When Is Tool-Observation Compression Worth Pursuing? Replay Audits Across Agent Interfaces

Anonymous authors
Paper under double-blind review

Abstract

Tool-observation compression is useful only if a transformation applies to the workload, can be decoded from available information, and reduces model-facing cost without damaging behavior. We present a replay audit separating structural support, reuse-weighted mass, paired behavioral leverage and execution integrity. On pinned BFCL tasks, a numeric-array summary saves 41.81% of prompt-plus-completion tokens on eight selected development families but has no qualifying target in 320 held-out reference variants. A typed-subtree state audit gives a different interface comparison: 8.53% of held-out BFCL JSON-object bytes are reconstructible, versus 100% among successful retail and airline reference actions in τ^2-Bench. Access to simulator state, however, does not establish decoder access. History-only controls yield little net saving, and neither tokenizer reaches the frozen 5% savings gate in the always-on counterfactual full-template comparison. A supplementary audit of 848 archived histories shows another boundary: structural checks miss same-type referent changes, whereas trusted fingerprints detect the injected changes but can consume the tool-level savings. In a separate 40-position diagnostic, reference encoding reduces exact JSON reconstruction from 80.0% to 37.5% for Qwen and from 52.5% to 5.0% for Llama. The 12-pair natural-task pilot remains inconclusive. Together, these findings show how interface support, decoder access, and cost accounting can invalidate a promising local compression result. The audit supports experiment selection; it does not establish a deployable codec.

1 Introduction

Language-model agents append tool results to their histories, so an early observation may be processed again at every later request. A long selected trace can make a compressor look useful even when the triggering structure is absent from held-out tasks. Conversely, a common structure may carry too little model-facing cost to matter or may have little causal influence on the next action. Positive development performance can therefore coexist with absent support, negligible leverage, or an invalid execution contrast.

These distinctions matter because "tool observation" is an interface role, not a content type. Results include numeric traces, serialized database objects, fresh search facts, and execution errors. An array summary is useless when arrays are absent; a state reference cannot recover facts outside public state; and exact textual redundancy does not imply policy invariance. An opportunity test should separate four questions: whether the candidate applies, how much repeated model-visible mass it covers, whether a paired intervention changes behavior, and whether execution and evaluation remain valid. Evidence at one stage does not establish the next stage.

We instantiate this test on two public agent benchmarks: BFCL (Patil et al., 2025) and τ^2-Bench (Barres et al., 2026). A deterministic numeric-array summary saves 41.81% of measured tokens on a selected BFCL development slice but has zero support in 320 family-held-out variants. A typed state-reconstructibility test shows the opposite cross-interface pattern: it covers only 8.53% of held-out BFCL JSON-object bytes but all audited τ^2JSON-object bytes, including model-selected calls.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A separate paired reobservation experiment supplies a leverage check: exact duplication is widely eligible yet changes only one of 12 first actions and one of 12 outcomes.

This paper makes three contributions:

1. We define four checks for choosing a compression experiment: held-out support, reuse-weighted mass, paired leverage, and execution integrity. Each check addresses a different reason why a local saving may not transfer.
2. We show an interface-dependent reversal across 2,345 BFCL reference calls, 692 τ^2reference actions, and 466 calls in saved τ^2model trajectories. Long numeric arrays vanish outside the selected development slice, while exact state reconstruction changes from 8.53% of held-out BFCL JSON-object bytes to 100% of τ^2JSON-object bytes.
3. We provide replayed execution, actual provider token counts, call-level ledgers, independent validators, uncertainty, negative outcomes, and recorded measurement corrections for every paper-facing result.

The resulting decision is whether to run a codec experiment. A candidate with substantial support and mass still needs a decoder that can access the referenced information, and a paired test of behavior and cost. Subtree equality alone answers neither question.

2 Related Work

LLMLingua compresses general prompts (Jiang et al., 2023); CoACT trains observation compressors using next-action preservation (Chen et al., 2026). Our question precedes that optimization: does the proposed mechanism apply to held-out tool outputs, and can its decoder use the required information? BFCL and τ^2-Bench expose different tool interfaces (Patil et al., 2025; Barres et al., 2026), making them useful for this comparison. We also charge reference metadata and repeated instructions, and replay execution separately from model-visible histories. These checks expose three specific errors: extrapolating from selected long arrays, treating simulator access as decoder access, and reporting payload savings without request overhead. Appendix H compares related compression and robustness studies.

3 Replay-Audited Opportunity Test

Let X be an evaluation instance, let $E(X) \in \{0, 1\}$ mark whether a candidate transformation is eligible, and let $Y_0, Y_1 \in [0, 1]$ be coupled baseline and intervention outcomes. If the transformation is inert when $E = 0$, then

$$\Delta = \mathbb{E}[Y_1 - Y_0] = \Pr(E = 1)\,\mathbb{E}[Y_1 - Y_0 \mid E = 1]. \tag{1}$$

This is an accounting identity, not a theoretical contribution. It separates held-out *support* from conditional behavioral *leverage*; in particular, $|\Delta| \leq \Pr(E = 1)$. Eligibility must be frozen before outcomes, since a post-hoc predicate that selects successful interventions makes the decomposition tautological.

Consider a trajectory with model requests $q_1, \ldots, q_T$ and tool observations $o_1, \ldots, o_K$. Let u_i be the number of later model requests whose rendered history includes o_i. For a representation cost $c(\cdot)$, the observation's reuse-weighted mass is

$$W(o_i) = u_i c(o_i). \tag{2}$$

Token counts are the correct c when comparing served model requests. Canonical UTF-8 bytes provide a tokenizer-independent structural measure, but they are not a latency or token estimate.

