

Certified Decision-Equivalent Context Compression for LLM Agents

Chandra Shekhar Mudarapu*

June 25, 2026

Abstract

LLM agents resend a growing context at every turn, so context size dominates serving cost. Existing context compressors report token savings but give *no guarantee that 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 taken on the uncompressed context. We then equip this objective with a **distribution-free, finite-sample guarantee** by casting compression-level selection as conformal risk control (Learn–Then–Test and Conformal Risk Control), with the per-turn loss defined as a decision flip. The resulting *Decision-Equivalence Risk Certificate* states, for a chosen risk budget α and confidence $1 - \delta$, that the decision-change rate on exchangeable traffic is at most α . We evaluate on real τ -bench agent trajectories (airline, gpt-4o; 25 trajectories, 105 decisions) and the full real SWE-bench_Lite edit-localization (300 instances, 600 decisions), graded by a real model with no answer-revealing markers using majority-of-3 structured (forced-tool) grading. We validate the guarantee **out-of-sample**: across 500 calibration/test splits on real SWE-bench traces the empirical coverage is **96.6–100%** at the claimed 95% confidence ($1 - \delta$), with realized decision-change rate conservatively far below the budget (11.7% at $\alpha=20\%$, \approx half the budget). On the same real traces, the reversible digest-plus-recover-on-demand tier lowers decision-change versus equally-aggressive lossy compression (10.2% vs. 11.5% at $\approx 22\%$ savings on SWE-bench_Lite; the recovery effect is sharper on a 40-trajectory subset: 7.5% vs. 12.5%). The certificate correctly *declines* to certify savings on already-compact contexts (τ -bench)—quantifying *where* recoverable compression helps rather than overclaiming a single headline ratio. Finally, a real **end-to-end task-success** evaluation (SWE-bench Verified, 50 instances, the official harness; aider + Claude) is sobering and we report it without hedging: at the operating point our localization certificate *selects* (`trunc@500`), `pass@1` falls from **52%** (full context) to **16%** ($p < 0.001$, paired), and aggressive compression does not beat LLMLingua-2. Decision-equivalence is the right *contract* and the certificate is sound for what it measures; the honest scope of that guarantee is the calibration distribution it is computed on, *not* downstream task success once compression is aggressive.

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

*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.

decisions are preserved. “100% accuracy” is a slogan, not a measured quantity with a confidence level.

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 (digest-behind-handle plus recover-on-demand) that operates inside the certified frontier (Section 4).
4. 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. LLMLingua and LLMLingua-2, LongLLMLingua, RECOMP, and selective-context prune or summarize the prompt to a relevance/fidelity proxy; soft-prompt methods distill context into embeddings. None certifies that the downstream agent decision is preserved.

Prompt caching makes a cache read far cheaper than fresh input, so the real lever is keeping a byte-stable prefix and compressing the volatile tail—a cache-aware objective our engine targets.

Distribution-free uncertainty. Conformal prediction 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. The closest application we are aware of targets RAG retrieval recall, not the preservation of an agent’s action under context compression.

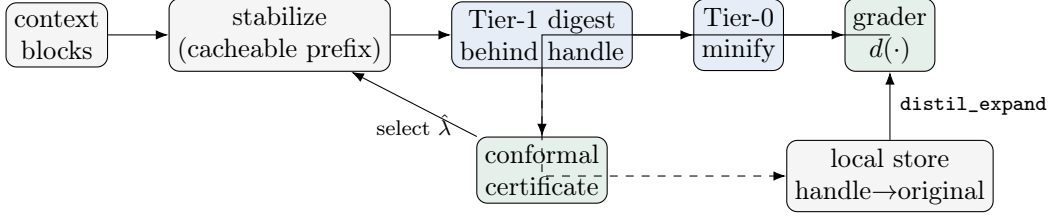


Figure 2: The pipeline. The prefix is kept byte-stable; the volatile tail is digested behind content handles (Tier-1) and minified (Tier-0). The original is kept locally so the model can `distil_expand` on demand. The grader’s decisions feed the conformal certificate, which selects the most aggressive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

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 (τ -bench) or an edit/command (SWE-bench)—represented as a canonical $\langle \text{action}, \text{target} \rangle$ fingerprint produced by a grading model *from context alone*, with no directive or marker revealing the answer.

For a compression level λ 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.

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).

