How Do Tool Surfaces Shape Inference Trajectories in LLM Agents?

Anonymous authors
Paper under double-blind review

Abstract

Tool surfaces, the tools exposed to an agent at each model call, define its available actions and can shape its observable inference trajectory before any tool executes. We investigate their immediate effects on generated reasoning, propagation across calls, and value for execution efficiency. We introduce Tool-Surface-Controlled Trajectory Dynamics (TSCTD), a state-conditioned theoretical framework that distinguishes immediate shaping, historical effects after withdrawal, and effects of continued exposure. Controlled experiments reveal immediate changes in generated reasoning and setting-dependent behavioral carryover after tool withdrawal. They further show that explicit task-closing instructions reduce the additional token cost of continued tool exposure. Building on these findings, we develop a state-dependent configuration policy that uses a compact tool set during execution and withdraws tools once public tests pass. We evaluate this policy on programming problems from the Mostly Basic Python Problems (MBPP) benchmark, adapted into interactive coding and testing tasks. Compared with keeping all tools available throughout execution, the policy reduces aggregate token usage across tasks by 34.74%, while meeting a prespecified success noninferiority criterion. This work establishes tool exposure as a design variable for agent inference and execution and provides an experimental framework for studying when to show, retain, and withdraw tools.

1 Introduction

Large language model (LLM) agents use tools to inspect information, modify their environments, and verify the outcomes of their actions. By interleaving generation with tool execution, they can solve tasks that require interaction beyond a single response (Yao et al., 2023; Schick et al., 2023; Yang et al., 2024). At each model call, however, the agent encounters tools before it executes them: their names, descriptions, and argument schemas enter the context from which it generates its next response. We call this collection of exposed tools the *tool surface*. A tool surface defines available actions and supplies information that can shape how the agent approaches its task. Understanding this dual role is essential to designing agents whose reasoning and execution remain focused as their available capabilities expand.

Tool selection and interface design already play central roles in agent systems. Prior work studies how agents retrieve relevant tools, choose among competing interfaces, and filter tools according to task state (Qin et al., 2024; Patil et al., 2024; Faghih et al., 2025; Blankenstein et al., 2026; Babu & Iyer, 2026). These approaches motivate a broader question: how does the exposed tool set influence the trajectory through which an agent reaches its decisions? For example, adding a search tool could change the agent's preference between two editing tools that were already available. Exposure during one call could also affect subsequent actions through the text, execution feedback, or workspace changes retained from that interaction. Near task completion, keeping tools available could influence whether the agent continues working or produces its final response. These possibilities make the timing of exposure, retention, and withdrawal consequential design choices.

We organize this investigation around three questions. **RQ1: How does the current tool surface shape generated reasoning and action choice before any tool feedback? RQ2: Under what conditions do the effects of earlier exposure propagate across calls after the surface changes? RQ3: Can active tool-surface configuration improve execution efficiency under a task-success**