The opportunity test has four parts. *Support* records whether the target structure appears in frozen held-out families and interfaces. *Mass* measures its share of model-facing cost, including reuse. *Leverage* measures actions and task outcomes under a coupled intervention. *Integrity* keeps two views of each run: literal tool calls, raw results, state transitions, and evaluator outputs in the execution view; exact rendered messages and provider token counts in the model view. A valid run must execute in a fresh environment, reconstruct the model-visible history, reproduce the original score, agree

Table 1: Information available to a check or decoder. Each condition must be stated separately from its accuracy or cost.

Source	What the audit can establish
Post-call public state	A typed value exists in simulator state; this does not establish model-visible access.
Previous visible wire	A reference resolves into a payload actually present in the declared history.
Trusted fingerprint map	The referent matches a supplied trusted binding; the map incurs cost and requires a trusted source.
Original observation	An offline oracle can compare reconstruction with the original value; availability at deployment is a separate assumption.

under literal replay, and satisfy its registered repeats. Infrastructure failures are invalid runs rather than zero rewards. Large mass on selected tasks therefore establishes neither held-out support nor behavioral safety.

3.1 Information access and the accounting unit

Reconstructibility is relative to an information source. Table 1 separates four sources used in this study; access granted to the auditor is not automatically granted to the model. Likewise, JSON bytes, single tool-message tokens, repeated request tokens, and prompt-plus-completion totals are different denominators. We do not interpret percentages across these units as stages of one measured savings funnel.

3.2 Two structural candidates

The first candidate is a deliberately simple positive control, not a prior method or competitive baseline. It was chosen to test whether the measurement stack can recover an obvious local saving before any learned compressor is trained. Its fixed 32-element threshold is a prespecified heuristic for that control, not a tuned optimum. In parsed JSON, it replaces every finite numeric array of at least 32 elements with a marked object containing the count, minimum, maximum, mean, first four values, and last four values; all other values remain unchanged. Its failure outside the selected slice is evidence about that candidate and evaluation workflow, not a refutation of array compression.

The second candidate partitions each successful JSON-object response using post-call public state. For response $o = \{k_j : v_j\}$ and state s', field (k_j, v_j) is *state-reconstructible* iff the complete typed value v_j occurs exactly as a subtree of s'. The remaining fields are called *epistemic-only*: they are values the agent must observe directly because this audit cannot recover them from public post-call state. Matching is recursive, type-sensitive, and performed after canonical serialization. Partition serializations include their own keys and braces, so their byte counts are signed representation costs and need not sum exactly to the original response length.

Partial and fuzzy matches are rejected. A positive match still does not make a field safe to delete: the model may lack access to s', keys carry meaning, and reconstruction has a cost. The measure therefore marks possible redundancy without claiming a codec or semantic equivalence.

3.3 Frozen gates and stopping

Thresholds are written before each downstream result is computed. The BFCL numeric control must preserve all five baseline successes on its selected eight long-context families, score at least $5/8$, and reduce aggregate measured prompt-plus-completion tokens by at least 30%. The state-ledger direction continues only if held-out JSON objects cover at least 20% of response bytes, both reconstructible and epistemic partitions cover at least 10% of JSON-object bytes, and at least 20 calls mix the partitions. The corrected redundant-observation pilot expands only if at least three of 12 first call batches or two of 12 task outcomes differ. A failed gate stops the associated training or scaling run; it is not retuned after seeing the outcome.

Operationally, we (1) define the eligible population and endpoint, (2) separate development discovery, held-out support, and paired-leverage instances, (3) freeze the predicate, threshold, coupling, and

stop rule, (4) validate execution and model-facing traces before reading outcomes, and (5) report the decision together with repeats, invalid runs, uncertainty, and negative outcomes. A pass licenses the next experiment; it does not validate the proposed compressor.

4 Experimental Protocol

4.1 BFCL split and executable baseline

We use the Gorilla BFCL executable environment. Base and long-context variants are grouped by family. Within each 50-family domain block, we sort a seeded SHA256 hash, assign the first 10 families to development and the remaining 40 to held out. This yields 40 development and 160 held-out families. The CPU reference audit executes all 400 trajectories twice and checks state consistency. It covers 2,345 calls and is used only for structural analysis.

The model baseline uses Qwen3.6-27B (Qwen Team, 2026) in BF16 on two H100 GPUs with tensor parallelism two and native 262,144-token context. We preserve the model's thinking trace and use seed 20260925, temperature 1.0, top-p .95, and top-k 20. Sixteen development episodes and two exact repeats are completed. For every request we store the raw API response, rendered history, provider usage, tool call, raw tool result, simulator state, and original evaluator output. An independent checker reconstructs each message and dispatches only literal tool names. It then replays the original executor and evaluator before comparing repeated trajectories for semantic equality.

The array-summary control uses the same serving configuration and only the eight long-context development families. Transformation occurs after execution and before the result is appended to model history. Raw results remain available for score and environment replay. Two prespecified episodes are repeated exactly.

4.2 Cross-interface reference audits

For BFCL, the state-ledger audit snapshots public simulator attributes before and after each reference call and applies the typed subtree rule. The same typed matching rule is used in both reference repeats.

For auii, The official retail or airline environment is reset for every public task and each ordered reference action is executed once. Exceptions are retained. Successful responses receive exactly the same numeric-array and state-subtree tests as BFCL. These runs do not produce an official τ^2score because they audit response structure rather than agent reward.

