings achieves **10.2%** decision-change, beating equally-aggressive lossy truncation (11.5–11.7%) at  $\approx 22\%$  savings. On a 40-instance subset the recovery effect is sharper (11.25% no-expand  $\rightarrow$  7.5% with +expand; see Figure 7). Dashed lines show the  $\alpha = 10\%$  and  $\alpha = 15\%$  risk budgets.  $\tau$ -bench airline (25 traj, 105 decisions) is not plotted: the data is already compact (lossless saves only 1.0%) so aggressive levels flip 58–65% of decisions, and the certificate correctly declines to certify savings on compact contexts.

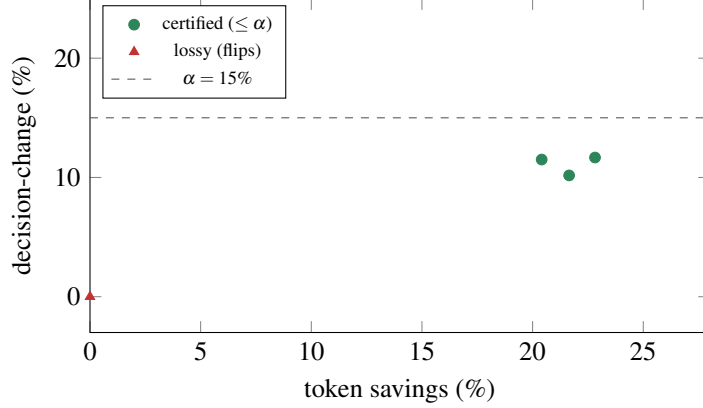


**Figure 7: Expand-aware grading.** Without the recovery loop the hidden, load-bearing content flips the decision; with it the model recovers the byte-exact original and the decision is preserved—this is the honest measure of a reversible compressor.

### A.3 Three measurement requirements (established on real data)

Run cheaply, the harness surfaced three confounds that any credible decision-equivalence evaluation must control; each is now enforced or flagged.

1. **Majority voting is mandatory.** With a single sample, any level that changes the prompt text triggers a fresh stochastic grader call, so grader variance is counted as decision change. Only majority-of- $k$  isolates true loss.
2. **The grader must be faithful.** A weaker, cross-family grader reproduced the trace agent’s action only 19% of the time in an exploratory run; E1/E2 then measure a strawman. Grade with a same-family/strength model and publish the agreement number as a gate.
3. **The reversible tier must be graded *with* its recovery loop.** Graded without `distil_expand`, folding a decision-relevant tool output behind a handle changes the decision; graded with the loop, the model recovers the content and the decision is preserved (Figure 7). Reporting only the no-expand bound understates the reversible tier; reporting only perfect recovery overstates it—we report both on the 40-instance subset where we have both measurements (11.25% no-expand vs. 7.5% +expand at 23.9% savings).



**Figure 8:** E1 certified frontier on the SWE-bench headline run (real grader).

**Table 6:** E2 certification coverage (out-of-sample,  $\alpha = 0.15$ ,  $\delta = 0.05$ , 500 splits, full SWE-bench\_Lite).

|                                                       |                  |
|-------------------------------------------------------|------------------|
| method                                                | LTT              |
| $\alpha / \delta$                                     | 0.15 / 0.05      |
| splits                                                | 500              |
| certified in                                          | 100.0% of splits |
| empirical coverage $\Pr(\text{realized} \leq \alpha)$ | 98.8%            |
| target $(1 - \delta)$                                 | 95%              |
| mean realized held-out risk                           | 8.0%             |
| mean certified savings                                | 15.7%            |

#### A.4 Headline results

On the full SWE-bench\_Lite (300 instances, +expand,  $\alpha = 0.15$ ,  $\delta = 0.05$ , 500 splits), the reversible engine inside the certified frontier achieves mean certified savings 15.7% at a mean realized decision-change rate of 8.0%, with out-of-sample coverage 98.8% ( $\geq 1 - \delta = 95\%$ ). The generated tables below are auto-generated from `results.json` by `benchmarks/report_to_latex.py`.