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

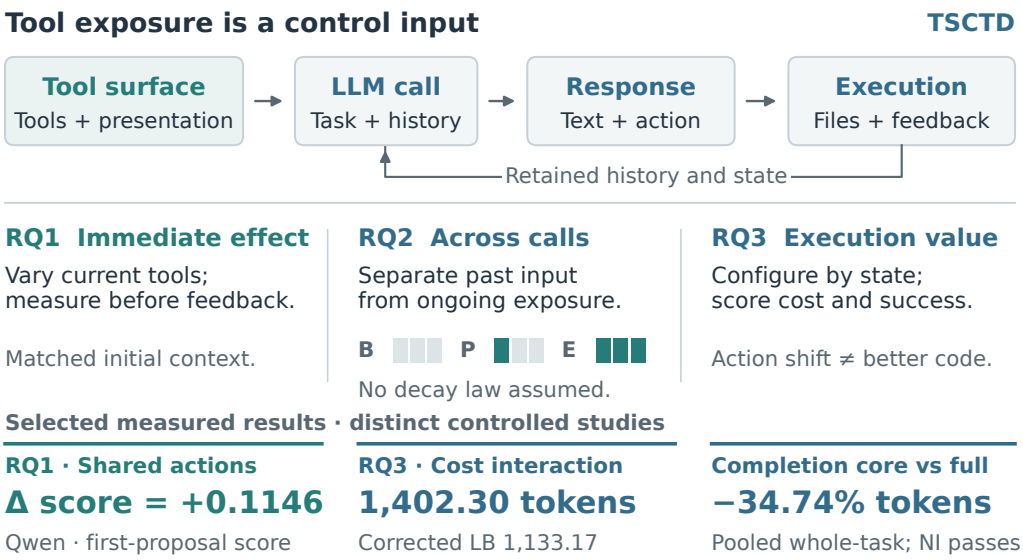

Figure 1: **Tool exposure is an intervention on the agent loop.** Baseline (B), pulse (P), and sustained (E) blocks depict common (gray) and expanded (teal) exposure schedules, not measured response curves. The three empirical summaries concern distinct Qwen studies: shared-action proposal scores, an instruction-by-exposure cost interaction, and pooled whole-task savings against full exposure. LB denotes a corrected one-sided confidence lower bound; NI denotes the frozen -5 percentage-point success noninferiority criterion. These findings do not identify one mediation chain.

constraint? Together, these questions connect an immediate generation effect to its possible downstream consequences and practical value. We use *observable inference trajectories* to denote the reasoning or planning text returned by the model and its associated action sequences.

To connect these questions, we introduce **Tool-Surface-Controlled Trajectory Dynamics (TSCTD)**, a state-conditioned theoretical framework that treats tool exposure as a controllable input to the agent's interaction loop. TSCTD distinguishes immediate shaping by the current surface, historical effects after withdrawal, and effects maintained by continued exposure. It also separates available capabilities, their presentation, and the interaction state carried into later calls. This formulation gives each claim a distinct experimental target. We compare responses before feedback, branch continuations from shared interaction states, and contrast initial exposure with continued exposure. The framework connects these interventions to observable text, action sequences, and executable outcomes (Figure 1).

Our experiments reveal several consequences of this view. Changing the surface can alter generated reasoning before a tool returns any feedback. In one evaluated setting, additional tools redirect choices between interfaces that remain available in both conditions, showing that action removal is unnecessary for such a shift. Exploratory withdrawal experiments also identify localized action carryover. At matched task states, an explicit instruction to finish the task reduces the additional token cost of continued tool exposure. A fixed first-request tool-input charge cancels from this interaction, so that charge alone cannot explain the result. These findings connect exposure effects to the agent's current context and the conditions under which it continues or ends its work.

Building on this framework, we implement a dynamic configuration policy that maintains a compact tool set during execution and withdraws tools once public tests pass, while allowing a final response. We evaluate it on programming problems from the Mostly Basic Python Problems (MBPP) benchmark (Austin et al., 2021), adapted into interactive tasks requiring code editing and test execution. Compared with keeping all tools available throughout execution, the policy reduces aggregate token usage across tasks by **34.74%**, while meeting a prespecified success noninferiority criterion. This result demonstrates the practical value of controlling exposure relative to continuous full-tool availability.

Our contributions are:

- **Tool exposure as a source of trajectory change.** Controlled interventions reveal pre-feedback changes in observable reasoning and choices among shared actions, together with setting-dependent historical action effects.
- **A framework for separating exposure effects.** TSCTD organizes immediate shaping, historical carryover, and continued exposure into distinct intervention targets, connecting trajectory measurements to execution outcomes.
- **Evidence for context-dependent exposure costs.** A prospective instruction intervention changes the cost of retaining tools at matched states, excluding a fixed first-request surcharge as the sole explanation for the interaction.
- **A practical exposure-control policy.** A compact execution surface combined with state-dependent withdrawal yields validated token savings against full exposure under a prespecified task-success constraint.