4.3 Model-trajectory replay

Reference policies can differ from model behavior, so we additionally inspect a pre-existing 48-episode Qwen3-14B τ^2retail archive. The earlier experiment excluded two engineering tasks and deterministically shuffled the rest of the training split. Its first eight tasks were selected without outcome filtering, then crossed with two agent seeds and three user-simulator seeds. We neither select episodes by tool output nor regenerate them for this paper. Starting from each task's official initialization, we replay all model-generated calls in recorded tool-message order. Archived and fresh results are normalized symmetrically; errors remain in the ledger. For each successful exact replay, we compute structure and u_i from later assistant messages that contain model usage. Each replay remains linked to its task and message position, so repeated uses of an observation contribute to its reuse-weighted count without creating new observations.

4.4 Uncertainty and unit of analysis

Structural audits enumerate the fixed corpora and are reported as descriptive corpus rates. They do not receive binomial confidence intervals. Model success intervals use Wilson 95% intervals at the episode level. Paired success differences use the two-sided exact McNemar test. The numeric control's token saving interval uses 100,000 task-level bootstrap resamples with seed 20260912 and recomputes the ratio of token sums. Because those eight tasks are a selected development slice, the interval describes task sensitivity within that slice rather than benchmark generalization.

Table 2: State reconstruction under one typed-subtree definition. Rates describe fixed audited artifacts. Successful τ^2calls exclude preserved errors.

Interface and policy	Calls	JSON bytes	Reconstructible
BFCL reference, development	506	596,786	6.38%
BFCL reference, held out	1,839	2,099,090	8.53%
τ^2retail reference	532	493,735	100.00%
τ^2airline reference	142	85,261	100.00%
τ^2retail Qwen3-14B	376	338,065	100.00%

5 RESULTS

5.1 A STRONG LOCAL CONTROL WITH WEAK SUPPORT

The raw Qwen3.6 baseline succeeds on 10/16 episodes (62.5%; Wilson 95% interval 38.64–81.52%), split evenly as $5/8$ base and $5/8$ long-context. In long-context episodes, replay-weighted tool content accounts for 630,499 of 1,139,023 input tokens (55.35%). All 18 executions, 201 model requests, token recounts, direct tool replays, original score replays, and two semantic repeats validate.

The numeric summary succeeds on $6/8$ long-context episodes and preserves all five raw successes. Aggregate prompt-plus-completion tokens fall from 1,164,656 to 677,657, a 41.81% reduction. One raw failure becomes a success and no success becomes a failure. The exact two-sided McNemar calculation returns $p = 1.0$ from this single discordance, an uninformative value rather than evidence of equality. The task-bootstrap 95% interval for aggregate saving is $[-1.12, 66.33]$ percentage points. Two families actually expand after intervention, and the largest absolute saving comes from one observation containing 2,000 values. It passes the frozen engineering gate but does not establish non-inferiority or a population effect.

The held-out audit changes the conclusion. Seven qualifying arrays with 8,459 elements occur in the 16 development episodes. The selected transform saves 43.40% in a model-free history rewrite, compared with 6.58% from JSON minification alone. Across 320 base and long-context reference variants from all 160 held-out families, however, there are zero qualifying arrays. There is no held-out target for the proposed learned method.

5.2 THE COMPRESSION TARGET REVERSES ACROSS INTERFACES

BFCL held-out responses contain 2,226,798 bytes, 94.26% of them in JSON objects. The state ledger identifies 179,128 reconstructible partition bytes, 8.53% of JSON-object bytes, and 1,872,099 epistemic partition bytes (89.19%). There are 361 mixed calls. JSON coverage, epistemic mass, and mixed-call gates pass; the reconstructible mass falls below the frozen 10% threshold. The largest contribution is concentrated in `book_flight`. We therefore stop this direction on BFCL without training or a GPU prototype.

The same definition yields the opposite structural result on τ^2(Table 2, Figure 1). The public files contain 114 retail and 50 airline tasks; 112 and 43 have reference actions. Retail produces 550 actions, of which 532 succeed and 18 reproduce errors. Airline produces 142/142 successful actions. Neither domain contains a qualifying numeric array. Yet every top-level field in every successful JSON-object response exactly matches a typed subtree of the post-call database. JSON covers 99.71% of successful retail bytes and 85.99% of successful airline bytes, and its reconstructible share is 100% in both. No call mixes the two partitions.

5.3 THE τ^2 PATTERN UNDER MODEL-SELECTED CALLS

All 466 tool calls in the 48 saved Qwen3-14B retail episodes replay exactly, including 90 recorded errors. The 376 successful calls produce 339,072 canonical response bytes, including 338,065 JSON-object bytes. Again there are zero qualifying numeric arrays and 100% of JSON-object bytes are state-reconstructible.

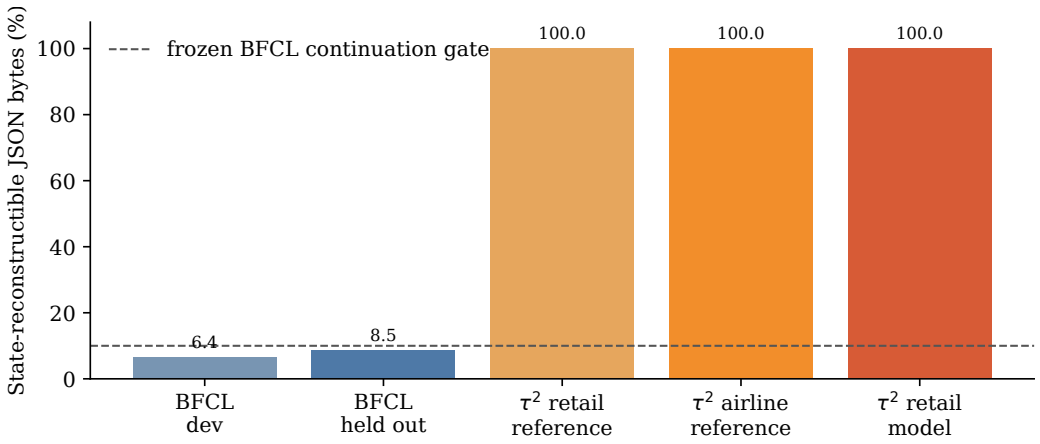

Figure 1: The state-backed opportunity is interface dependent. BFCL development and held-out reference responses remain below the frozen 10% continuation gate; all audited τ^2JSON-object bytes are reconstructible under the same syntactic test. Bars are corpus summaries, not sampling estimates.