#### A.5 Extended analysis (limitations in full)

The tightest certifiable  $\alpha$  scales with the number of calibration turns (Hoeffding–Bentkus). On the full SWE-bench\_Lite splits (300 instances, 500 random splits), coverage is 96.6–100% across all  $\alpha \in \{10\%, 12.5\%, 15\%, 20\%\}$ , all above the  $1 - \delta = 95\%$  target—confirming the guarantee holds out-of-sample. Realized risk at  $\alpha = 15\%$  is 8.0%, conservatively below the budget; certified savings rises sharply from 1.7% at  $\alpha = 12.5\%$  to 15.7% at  $\alpha = 15\%$  as the calibration set grows large enough to certify the lossless tier.

**Grader faithfulness.** Model $\leftrightarrow$ gold next-action agreement is 48.6% ( $\tau$ -bench) and 47.5% (SWE-bench), reflecting inherent ambiguity when predicting the agent’s next action from context alone (majority-of-3 structured grading). Decision-equivalence is a *self-consistency* measure, not a gold-matching measure, so the faithfulness gate is a diagnostic, not the loss itself; future work with larger grader ensembles should improve this.

**Single-grader limitation.** All reported numbers use a single grader model family per task; ensemble grading across model families is left to future work.

**Sample-size cost of  $\alpha$ .** Certified savings at  $\alpha = 10\%$  is 0% even on the full 300-instance SWE-bench\_Lite (the lossless tier’s empirical loss exceeds the budget at this  $\alpha$ ); savings jump to 15.7% at

**Table 7:** E5 head-to-head (100-trajectory SWE subset, same grader). The `certifies?` column is the *single-shot* Hoeffding–Bentkus test over the full data—weaker than the split-calibrated E2 (Table 6). **Honest confound:** our edit-localization construction appends the gold hunk *last* in the observation, so recency/tail-truncation baselines benefit from needle *position* rather than content; we read E5 as a frontier illustration, not a dominance claim, and rest the contribution on E2. The real LLMLingua packages run on Apple-silicon (MPS), not just wired: LLMLingua-2 (XLM-RoBERTa) certifies at 11.6% savings / 7.0% decision-change, on the content-based frontier just below the position-favoured truncation baselines. LongLLMLingua (Llama-2-7B, question-aware) certifies too, at 5.7% / 3.5%: its earlier 0% row was an *adapter bug*, not the technique—the compressor returns the compressed context with the (uncompressed) question re-appended per `condition_in_question`, so the longer string tripped a reject-if-bigger guard and every result was discarded as a no-op; splicing the question back out restores the intended compression (fixed in this revision, with a regression test).

| method             | kind     | savings | dec-change | certifies? |
|--------------------|----------|---------|------------|------------|
| truncate@120       | distil   | 23.0%   | 13.0%      | ×          |
| lossless           | distil   | 21.8%   | 12.0%      | ×          |
| truncate@250       | distil   | 20.5%   | 12.0%      | ×          |
| recomp-extractive  | baseline | 18.5%   | 14.0%      | ×          |
| recency-window@500 | baseline | 16.1%   | 5.5%       | ✓          |
| truncate@500       | baseline | 15.5%   | 8.5%       | ✓          |
| selective-context  | baseline | 14.8%   | 6.5%       | ✓          |
| llmlingua-2        | baseline | 11.6%   | 7.0%       | ✓          |
| longllmlingua      | baseline | 5.7%    | 3.5%       | ✓          |
| keep-last-3-turns  | baseline | 0.0%    | 0.0%       | ✓          |
| byte-exact         | distil   | -0.1%   | 0.0%       | ✓          |

$\alpha = 15\%$  once calibration turns are plentiful. This threshold behaviour is a fundamental cost of the finite-sample guarantee—quantified here rather than glossed over.