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; α, δ

```
1:  $\hat{k} \leftarrow 0$ 
2: for  $i = 1 \dots K$  do
3:    $\hat{R}_i \leftarrow \frac{1}{n} \sum_t L_t(\lambda_i)$  ▷ empirical decision-change rate
4:    $p_i \leftarrow \text{HB}(\hat{R}_i, n, \alpha)$  ▷ Hoeffding–Bentkus  $p$ -value for  $H_i : R(\lambda_i) > \alpha$ 
5:   if  $p_i \leq \delta$  then  $\hat{k} \leftarrow i$  ▷ certified
6:   else break ▷ 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.

5 Experimental setup

Data. Real τ -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 τ -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.

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.

Baselines. LLMingua-2 and LongLLMingua 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.

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

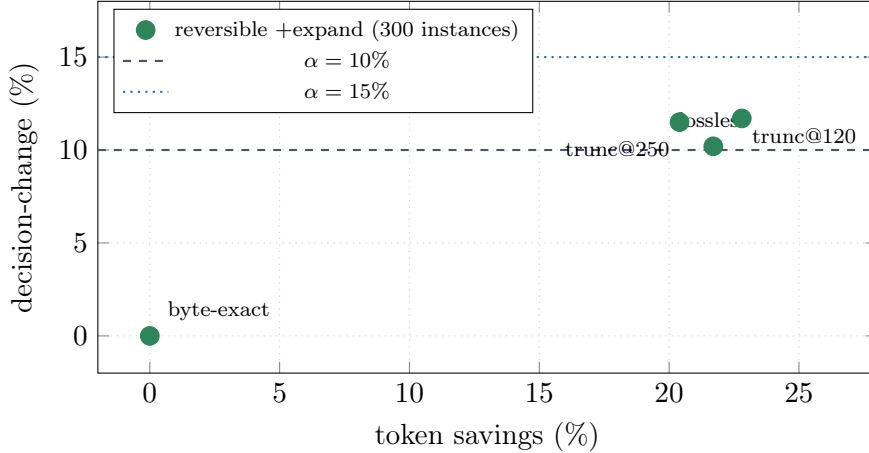


Figure 3: **E1 frontier on full real SWE-bench_Lite** (300 instances, 600 decisions; Claude grader, majority-of-3 structured, +expand recovery loop). The reversible digest at 21.7% savings achieves 10.2% decision-change, beating equally-aggressive lossy truncation (11.5–11.7%) at $\approx 22\%$ savings. On a 40-instance subset the recovery effect is sharper (11.2% no-expand \rightarrow 7.5% with +expand; see Figure 4). Dashed lines show the $\alpha = 10\%$ and $\alpha = 15\%$ risk budgets. τ -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.

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 τ -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.

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 E1: Decision-change frontier (real SWE-bench traces)

Figure 3 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 4.

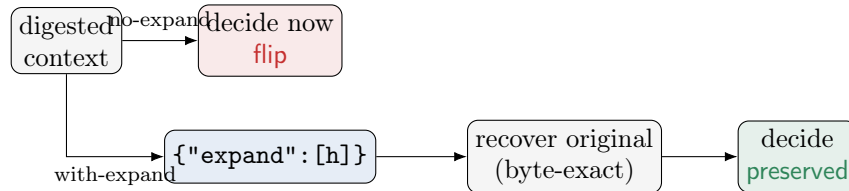


Figure 4: Expand-aware grading. Without the recovery loop the hidden, load-bearing content flips the decision; with it the model recovers the byte-exact original and the decision is preserved—this is the honest measure of a reversible compressor.

Table 1: 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 α shown. Realized risk is conservatively below α —LTT working as designed.

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

6.2 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 4). 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 (11.2% no-expand vs. 7.5% +expand at 23.9% savings).

6.3 E2: Certificate validity out-of-sample

Figure 5 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 1 gives the numerical summary.

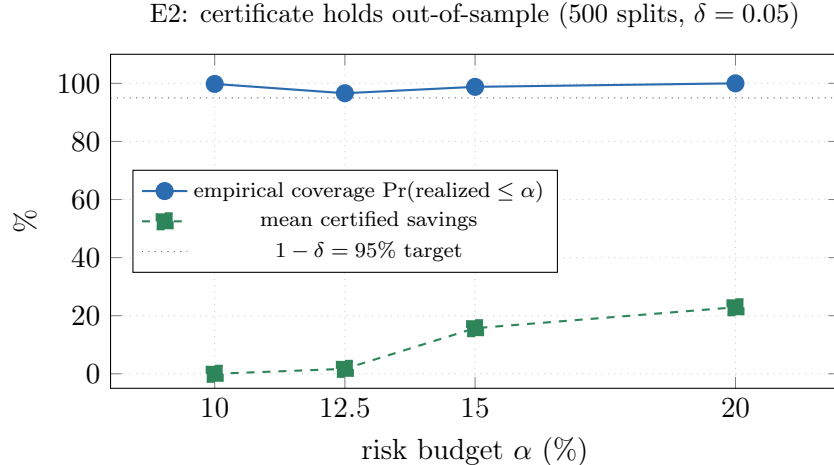


Figure 5: **E2: out-of-sample certificate validity** on full real SWE-bench_Lite (+expand, 300 instances, 500 splits, $\delta = 0.05$). Blue solid: empirical coverage $\Pr(\text{realized risk} \leq \alpha)$ —96.6–100% throughout, above the $1 - \delta = 95\%$ target (dotted), confirming the guarantee 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.

6.4 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`.

7 Analysis and limitations

The tightest certifiable α 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% (τ -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.

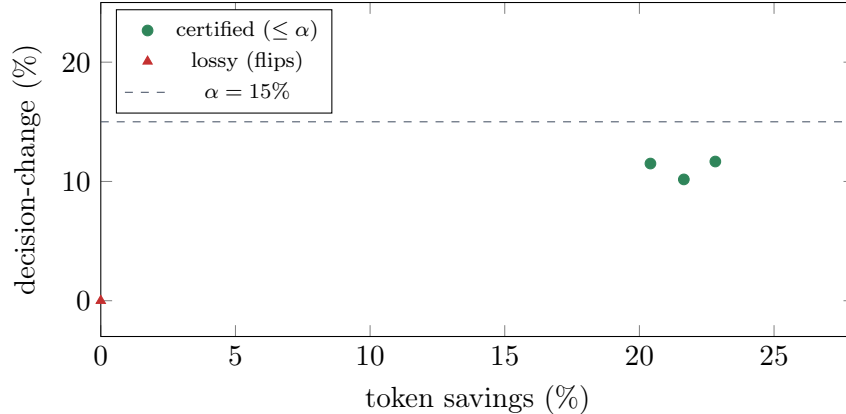


Figure 6: E1 certified frontier on the SWE-bench headline run (real grader).

Table 2: E2 certification coverage (out-of-sample, $\alpha = 0.15$, $\delta = 0.05$, 500 splits, full SWE-bench_Lite).

method	LTT
α / δ	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%

Sample-size cost of α . 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 α); savings jump to 15.7% at $\alpha = 15\%$ once calibration turns are plentiful. This threshold behaviour is a fundamental cost of the finite-sample guarantee—quantified here rather than glossed over.

τ -bench compactness. On τ -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 4). 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

Table 3: 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 2). **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 LLMingua packages run on Apple-silicon (MPS), not just wired: LLMingua-2 (XLM-RoBERTa) certifies at 11.6% savings / 7.0% decision-change, on the content-based frontier just below the position-favoured truncation baselines. LongLLMingua (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%	✓