Reuse changes the practical scale. Weighting each result by its number of later model requests produces 2,581,502 successful response bytes repeatedly presented to the model. State-reconstructible partitions contribute 2,570,945 bytes (99.59%); non-object epistemic results contribute 10,557 bytes (0.41%). Forty-eight successful calls mutate the database. Model-selected calls therefore expose the same structural target as the reference actions. The estimate remains an upper bound on representation mass. An actual state-reference codec must still pay for identifiers and reconstruction and demonstrate policy safety.

Tokenizer accounting puts this target in model-facing units. Successful JSON-object results contribute 1,216,414 reuse-weighted Qwen3-14B content tokens, 22.12% of 5,499,801 actual agent prompt tokens. This is a content-only attribution, not the exact marginal chat-template saving of a codec.

5.4 Paired leverage audit for exact redundancy

This bounded pilot demonstrates the paired leverage-audit design; its sample is not intended to establish a general behavioral effect. The BFCL reference audit finds exact state-preserving reobservation witnesses in 176/320 held-out variants and 100/160 families. In the corrected paired intervention, both arms first receive the same successful anchor result; one arm receives that exact result a second time. Forced calls are excluded from score. Across 12 greedy Qwen3.6 pairs, anchor and repeated arms score $7/12$ and $8/12$. Only one pair differs on both task outcome and first model-call batch. Its exact McNemar $p = 1.0$ has essentially no power to distinguish effects. The first-action change rate is 8.33%, with a Wilson interval of 1.49–35.39%. Later model actions change in eight pairs, showing that trajectory inequality is common even when measured task impact is small. The 12 paired comparisons comprise 24 unique arm–task episodes; four prespecified exact repeats bring the execution total to 28 episodes and 275 requests. Both frozen expansion counts fail, so we stop the direction under the registered project rule. The small sample and wide interval make the scientific result inconclusive rather than evidence that exact reobservation has no effect.

The "Advance" entry records the historical opportunity screen. The later natural-task expansion was cancelled after its baseline failed; it is not an outstanding successful evaluation or a deployment recommendation.

5.5 An implementable history-only control

Post-call database equality does not establish decoder access. To separate these conditions, we implement a lossless codec whose decoder can access only earlier model-visible messages. A reference includes a message index, JSON Pointer and JSON type; it cannot query the simulator database. Each new history entry is the actual serialized wire message, including any earlier reference

Table 3: The opportunity test separates local performance from the evidence needed to advance a candidate. "Advance" licenses an experiment, not deployment.

Candidate	Held-out support	Mass or paired leverage	Historical decision
Numeric arrays	0/160 BFCL families	41.81% saving on selected development tasks	Stop
BFCL state ledger	8.53% of JSON bytes	Below the frozen 10% mass gate	Stop
τ^2state reference	100% of JSON bytes	22.12% of actual prompt tokens; codec safety unknown	Advance
Exact reobservation	100/160 BFCL families	First action and outcome each change in 1/12 pairs	Inconclusive

encoding, rather than an implicitly reconstructed original object. Independent decoding checks recover the exact typed JSON value for every eligible observation.

The expanded reference audit includes 200 tasks each from base, long-context, missing-function and missing-parameter categories: 800 variants in 200 shared families and 4,625 calls. All 800 pass reference self-checks and repeated response and final-state equality. Of the calls, 4,365 contain JSON; the other 260 remain in the reported denominator and are ineligible. Only 40 calls across 17 families use a shorter reference encoding under either tokenizer.

For Qwen2.5-7B-Instruct, canonical original JSON totals 1,618,727 tokens and sent encodings total 1,604,180; for Llama-3.1-8B-Instruct the corresponding counts are 1,072,539 and 1,069,992. Net savings are 0.899% and 0.237%. Resampling the 200 families, preserving all four variants together (10,000 draws, seed 20260913), yields percentile 95% intervals of $[-1.251, 4.693]$ and $[-1.887, 3.627]$ percentage points. Relative to an artificial literal-envelope baseline, savings would instead be 2.511% and 2.666%; reporting only that comparison would conceal the envelope overhead. These are observation-serialization counts, excluding chat templates, decoder instructions, repeated-context use and latency. Neither syntax-preserving decoding nor these small aggregate savings establishes agent behavior preservation. This control therefore supports retaining a methodology claim and withholding a general compression-efficacy claim.

5.6 Explicit reconstruction at fixed prefixes

At all 40 eligible positions in 17 task families, each model receives a fixed history and must reproduce the complete typed JSON observation. Exact reconstruction falls from 32/40 to 15/40 for Qwen2.5-7B and from 21/40 to 2/40 for Llama-3.1-8B when a literal is replaced by a reference. The paired differences are -42.5 and -47.5 percentage points, with family-bootstrap 95% intervals $[-69.2, -14.7]$ and $[-69.8, -23.4]$. All 160 conditions remain in the denominator, including two context-limit rejections. The deterministic decoder succeeds on every case, so machine reconstructibility is insufficient for this model-facing reconstruction task. This copying diagnostic does not measure continuation likelihood or natural agent success; Appendix G specifies its protocol and limits.