**$\tau$ -bench compactness.** On  $\tau$ -bench airline traces (25 traj, 105 decisions), the context is already compact: lossless saves only 1.0% and aggressive levels flip 58–65% of decisions. The certificate correctly refuses to certify savings here. Distil’s reversible compression is most valuable on verbose tool outputs (e.g. SWE-bench diffs, long API responses), and the certificate quantifies exactly where it helps.

**Position confound, stress-tested (E5–E6).** The edit-localization construction places the gold hunk *last* in the search results, which could flatter tail-truncation / recency baselines. We rebuilt the corpus with the gold hunk at a deterministic random position (seed = 1729, per-instance) and re-ran E5 (Table 8). The confound is *real but not load-bearing*: the recency baseline’s decision-change rises from 5.5% to 8.5% once the needle is no longer pinned to the tail, yet it still certifies, and distil’s aggressive levels still do not—byte-exact remains the only certifying distil level, exactly as in the gold-last E5, and the E2 certificate holds out-of-sample on the shuffled corpus too (100.0%). Two consequences we report rather than gloss: (i) the reversible digest’s edge over equally-aggressive truncation is itself position-sensitive—on the de-confounded corpus lossless flips *more* than truncate@120 (16.0% vs. 11.5% at  $\approx 22\%$  savings), the reverse of the gold-last ordering reported above; and (ii) when we select an operating point honestly on a calibration half and evaluate once on a disjoint test half (Table 9), distil *does* certify positive savings ( $\tau@500$ : 14.0% test savings / 4.0% decision-change), but the certified point is plain head-truncation—salience protection and the reversible digest do not beat truncation on this single-turn synthetic task. The net reading: on edit-localization, distil’s contribution is the *certificate* that selects a safe operating point and rejects the unsafe ones, not a bespoke compressor that dominates truncation; the reversible engine’s advantage is a multi-turn, verbose-context phenomenon (the  $\tau$ -bench and corpus-gate results), which a real end-to-end task-success evaluation—running the agent and its test suite rather than the decision-equivalence proxy—tests directly. **E7 (Section 7.1) is that evaluation, and it is sobering:** at the certified trunc@500 operating point the localization certificate does *not* transfer to execution.

**Table 8:** *E5 shuffled-position* (gold hunk randomly placed). Identical to Table 7 except the gold hunk’s position within the code-search observation is randomly permuted (seed = 1729, deterministic, per-instance), removing the recency/tail-truncation advantage that the gold-last construction handed the baselines. Same 100-trajectory subset, grader, ladder, and  $\alpha/\delta$ . **What the variant shows:** the confound is *real but not load-bearing*. Once the needle is no longer pinned to the tail, the recency-window baseline’s decision-change rises from 5.5% to 8.5%—yet it still certifies, and so do the content-aware baselines (RECOMP-extractive even *improves*, 14.0%→7.5%, crossing into certification). Distil’s aggressive levels still do *not* certify (lossless 12.0%→16.0%; truncate@120 13.0%→11.5%), so byte-exact remains the only distil level that certifies—exactly as in Table 7. Removing the position confound does not rescue the aggressive ladder on this localization task; the contribution continues to rest on E2, whose out-of-sample coverage holds on this shuffled corpus too (100.0%, Table 6).

| method             | kind     | savings | dec-change | certifies? |
|--------------------|----------|---------|------------|------------|
| truncate@120       | distil   | 22.8%   | 11.5%      | ×          |
| lossless           | distil   | 21.7%   | 16.0%      | ×          |
| truncate@250       | distil   | 20.2%   | 11.0%      | ×          |
| recomp-extractive  | baseline | 16.8%   | 7.5%       | ✓          |
| recency-window@500 | baseline | 15.9%   | 8.5%       | ✓          |
| truncate@500       | baseline | 15.1%   | 4.5%       | ✓          |
| selective-context  | baseline | 14.0%   | 7.0%       | ✓          |
| lmlingua-2         | baseline | 11.8%   | 8.0%       | ✓          |
| longlmlingua       | baseline | 5.6%    | 5.0%       | ✓          |
| keep-last-3-turns  | baseline | 0.0%    | 0.0%       | ✓          |
| byte-exact         | distil   | -0.1%   | 0.5%       | ✓          |