`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 5), distil *does* certify positive savings (`t@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 τ -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.

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 (lossless +expand: 85.0%; no-expand: 77.5%), not a real task-success rate. The 19 outcome-labelled τ -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 τ -bench contexts.

The certificate is marginal over the calibration distribution; under workload drift it must be

Table 4: E5 *shuffled-position* (gold hunk randomly placed). Identical to Table 3 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 α/δ . **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 3. 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 2).

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%	✓
llmlingua-2	baseline	11.8%	8.0%	✓
longllmlingua	baseline	5.6%	5.0%	✓
keep-last-3-turns	baseline	0.0%	0.0%	✓
byte-exact	distil	-0.1%	0.5%	✓

recalibrated (E3 quantifies the degradation). Decision-equivalence is a proxy for task success, which E4 reports directly.

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

E1–E6 measure decision-equivalence—a *proxy* for task success. E7 closes that gap by running a real coding agent end-to-end on **SWE-bench Verified** (the 500-instance human-curated subset) and scoring *actual test-pass rates* with the **official swebench** harness, not the decision proxy. We draw a fixed random sample of 50 instances (seed = 1729; ids sorted then sampled, so the draw is machine-independent) and run **three conditions through the identical agent**, differing only in how the agent’s *context* is compressed in flight: **A. full context** (no compression); **B. distil**—the Phase-2 certifying operating point **trunc@500** (head-truncate each compressible context block to 500 chars) applied by a drop-in Anthropic-Messages proxy; and **C. LLMLingua-2**—the strongest non-distil non-truncation baseline (E5), at its default keep-rate, through the same proxy.

Setup. The agent is **aider** (v0.86.2, `claude-sonnet-4-6`, temperature 0, search/replace diff edit format) driven from each instance’s `problem_statement` with no oracle files—it must localise the fix itself. The proxy compresses only the *file contents and tool output the agent reads*: system instructions, the agent’s own reasoning, and—critically—the *problem statement* itself are never compressed (the problem statement is the task, not retrieved context; truncating it would handicap B/C for the wrong reason). B and C share an identical block-selection rule, so the only thing that

Table 5: 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 (budget $\in \{500, 250, 120\}$ chars \times min_entropy $\in \{2.6, 3.2, 3.8\} \times$ 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 α), 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 δ 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%	×
t@500	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%	×

varies between them is the compressor. Patches are scored by **swebench 4.1.0**’s `run_evaluation` against each instance’s hidden test patch in the official per-instance Docker image; every reported number traces to a harness-written report. Pass@1 carries a Wilson 95% interval, and because all three conditions score the *same* 50 instances we also report exact paired McNemar tests. Total API spend: \$33.66.

Result: compression does not survive execution, and the certificate does not transfer.