5.7 Integrity requires information and can consume savings

We separately audit the existing literal-envelope codec on 800 BFCL and 48 archived τ^2 histories under Qwen2.5-7B and Llama-3.1-8B tokenizers. This retrospective fault-injection experiment is not a new agent rollout. All 4,686 JSON observations roundtrip correctly under each tokenizer. Structural checks detect every removed referenced message but miss all 133 and 128 same-type referent changes, respectively. A trusted fingerprint map or original-observation oracle detects all of these injected changes. The 121 and 119 unused-history controls remain valid; within-history faults are not independent trajectories.

Figure 2 charges the serialized pointer-to-SHA map at the tool-message level. On τ^2, its cost already exceeds the literal codec's savings under both tokenizers. A SHA fingerprint is not authentication: replacing both the referent and its claimed fingerprint defeats this condition. These counts still omit authentication, instructions and repeated request charging. Appendix F records the full counts; the separate full-template controls retain their original accounting and negative gate decision.

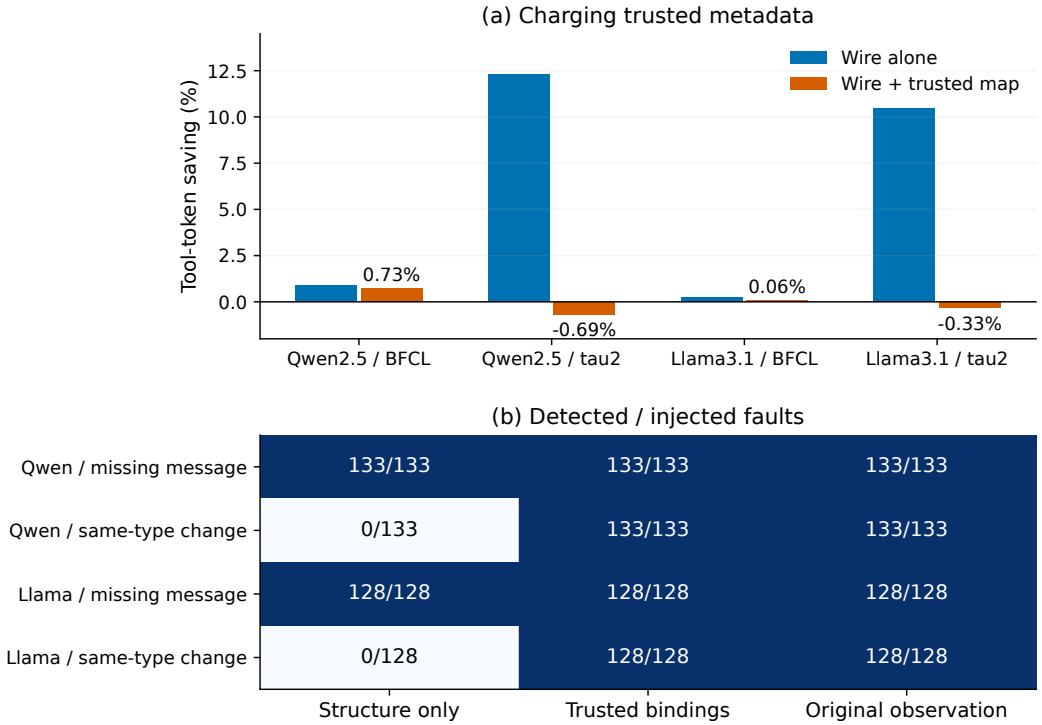

Figure 2: Information and cost in the archived-history audit. Top: tool-message savings before and after charging a trusted pointer-to-SHA map. Bottom: detected/injected counts, not natural failure rates. All 848 histories per tokenizer are retained. Metadata counts exclude authentication, instructions and repeated requests, and are not full-request savings.

6 Discussion

Opportunity belongs to an interface. The numeric transform appears compelling when viewed only through the selected BFCL development run: observations dominate input tokens, the transform is cheap, and the engineering gate passes. Its defining structure is absent from every held-out BFCL family and both τ^2domains. Conversely, a state ledger looks unsupported on BFCL but covers every audited JSON-object byte on τ^2, under both reference and model policies. This reversal leaves no single benchmark-agnostic account of "compressible tool output." The response schema, state exposure, call distribution, and reuse all affect which mechanism is plausible.

This result does not imply that $\tau^2$is defective: record-returning APIs are valid, and the benchmark targets dual-control interaction rather than compression realism. It does limit external validity to APIs that aggregate, transform, or conceal state.

Stopping before training. Training can obscure a missing precondition: a learned compressor may shorten cases where its motivating structure is rare, but the gain must then come from some other mechanism. A structural audit makes that shift visible and reserves expensive model evaluation for settings where the target exists. Here, two directions end after CPU audits and a third after a small paired intervention. The τ^2state-backed direction first requires a decoder with legitimate access to the referenced state; the history-only control in Section 5.5 demonstrates why structural coverage alone does not meet that requirement. The screen can err in both directions: a narrow predicate or underpowered pilot can stop a useful idea, whereas a permissive predicate, lucky slice, or surrogate endpoint can advance an unhelpful one. Frozen predicates and reported uncertainty make these decisions auditable. We have not calibrated false-stop or false-go rates; that would require prospective evaluation of candidates both accepted and rejected by the screen. A stop applies to the stated project rule, while a pass motivates the next experiment.