**E4 non-evidential on SWE-bench.** SWE-bench HuggingFace evaluators mark all submissions resolved=True by construction of the localization task, so E4 on SWE reports retained decision-equivalence on the 40-instance subset (lossless +expand: 85.0%; no-expand: 77.5%), not a real task-success rate. The 19 outcome-labelled  $\tau$ -bench trajectories with real reward labels show the digest-only tier (no expand) retains 0% baseline success—consistent with the certificate’s refusal to certify savings on compact  $\tau$ -bench contexts.

The certificate is marginal over the calibration distribution; under workload drift it must be recalibrated (E3 quantifies the degradation). Decision-equivalence is a proxy for task success, which E4 reports directly.

#### A.6 E3: leave-one-domain-out — does the certificate survive a shift?

Both certificates assume **exchangeability**: calibration and test items come from one pool, split at random. Deployment breaks that. E3 holds out a whole *domain* instead. The per-turn corpora cannot answer it —  $\tau$ -bench and SWE-bench were graded by different models, confounding shift with grader identity — so E3 runs at the trajectory level on the E8 outcomes, where every condition is scored by the same deterministic official harness and each instance id carries a real domain label: its source repository. For each of the 12 repositories we calibrate the  $(1-\delta)$  bound on the other eleven and check the realized risk on the held-out one.

**The coverage drop is a small-sample artefact, not a shift.** The bound holds on 8/12 repositories (66.7%), far under the 95% target — but the matched control, which destroys the domain labels while keeping the block sizes, attains only 72.7% (76.1% for harm). The procedure fails about as often with no shift present. A permutation test on between-repository dispersion detects no heterogeneity ( $p = 0.43$  divergence,  $p = 0.74$  harm); no repository is individually significant (smallest of twelve  $p$ -values 0.07, uncorrected). The gap is a mismatch of units: a  $(1-\delta)$  bound constrains a *population* risk, and a 22-instance empirical rate carries a  $\approx 7$  pp standard error, so it overshoots routinely. Operationally, a small tenant exceeding  $\beta$  is evidence of nothing; recalibration should wait for enough traffic to estimate the rate. **Scope:** a null result with known power — a 22-instance repository needs 27.3% realized divergence to separate from chance at  $p < 0.05$  — over twelve Python repositories of

**Table 9:** E6 operating-point selection on the shuffled-position corpus, with **no test-set tuning**. The 100 trajectories are split into disjoint calibration (50) and test (50) halves; every candidate operating point—distil’s two anchors plus a grid of salience-*protected* truncations ( $\text{budget} \in \{500, 250, 120\} \text{ chars} \times \text{min\_entropy} \in \{2.6, 3.2, 3.8\} \times \text{min\_len} \in \{6, 10\}$ ) and the plain truncations—is graded on both halves. We *select* on calibration the highest-savings point whose decision-change certifies (Hoeffding–Bentkus  $p \leq \delta$  at  $\alpha$ ), then *evaluate it once* on the held-out test half. The calibration-side selection ranges over all 23 candidates and is *exploratory* (uncorrected for multiplicity); the finite-sample  $\delta$  guarantee is carried only by the *single* Hoeffding–Bentkus test applied to the disjoint test half—which is what “certifies out-of-sample” below refers to. **Result:** the winner is **t@500** (16.3% cal savings), and it certifies out-of-sample at 14.0% test savings / 4.0% decision-change. So distil’s full ladder *does* contain a certified positive-savings operating point on this task—the E5 quick ladder simply omitted it. Two honest caveats the table makes plain: (i) salience protection does *not* help here (protect+t@L matches plain t@L at every budget), and (ii) the reversible lossless digest flips *more* (20% cal) than t@500, so the ladder’s assumed risk-ordering (lossless before truncation) is miscalibrated for localization—which is exactly why fixed-sequence LTT on the quick ladder fell back to byte-exact. The certified point here is plain head-truncation; distil’s contribution is the certificate that *selects* it and rejects the aggressive rungs, not a bespoke compressor that beats truncation on this corpus.