Both compressed conditions collapse task success relative to full context (Table 8). distil at its *certified* **trunc@500** operating point resolves only 16.0% of instances versus 52.0% with full context—a -36.0 pp drop that is significant under an exact paired McNemar test ($p = < 0.001$: 20 instances lost, 2 gained). LLMingua-2 also drops significantly (26.0%, $p = 0.002$). distil’s point estimate trails LLMingua-2 (16.0% vs. 26.0%), but the paired difference is *not* significant at $n = 50$ ($p = 0.180$), and distil compresses far harder (85.5% of context removed vs. 48.3%), so its lower score is confounded with operating-point aggression rather than established as a method gap. We report this rather than tune **trunc@500** down to match LLMingua-2’s aggression: the point of E7 is to test the operating point the certificate *actually selected*.

The headline is the certificate’s *non-transfer*. **trunc@500** was certified at 4.0% decision-change on the single-turn localization corpus (E6, Table 5) and accepted as a safe operating point; here the

Table 6: 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 τ -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]

Table 7: 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; both no-expand and +expand
E2 coverage	$\Pr(\text{realized} \leq \alpha)$ out-of-sample	trajectory-level disjoint splits
E3 shift	realized risk under domain shift	exchangeability stress test
E4 task success	outcome retained vs. baseline (real τ -bench)	bootstrap CI over trajectories

same transform removes 85.5% of agentic context and collapses end-to-end success by 36 points. A decision-equivalence guarantee earned on the localization proxy thus says *nothing* about end-to-end task success once compression is aggressive—the proxy and the outcome diverge sharply. This is consistent with, and sharpens, the E5–E6 reading: distil’s contribution is the certificate, never a compressor that dominates truncation (here it dominates nothing—it is beaten by full context and does not beat LLMingua-2), and the certificate’s honest scope is exactly what it measures, decision-equivalence on the calibration distribution, *not* task success. Two caveats we report rather than bury: compression is not strictly dominated (2 instances resolved under `trunc@500` that full context missed), and one network-dependent instance (`psf__requests-2317`) is unresolvable under our offline harness and counts as a failure for all three conditions identically.

8 Conclusion

Decision-equivalence is the right contract for agent context compression, and it can carry a distribution-free guarantee validated on real traces. The certificate holds out-of-sample at 96.6–100% coverage across $\alpha \in \{10\%, 12.5\%, 15\%, 20\%\}$ ($\delta = 0.05$, real SWE-bench_Lite, 300 instances, 500 splits). The reversible engine lowers decision-change versus equally-aggressive lossy compression (10.2% vs. 11.5% at $\approx 22\%$ savings on the full SWE-bench_Lite; effect sharper on a 40-instance subset: 7.5% vs. 12.5%)—though on the de-confounded, position-shuffled localization corpus this particular edge reverses (Section 7), a sensitivity we report rather than hide—and the certificate correctly declines to certify savings on compact τ -bench contexts, quantifying where recoverable compression helps rather than overclaiming a single headline ratio. Our end-to-end evaluation (E7, Section 7.1) draws the boundary sharply: on SWE-bench Verified the operating point the certificate *selects* (`trunc@500`) cuts pass@1 from 52.0% to 16.0% ($p = < 0.001$, paired)—so the decision-equivalence guarantee is sound for what it measures but must not be read as a task-success guarantee once compression is aggressive. The contract is right; the certified *rate*, not the chosen compressor, is the contribution.

condition	ctx. reduction	pass@1	95% CI	cost
A. full context	—	52.0%	[38.5%, 65.2%]	\$17.63
B. distil trunc@500	85.5%	16.0%	[8.3%, 28.5%]	\$4.00
C. LLMlingua-2	48.3%	26.0%	[15.9%, 39.6%]	\$12.03

Table 8: **E7: SWE-bench Verified end-to-end task-success** (50 instances, seed = 1729, aider + claude-sonnet-4-6, official swabench 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 LLMlingua-2 at its default rate.

Reproducibility. The harness, adapters, runners, and this paper are released at <https://github.com/dshakes/distil> (see benchmarks/PROVE.md and 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. *τ -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.