Together, these contributions establish tool exposure as a design variable for agent inference and execution. They provide a basis for studying when tools should be shown, retained, and withdrawn as an agent progresses through a task.

Additional related work appears in Appendix V.

2 TSCTD: A STATE-CONDITIONED ACCOUNT

2.1 THE OBSERVABLE INTERACTION LOOP

Write the tool surface as $S_t = (\mathcal{U}_t, \Pi_t)$: $\mathcal{U}_t$ specifies available capabilities and their implementations, and Π_t their presentation, including descriptions and order. Let H_t be model-visible history, W_t the workspace, and C_t requirements and control context. Visible text V_t and proposed actions A_t lead to execution feedback R_t:

$$(V_t, A_t) \sim P_\theta(\cdot \mid S_t, H_t, C_t), \tag{1}$$

$$(W_{t+1}, R_t) = E(W_t, A_t; \mathcal{U}_t, C_t), \qquad H_{t+1} = \mathrm{Append}(H_t, V_t, A_t, R_t). \tag{2}$$

Proposals may name unavailable tools; E accepts or rejects them using $\mathcal{U}_t$. Workspace contents reach the model through supplied materials or feedback. Let Y_t collect observable response, action, and execution records, with fixed representation $\phi(Y_t)$. Earlier exposure can affect later calls through retained text, actions, files, or feedback. This loop does not identify a latent reasoning state or response law.

2.2 IMMEDIATE, HISTORICAL, AND SUSTAINED EFFECTS

L1 changes the current surface from the same context and measures the response before feedback. **L2** measures downstream consequences of initial exposure after future surfaces become common. **L3** compares continuing exposure regimes. For a shared starting context c, let baseline B use common surface M throughout, pulse P use expanded surface S only at $h = 0$, and sustained regime E use S throughout (Figure 3). Define

$$\mu_h^a(c) = \mathbb{E}[\phi(Y_h) \mid do(a), c], \qquad D_h^a(c) = \|\mu_h^a(c) - \mu_h^B(c)\|_w^2, \quad a \in \{P, E\}. \tag{3}$$

This target separates changes in mean observables from sampling variability. Negative finite-sample estimates remain in the analysis; the estimators and earlier frozen readouts are specified in Appendix B.

2.3 CURRENT EXPOSURE AND RETAINED STATE

For a fixed next-response readout ψ, such as indicators of proposed inspection, modification, and finishing, the current-exposure contrast at matched history and context is

$$\begin{aligned}\boldsymbol{\tau}_t(s, s'; h, c) = \mathbb{E}[\psi(V_t, A_t) \mid do(S_t = s), H_t = h, C_t = c] \\ - \mathbb{E}[\psi(V_t, A_t) \mid do(S_t = s'), H_t = h, C_t = c].\end{aligned} \tag{4}$$

When $\mathcal{U}(s) = \mathcal{U}(s')$, this contrast changes presentation without changing executable capability. TSCTD imposes no common sign across action classes or states. A changed next-action distribution need not improve execution. Contrasts involving actual code outcomes additionally match workspace and execution rules.

Let X_t contain the retained history, workspace, and other interaction state. Under fixed-model calls without additional unrecorded memory, earlier exposure influences later responses through retained state supplied by the interaction loop. Matching current surfaces or workspace bytes alone need not match X_t. Decay requires further assumptions on transitions and readouts; full-history retention and changing task demands need not satisfy them. Appendix B.6 states these assumptions, and Appendix B retains the conditional contraction result. Neither a detected action shift nor a historical effect by itself establishes policy value.

3 Experimental setup

Models and tasks. The studies use DeepSeek V4 Pro and Flash (DeepSeek-AI, 2026), Qwen3.6-35B-A3B (Qwen Team, 2026), and GLM-5.3 (GLM-5 Team, 2026; Z.ai, 2026), abbreviated Pro, Flash, Qwen, and GLM. Request settings are fixed within each comparison; recorded model identifiers appear in Appendix C. Tasks comprise controlled algorithm problems, generated editing problems, and interactive adaptations of Mostly Basic Python Problems (MBPP) (Austin et al., 2021). Appendix A maps each study to its sample and evidence status.