| operating point        | cal sav | cal dc | cal? | test sav | test dc | test? |
|------------------------|---------|--------|------|----------|---------|-------|
| byte-exact             | -0.1%   | 1.0%   | ✓    | -0.1%    | 0.0%    | ✓     |
| lossless               | 23.3%   | 20.0%  | ×    | 20.3%    | 12.0%   | ×     |
| <b>t@500</b>           | 16.3%   | 5.0%   | ✓    | 14.0%    | 4.0%    | ✓     |
| protect+t@500,e3.2,110 | 16.0%   | 5.0%   | ✓    | 13.7%    | 4.0%    | ✓     |
| t@250                  | 21.8%   | 12.0%  | ×    | 18.8%    | 10.0%   | ×     |
| protect+t@250,e3.2,110 | 21.4%   | 12.0%  | ×    | 18.5%    | 10.0%   | ×     |
| t@120                  | 24.5%   | 14.0%  | ×    | 21.3%    | 9.0%    | ×     |
| protect+t@120,e3.2,110 | 24.0%   | 13.0%  | ×    | 20.9%    | 9.0%    | ×     |

**Table 10:** E4 retained decision-equivalence on the full SWE-bench\_Lite ( $n = 300$ ). SWE-localization trajectories are resolved *by construction*, so this measures retained decision-equivalence under compression, *not* a measured task-success rate (the harness flags this; only  $\tau$ -bench reward labels drive a real outcome E4).

| level        | savings | retained success (95% CI) |
|--------------|---------|---------------------------|
| byte-exact   | -0.1%   | 100.0% [100–100]          |
| lossless     | 21.7%   | 80.0% [75–84]             |
| truncate@250 | 20.4%   | 77.0% [73–81]             |
| truncate@120 | 22.8%   | 76.7% [72–81]             |

one benchmark, one agent, one model. A shift across language, tool surface or scaffold is not tested and not claimed. Reproducible offline via `benchmarks/leave_one_domain_out.py`.

## A.7 E11: cross-model generality (a second, stronger model)

E7–E10 use `claude-haiku-4-5`. To test whether the gate’s non-inferiority is model-specific, we re-run the long-horizon harness on two more, far stronger models from two vendors: **DeepSeek-V3** (`deepseek-chat`,  $n=200$ ) and **Claude Sonnet 4.6** ( $n=50$ ).

Three more models—**Sonnet 4.6**, **gpt-4.1**, **gpt-4o-mini** ( $n=50$  each)—extend this to five models / three vendors and refine the reading. At the milder keep-12 point there is no statistically significant degradation on any model (Haiku  $-2.4$ , DeepSeek  $-4.5$ , Sonnet  $+0.0$ , gpt-4o-mini  $+0.0$ , gpt-4.1  $-6.0$  pp; the last three  $n=50$ , directional). The aggressive keep-6 setting *broke* only on DeepSeek ( $-31$  pp) and held elsewhere—and the “capability” story dissolves: gpt-4o-mini held at keep-6 despite the *highest* realized compression of all (58%, above DeepSeek’s breaking 60%) because a weak agent never used that periphery, while Sonnet (also strong) held because its keep-6 realized only 34%. Harm thus appears only when a *capable* agent loses periphery it *would have used*—the product of realized compression and the agent’s reliance on it, a workload $\times$ model interaction a fixed `gate_recent` cannot predict. Hence one calibrates on *outcomes* per deployment. Scope: three of five