Structural redundancy is not behavioral redundancy. State equality, exact subtree occurrence, and repeated text answer different questions. A field already stored by the environment may still be necessary evidence for the model, while repeated text can shift token probabilities through recency or formatting without adding world information. Our reobservation study finds later trajectory changes in eight of 12 pairs, but first-action and success changes in only one pair; the success-effect estimate remains imprecise. A codec should therefore report at least application frequency, actual served token cost, next-action change, environment transitions, and final reward. No one measure is an equivalence oracle.

7 LIMITATIONS

The Qwen3.6 BFCL experiments use development families, and the array-control slice was selected for long observations. Its bootstrap interval does not repair this selection. BFCL held-out analyses use reference outputs, not held-out model scores. The τ^2model archive covers one Qwen3-14B retail configuration and was created for an earlier experiment; airline has reference actions only. Two exact repeats per BFCL model condition test pipeline determinism narrowly rather than estimating broad sampling variance.

Because the typed-subtree rule is syntactic, it can miss values recoverable by computation and count coincidental matches as reconstructible. Its top-level partition also misses finer nested decompositions. Canonical bytes support cross-tokenizer comparison but differ from served tokens and KV-cache memory. Errors are preserved but excluded from successful-response structure rates. The original BFCL checker is stricter than final-answer correctness, and successful reference execution does not establish semantic label quality.

The stopping gates encode engineering priorities rather than universal thresholds. Fixing them before the relevant result, and reporting the failures, prevents post-hoc expansion. Other deployments can choose different gates if they declare the cost model and decision rule in advance. The four audits were organized retrospectively under one test rather than executed as a single preregistered study, and we have not yet shown that a candidate advanced by the full test succeeds after scale-up. The small 12-pair reobservation experiment is a leverage diagnostic with wide uncertainty, not evidence of behavioral equivalence.

Natural-task pilot failure. All eight engineering episodes receive zero ENV reward, and the user simulator introduces unsupported identifiers or payment details. ENV scoring is itself only a subset of official task evaluation. These failures invalidate the pilot as a baseline for estimating a compression effect; a larger sample from the same protocol would not remove that confounding. The artifact retains all failures. The separate 160-condition reconstruction diagnostic measures exact recovery under its declared decoder information, not success on natural tasks. The evidence therefore supports a reconstruction audit, while a natural-task compression claim remains outside the scope of these results.

Reconstructible response bytes and complete-request savings have different denominators: instructions, schemas and repeated history remain. The charged-reference control measures this distinction, constraining the tested representation rather than every possible compression method.

8 CONCLUSION

A locally successful compression control is not enough evidence to train a general tool-observation compressor. In our executable BFCL study, a numeric summary saves 41.81% on eight selected tasks but has no target in 320 held-out variants. Under an identical state test, reconstructible JSON-object bytes occupy only 8.53% on held-out BFCL and 100% on τ^2retail and airline. The τ^2result persists across 466 exactly replayed model-selected calls and 99.59% of reuse-weighted successful response bytes. The practical result is the reversal: interface design and decoder information determine which compression mechanism has support. Separate audits of support, mass, leverage, and integrity provide grounds to stop unsupported work and identify where a compressor experiment is warranted. The expanded history-only control further shows that database reconstructibility and deployable token savings are distinct: net JSON savings are below 1% for both tokenizers, with intervals crossing zero.

Reproducibility Statement

The appendix specifies replay normalization, intervention semantics, and validation of environment effects. The supplementary material provides the evaluation protocols, analysis code and retained observations needed to reproduce the reported statistics and figures. Raw tool results remain separate from transformed model-visible observations.

AI Use Statement

OpenAI Codex was used to aid and polish manuscript writing and to assist with literature retrieval and discovery, including finding related work. The authors are responsible for the final manuscript and citations.

References

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. In *Proceedings of the International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64377.

Haorui Chen, Yuancheng Zhu, Yitong Zhang, and Jia Li. CoACT: Action-preserving observation compression for coding agents. *arXiv preprint arXiv:2607.02911*, 2026. URL https://arxiv.org/abs/2607.02911.

Xinyu Guan, Qianyang Zhao, and Yuming Deng. Decision-aware memory cards: Counterfactual-inspired context selection and compression for tool-using LLM agents. *arXiv preprint arXiv:2606.08151*, 2026. URL https://arxiv.org/abs/2606.08151.

Yinghan Hou and Zongyou Yang. Control under compression: Reliability frontiers for tool-using agents. *arXiv preprint arXiv:2608.01056*, 2026. URL https://arxiv.org/abs/2608.01056.

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

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

Seongyun Lee, Yongrae Jo, Minju Seo, Moontae Lee, and Minjoon Seo. Lost in the noise: How reasoning models fail with contextual distractors. *arXiv preprint arXiv:2601.07226*, 2026. URL https://arxiv.org/abs/2601.07226.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392. PMLR, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026. URL https://qwen.ai/blog?id=qwen3.6-27b.

Ruipeng Wang, Yuxin Chen, Yukai Wang, Chang Wu, Junfeng Fang, Xiaodong Cai, Qi Gu, Hui Su, An Zhang, Xiang Wang, Xunliang Cai, and Tat-Seng Chua. AgentNoiseBench: Benchmarking robustness of tool-using LLM agents under noisy condition. In *Proceedings of the International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64355.

Haoran Zhang and Zhaohua Sun. AGORA: Adapter-grounded observation-action retention for inference-free prompt compression in LLM agents. *arXiv preprint arXiv:2605.26596*, 2026. URL https://arxiv.org/abs/2605.26596.