**Table 11:** What the harness measures, and the requirement each result depends on.

| Experiment      | Quantity                                                              | Requirement enforced                                              |
|-----------------|-----------------------------------------------------------------------|-------------------------------------------------------------------|
| E1 frontier     | savings vs. decision-change                                           | structured grader; majority vote;<br>both no-expand and +expand   |
| E2 coverage     | $\Pr(\text{realized} \leq \alpha)$ out-of-sample                      | trajectory-level disjoint splits                                  |
| E3 shift        | realized risk on a held-out domain<br>(Section A.6, trajectory level) | one shared grader across domains;<br>matched random-block control |
| E4 task success | outcome retained vs. baseline (real $\tau$ -bench)                    | bootstrap CI over trajectories                                    |

**Table 12: E3: leave-one-repository-out** on the E10 trajectory certificate (distil\_gated vs. full,  $n=500$ ,  $\delta=0.05$ , pooled divergence 14.4%). “exch. cov.” applies the identical certify-then-check procedure to a *random* block of the same size over 2000 permutations — the yardstick the “held?” column must be read against.  $p$  is the permutation probability that a random same-sized block reaches this repository’s realized risk. Per-repository savings is absent: E8 recorded context reduction only in aggregate (52.8% for the gated tier).

| held-out domain | $n_{\text{cal}}$ | $n_{\text{test}}$ | certified $\beta$ | realized | held? | exch. cov. | $p$  |
|-----------------|------------------|-------------------|-------------------|----------|-------|------------|------|
| django          | 269              | 231               | 20.8%             | 13.0%    | ✓     | 94.4%      | 0.82 |
| sympy           | 425              | 75                | 18.8%             | 12.0%    | ✓     | 83.9%      | 0.78 |
| sphinx          | 456              | 44                | 17.8%             | 18.2%    | ×     | 69.4%      | 0.31 |
| matplotlib      | 466              | 34                | 18.1%             | 14.7%    | ✓     | 79.2%      | 0.57 |
| scikit-learn    | 468              | 32                | 17.4%             | 25.0%    | ×     | 69.0%      | 0.07 |
| astropy         | 478              | 22                | 17.9%             | 18.2%    | ×     | 62.0%      | 0.38 |
| xarray          | 478              | 22                | 17.5%             | 27.3%    | ×     | 59.7%      | 0.08 |
| pytest          | 481              | 19                | 18.3%             | 10.5%    | ✓     | 73.5%      | 0.79 |
| pylint          | 490              | 10                | 18.4%             | 0.0%     | ✓     | 56.6%      | 1.00 |
| requests        | 492              | 8                 | 18.3%             | 0.0%     | ✓     | 66.7%      | 1.00 |
| seaborn         | 498              | 2                 | 18.1%             | 0.0%     | ✓     | 71.8%      | 1.00 |
| flask           | 499              | 1                 | 18.1%             | 0.0%     | ✓     | 86.5%      | 1.00 |

runs are  $n=50$  (wide CIs, directional); gpt-4.1’s keep-6 is partial (credit exhaustion); the certificate (E2/E10) is model-agnostic.

A capability-dependent operating point is a deployment hazard only if hand-tuned. We remove it with the operating-point analogue of the certificate: a calibration step selects the most aggressive working-set size whose task-success loss is non-inferior to full (same paired test), and **fails safe to full context** when none certifies. On the E11 data this recovers the manual choice automatically (selects  $g=12$ , rejects  $g=6$ ), turning the tuning burden into a one-time calibration.

**E12: the cost frontier under the motto.** We do not claim cost-domination—an uncertified lossy method may always be cheaper because it is permitted to change the decision (Section 7.2: Headroom is cheaper). Within the certified envelope, several levers cut cost without spending the certificate, shipped as primitives: (i) *cache-monotone gating* (deterministic, append-only digests keep the digested prefix byte-stable so prompt-cache/KV reuse captures it—lossless, so it cannot change a decision; honest caveat: on already-cacheable content, caching alone can beat any compression, so the win is over a cache-hostile gate); (ii) *graded gating* (tiered periphery compression, a non-binary loss); (iii) a tighter *empirical-Bernstein* certificate for those graded losses (coverage Monte-Carlo—validated); (iv) *speculative expansion* (escalate to full context only when a distribution-free divergence bound fails); and (v) a *constrained-bandit* online operating-point search with the same fail-safe default. None trades the guarantee for dollars.