YiShan Zheng, Yuan Wu, and Yi Chang. ToolRobustBench: Stage-wise perturbation evaluation and failure diagnosis for tool-calling agents. *arXiv preprint arXiv:2608.23635*, 2026. URL https://arxiv.org/abs/2608.23635.

A VALIDATION CONTRACT

For BFCL model runs, validation checks: requested sampling parameters; number of actual API responses; provider versus locally rendered token counts; literal safe tool dispatch; original executor replay; original score replay; message history reconstruction; and semantic repeat equality. Raw execution results are kept separate from transformed model-visible content, preventing compression from silently changing environment state or evaluator input.

For BFCL reference structure, both executions agree on all 2,345 results and their byte counts. For τ^2reference structure, failed actions are kept in the ledger and successful aggregate fields are recomputed independently. For the τ^2model archive, all 466 calls are matched by episode, task and message order. The analysis retains errors and recomputes byte counts and reuse weights from the corresponding observations.

B FAILURE AND STOP TAXONOMY

We distinguish four stop reasons. A *population failure* occurs when the target structure is absent on held-out families, as with BFCL numeric arrays. A *mass failure* occurs when structure exists but falls below its declared share, as with the BFCL state ledger. A *behavioral-effect failure* occurs when an intervention misses the expansion threshold, as with exact reobservation. A *measurement failure* occurs when the intended contrast is not represented by the implementation; such a result is corrected or discarded, never reinterpreted as evidence.

These categories are local to a candidate and interface. The BFCL state-ledger stop does not imply that state-backed compression is useless; the τ^2audit directly shows otherwise. Likewise, the array population failure does not refute array compression in workloads that routinely return traces. It refutes using this BFCL development slice as evidence for a held-out or cross-benchmark claim.

C REPLAY NORMALIZATION AND INTERVENTION SEMANTICS

The first τ^2model replay normalized archived string results by attempting JSON decoding but applied only type normalization to freshly replayed strings. This asymmetry falsely marked 25 `list_all_product_types` calls and five `calculate` calls as unequal. Manual grouping localized every mismatch to those methods; the database state was unchanged. The parser was changed so both sides are normalized and then JSON-decoded by the same function. The corrected normalization yields 466/466 exact replays. Only the corrected run enters tables and claims.

An earlier redundant-observation design compared a detour with no detour. Although the detour returned the same environment state, it could reveal information absent from the control arm. We replaced it with an anchor design: both arms receive the same successful result and only one repeats it. The earlier contrast is excluded from behavioral claims. These corrections illustrate why mismatch tables and explicit intervention semantics belong in an opportunity audit.

D VISIBLE-WIRE REFERENCES WITH FULL-TEMPLATE CHARGING

We recover the original 800 BFCL reference variants and the first 48 archived τ^2 episodes (the original audit subset, despite the archive containing 50 episodes). We compare raw JSON, order-preserving minification, the existing literal-envelope codec, and a reserved-marker codec that sends bare minified data when a reference does not shorten the payload. Reserved-marker collisions are explicitly escaped. Every decoded value must match typed JSON, and references resolve only into previous actually visible wire payloads, never a hidden decoded buffer. Message indices include non-JSON responses.

Token accounting uses the declared chat template at every subsequent assistant request and repeats the codec instruction on every request. The reference BFCL histories are reconstructed from questions, reference calls, and saved responses. The τ^2 archive does not establish the exact historical system message and tool schemas. Consequently these are counterfactual template costs, not a reconstruction of all historically billed tokens, an agent rollout, or evidence of unchanged success.

Tokenizer	BFCL saving vs. minification	τ^2 saving vs. minification
Qwen3-14B	-5.44%	$+0.59\%$
Llama-3.1-8B	-7.53%	-0.91%

Negative saving denotes increased input. Neither tokenizer/dataset pair reaches the frozen 5% net-savings gate. We therefore stop the proposed natural-task expansion. This negative result supports a disciplined compression audit, not a general performance claim for the codec.

Retrospective instruction-activation sensitivity. After observing the always-on costs above, we froze a supplementary CPU-only comparison that adds the instruction only after the first reference or reserved-marker escape, retaining it thereafter. The codec decisions and all four original cost totals reproduce exactly. Savings relative to minification become 0.57% [0.16, 1.18] and 0.62% [0.20, 1.19] for Qwen and Llama on BFCL, and 3.38% [2.48, 4.49] and 2.50% [1.62, 3.67] on τ^2. Brackets are descriptive 95% intervals from 10,000 task-family bootstrap replicates: 200 BFCL families and only eight τ^2 tasks. Deferring the instruction improves this implementation, but all point estimates remain below the original 5% gate. These results are explicitly retrospective and preserve the original negative always-on findings; they do not establish actual deployment savings or agent success.

E Additional evidence boundary

Completeness of an object archive is not completeness of a model request. Where the original system message, tool schemas, and reconstruction calls are unavailable, costs are counterfactual under the specified template, not recovered historical bills. Lossy compressors also require a separate preservation criterion and cannot inherit a lossless roundtrip guarantee.

F Information access in a retrospective integrity audit

We audit the same 848 archived histories (800 BFCL variants and 48 τ^2 episodes) using Qwen2.5-7B and Llama-3.1-8B tokenizers. This supplementary study uses the existing literal-envelope codec, not a new agent rollout or the reserved-marker cost comparison above. All 4,686 JSON observations per tokenizer roundtrip correctly. We distinguish structural checks, checks supplied with a trusted pointer-to-SHA binding, and an offline oracle supplied with the original observation.