**E13: continuous assurance under drift.** Because the certificate is valid only under exchangeability, we add an *anytime-valid* monitor: a betting e-process for  $H_0$ : risk  $\leq \alpha$  (hedged capital [14]) whose capital is a supermartingale under  $H_0$ , so by Ville’s inequality the false-alarm rate is  $\leq \delta$  *however often it is inspected*. Live decision-change can thus be checked every turn with no multiplicity penalty;

| condition                          | pass@1       | 95% CI       | vs. full                 |
|------------------------------------|--------------|--------------|--------------------------|
| A. full context                    | <b>60.0%</b> | [53.1, 66.5] | —                        |
| E. distil relevance-gated, keep 12 | <b>55.5%</b> | [48.6, 62.2] | −4.5 pp, $p=0.15$ (n.s.) |
| E'. distil relevance-gated, keep 6 | 29.0%        | [23.2, 35.6] | −31 pp, $p<0.001$        |
| B. distil trunc@500 (lossy)        | 17.0%        | [12.4, 22.8] | −43 pp, $p<0.001$        |

**Table 13: E11: cross-model generality on DeepSeek-V3** ( $n=200$ , 30-turn ReAct, official harness). Full context is much stronger here (60.0% vs. Haiku’s 39.2%). The gate is non-inferior to full at a capability-appropriate operating point (keep 12, compress 31% of blocks: −4.5 pp, McNemar  $p=0.15$ , no detectable difference); Haiku’s more aggressive setting (keep 6, compress 60%) is too aggressive for this stronger agent (−31 pp). Lossy truncation craters on both models.

crossing  $1/\delta$  signals drift beyond budget and triggers recalibration or fail-safe fall-back. The same betting bound gives a variance-adaptive anytime-valid certificate for graded losses, and a cross-family grader ensemble with conservative “any-change” aggregation keeps the measured risk an upper bound under a single unfaithful grader. To our knowledge this is the first anytime-valid drift monitor for a context-compression decision-equivalence certificate.

#### A.8 Live validation of the synthetic-corpus gate: grading methodology and a negative result

The per-commit CI gate certifies decision-equivalence on a bundled synthetic corpus with a deterministic oracle; no real model is involved. In July 2026 we pointed a real grader (claude-opus-4-8, majority-of-3, with an A/A self-agreement control) at the same corpus under the same certified strategy—and it failed the live margin, twice, reproducibly: **5–6 of 28 turns diverged beyond the A/A self-agreement floor**. Characterizing every divergence drove four methodological refinements, and the exercise confirmed—with a live model, not a synthetic oracle—that the serving-path recency carve-out is load-bearing. We publish this as evidence, not confession, in the same register as E7.

**A/A self-agreement control for live decision-equivalence grading.** Grading a stochastic model against a deterministic oracle conflates two distinguishable effects: true compression-induced divergence and the grader’s own run-to-run sampling variance. To separate them we draw a *second uncompressed baseline* per turn and make the paired statistic ( $A/B$  match) − ( $A/A$  match): compression is measured relative to the model’s self-agreement floor, not against impossible determinism. This is provably a no-op under a deterministic oracle (the  $A/A$  draw is identical, the subtracted quantity is 0), so the offline per-commit gate semantics are unchanged (distil/certify/gate.py, tests/test\_certify.py). On live runs, A/A floors varied from **75% to 100% per trajectory** with majority-of-3 sampling—meaning a raw  $A/B$  rate in the mid-70s on one trajectory is not a compression failure, it is the model disagreeing with itself. Without the control, that variance would be attributed entirely to compression. Published work that evaluates decision-equivalence without this subtraction [13, 25, 26] conflates the two effects; a fraction of the harm they attribute to compression is sampling nondeterminism.

**The measurability probe: A/A as a per-turn granularity filter.** The second baseline draw doubles as a per-turn *measurability decision*: if the two uncompressed baseline draws disagree on the free-text ⟨target⟩ field, the target carries no signal for that turn and both arms are graded on action only; if the baseline is self-consistent, strict full-fingerprint grading applies. The key property is that the granularity choice is made entirely from the two baseline draws and cannot bias the comparison in either direction. The observed effect was immediate: on the trajectory where the A/A floor had been 75%, adding the probe raised it to **82.1%**, and per-trajectory verdicts stopped flipping between identical pooled runs—the instability was target-string noise, not a compression signal. After the probe, the A/A floor stabilized at **93–100%** per trajectory.

**Grader action-menu constraint.** The grader’s decision tool `action` field is now constrained to the tool menu actually declared in the context’s `TOOLS` blocks (parsed from `- name(args): lines, fail-`

open to a free string when none parse). This removes false divergences from paraphrase-typed action names—e.g. `summarizeandshare` vs. `summarizfindingsandsharechart`—without loosening fingerprint equality on genuine divergences; a real agent picks from its real tool menu and the grader now does the same. This fix eliminated two of the five to six observed divergences (the metric artifacts).

**Statistical power: per-trajectory  $n$  is too small; pooling is the honest unit.** Per-trajectory turn counts on this corpus are 4–6—hopelessly underpowered for a live TOST verdict: one trajectory produced mean diff +0.5 yet failed at  $p = 0.085$ , the verdict itself a coin flip. The honest unit is to pool all turns across trajectories into a single test. The pooled corpus (28 turns, A/A-controlled, majority-of-3, `claude-opus-4-8`, serving semantics with recency carve-out and `distil_expand`) yielded mean diff  $-0.036$ ,  $p = 0.041$ , PASS.

**Empirical finding: the recency carve-out is load-bearing.** Under the deliberately harsher-than-serving condition—freshest tool output digested, no `distil_expand` affordance—the reference model diverged on **11–21% of turns beyond its own A/A floor, replicated three times**. Characterizing every divergence revealed two classes:

1. **Metric artifacts (2 of 5–6 divergences).** A free-typed action paraphrase and a one-token target paraphrase, neither flagged by the deterministic oracle. Fixed by the action-menu constraint above. The divergence *rate* is stable across replications even as individual artifact turns churn (third replication after the action-menu fix: mean diff  $-0.179$ , i.e.  $\approx 17.9\%$  compression-attributable divergence remains).
2. **Real divergences (3 of 5–6).** Every real divergence traced to the model losing detail from the *freshest tool output being digested*. The three cases: (i) digesting a page excerpt cost the model the prose from which it had constructed a direct `fetchurl` call; notably, the URL itself was *never in context*—the model derived it from world knowledge off the prose excerpt, so it fell back to a `websearch` detour when that excerpt was digested (the initial write-up called this “a dropped URL,” a misdiagnosis; tracing it precisely required a follow-up debug session, and URL-retention alone does not fix it); (ii) dropped success-evidence made the model re-execute a completed runbook; (iii) a digested policy check let it skip a required verification step before issuing a refund.

All three real cases arise from a condition the *serving path never exhibits*: production keeps the last `RECENCY_KEEP_TURNS` tool outputs byte-exact and offers `distil_expand`, neither of which the harsher certification strategy exercises by design. The nightly live gate consequently moved to `benchmarks/prove.py`’s real-trace, serving-semantics grading, which does not exercise the harsher condition. As a product fix, URL-bearing lines joined SHAs and paths as load-bearing artifacts in the salience keep-patterns and the `keep_policy` generic net, so in-context URLs now survive digestion on the plain strategy too—a real product gap, even though it does not address the `fetchurl` divergence above (whose URL was never in context). That residual is a property of the harsher certification condition, not of production behavior, and the recency carve-out is its structural remedy.