There are 133 and 128 referenced observations, respectively. Removing a referenced visible message is detected in every injected case by all three checks. Changing a referenced value while preserving its JSON type is detected in 0/133 and 0/128 cases by structure alone, and in every case by trusted bindings or the original-observation oracle. The 121 and 119 controls that change unused history remain valid. These mutations do not estimate naturally occurring fault prevalence; related histories and within-history mutations are not independent trajectories.

Tokenizer	Data	Minified	Wire	Binding
qwen	bfcl	1,618,727	1,604,180	2,665
qwen	tau2	130,405	114,394	16,916
llama	bfcl	1,072,539	1,069,992	1,861
llama	tau2	95,391	85,416	10,291

Counts are tool-message tokens. Binding metadata alone exceeds the wire saving on τ^2 under both tokenizers. The counts omit authentication envelopes, instructions, and repeated-history charging, so they are not full-request costs. A hash is not authentication: the binding must come from a trusted source and cannot establish integrity if an attacker can also replace it. An offline oracle is not a deployable detector without access to the original observation. This audit therefore supports an explicit information-access boundary, not free integrity or preserved task success.

Table 4: Explicit reconstruction at all 40 reference-eligible positions. Differences are reference minus literal. Bootstrap intervals resample 17 families jointly across variants (10,000 draws, seed 260915). This is not agent task success.

	Qwen2.5-7B	Llama-3.1-8B
Literal exact reconstruction	32/40 (80.0%)	21/40 (52.5%)
Reference exact reconstruction	15/40 (37.5%)	2/40 (5.0%)
Difference (percentage points)	-42.5	-47.5
95% interval (percentage points)	$[-69.2, -14.7]$	$[-69.8, -23.4]$
Literal-only correct / reference-only correct	19 / 2	19 / 0
Both incorrect	6	19

G FIXED-PREFIX EXPLICIT RECONSTRUCTION DIAGNOSTIC

To test a narrower behavioral requirement than full agent success, we freeze all 40 positions where either tokenizer activates a history reference (17 BFCL families). For each position and each of Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct, we hold all preceding wire JSON messages fixed and replace only the final observation's literal envelope with its reference envelope. The system prompt explains the codec and asks for the complete decoded JSON, using only visible history. The reference policy supplies these histories; this is not a trajectory sampled from either model. Both models receive the same 40 selected positions, chosen before generation.

All 160 scheduled condition evaluations are retained under the frozen protocol: 158 generate outputs and two are rejected by the context check. We use greedy decoding, at most 4,096 output tokens and 32,768 total context tokens, with no silent input truncation. A request exceeding that context budget is retained as incorrect. JSON must parse strictly and equal the complete typed original value, including exact string content; we do not remove Markdown wrappers or repair Python literals. Independent synthetic engineering checks verify execution and logging, not accuracy, and their model failures do not trigger prompt retuning.

The deterministic decoder reconstructs all selected JSON values exactly, but model reconstruction deteriorates under references (Table 4). Each model has one literal-arm context rejection and none under references; these cases remain in the primary denominator. Each reference arm has one output-length termination. Some literal cases also fail, so the diagnostic separates a paired representation effect from baseline copying/formatting difficulty. It does not prove that every failure is a reference-resolution error or that a natural next action would change. Only 17 of the 200 audited families activate references, and the explicit reconstruction instruction and output-copying requirement limit extrapolation. These results strengthen the distinction between deterministic reconstructibility and model usability, while leaving the full agent-success preservation question unresolved.

H EXTENDED RELATED WORK

Prompt and agent-context compression. General prompt compressors use token importance and instruction-tuned reconstruction to shorten long inputs (Jiang et al., 2023; 2024). Agent histories impose a stricter contract because punctuation, identifiers, action syntax, and environment feedback can determine executability. AGORA reports that generic token extraction can destroy action grammar and instead protects step-level structure (Zhang & Sun, 2026). CICL ranks context units by their expected effect on action rather than semantic similarity (Guan et al., 2026). CompressAgent measures reliability frontiers for static agent control contexts and finds strong variation across contexts and retained budgets (Hou & Yang, 2026). We examine dynamic post-execution observations and test whether a proposed mechanism has a held-out target before training.

Action preservation and robustness. CoACT trains observation compressors using a next-action-preservation signal and length reward for coding agents (Chen et al., 2026). This makes the next action a useful local contract, but local agreement does not by itself establish final reward or benchmark-wide applicability. NoisyBench, AgentNoiseBench, and ToolRobustBench show that irrelevant histories and tool-output perturbations can substantially alter agent behavior (Lee et al., 2026; Wang et al., 2026; Zheng et al., 2026). Our redundant-observation control is different: both arms contain the same

executed information and reach the same environment state; only one arm repeats the exact result. We measure first action, full trajectory, and task outcome separately.

Executable agent benchmarks. BFCL includes executable multi-turn API suites and evaluates sustained function calling (Patil et al., 2025). τ^2-Bench evaluates conversational agents in stateful retail, airline, and dual-control settings (Barres et al., 2026). Their tool interfaces expose different relationships between returned text and environment state. Rather than pooling their outcomes into one compression rate, we use the contrast as the object of study.

I SCOPE OF DEPLOYMENT INFERENCE

The 99.59% reuse-weighted share measures access to state-backed content under the replay contract. It does not measure the token cost of communicating a typed reference, retrieving its value, or executing subsequent tool calls. Raw observations, minified JSON, textual summaries and materialized references have different decoding and information requirements. A deployment comparison must charge these costs under a common task and evaluator. The present audit identifies that distinction; it does not convert reconstructible bytes into measured savings or downstream task success.