Executable agent. Programming episodes run in isolated workspaces. At each call, the agent receives the current task, retained history, and declared tools. Core actions read files, replace a string, write a file, and run public tests; expanded inventories add file search, text search, and planning. Proposals follow each study's frozen admission and stopping rules; actual feedback enters the next call. Unavailable or invalid calls are rejected, and edits invalidate verification where the study tracks it. Public tests guide behavior; private tests score outcomes. Immediate studies measure the first response before new feedback; the description-order study executes one action locally without another model call.

Controls and measurements. Paired surface interventions preserve prompts and starting state. Common-prefix experiments clone retained history and workspace bytes before changing exposure; their continuations then evolve independently. Baseline/pulse/sustained comparisons separate ongoing input from earlier exposure. Frozen text decoders measure distinguishability, tool proposals measure action choice, and code tests plus input/output tokens measure execution outcomes. Task-level analysis preserves correlated repetitions and branches. Primary tests, directions, and within-study corrections remain as frozen; descriptive and post hoc analyses are identified separately. Surface inventories differ across studies, so the labels M and S always refer to the local definitions in Appendix C.

4 RQ1: immediate shaping before tool feedback

4.1 Additional tools change choices among shared actions

The strongest action-choice control keeps both editing interfaces available. On 24 MBPP tasks, we compare common read/edit/test/write tools with an expanded inventory that adds glob, grep, and planning. Shared tool definitions and relative order are unchanged. Two constructed code states and two repetitions per task yield 576 evaluation responses across Flash, Qwen, and GLM. Tasks are project-new for the first two models and reused for GLM. Paired branches receive identical task and code state, with no new feedback before the measured response.

We freeze the directional prediction and analysis before each model's collection. The first-proposal score is $+1$ for replacement editing, -1 for write, and zero otherwise, including no tool. Invalid parameters retain the named-interface score. Table 1 reports the task-level tests. Among 96 responses per surface, Qwen's write proposals change from 7 to 3 and replacement-editor proposals from 45 to 52 (Appendix L).

Finding 1: adding tools can redirect choice among actions available in both conditions. The Qwen result supports the prespecified direction on new project tasks (Holm $p = .04180$; Flash:

Table 1: Shared-interface results on the same 24 tasks, 192 responses per model. Both interfaces remain available. Intervals are descriptive and unadjusted; p values are nominal, one-sided. The original Flash/Qwen Holm tests are reported below.

Model	Expanded minus common	95% interval	Nominal p
Flash	$+0.0208$	$[-0.0549, 0.0966]$	0.38675
Qwen	$+0.1146$	$[0.0162, 0.2129]$	0.02090
GLM	-0.0104	$[-0.0956, 0.0747]$	0.69255

.38675). It cannot be explained by removing write or by feedback from a newly executed action. Tool count, descriptions, and schema length still change together. The endpoint identifies interface choice, not edit locality: a replacement editor can replace the entire old file. GLM's estimate is close to zero, with an interval spanning both signs. These model-specific results do not establish a between-model interaction.

4.2 Matched capabilities reveal a reversed description-order prediction

A stricter Qwen study holds all seven tools, names, schemas, executable capabilities, and their order fixed. Three identical facts about `grep` output are arranged with line information first (L) or file information first (G). Each description is 171 bytes with the same words and counts. TL/TG vary tool metadata while ordinary reference text uses a fixed third ordering; CL/CG vary the ordinary text while tool metadata uses that ordering. Every four-arm block has identical reported input tokens.

The evaluation uses 128 new parameterized repair tasks from four fixed families, two repetitions, and four arms: 1,024 first responses. Each starts from a faulty implementation with code, specifications, and a public failure supplied. The first action is executed without returning feedback. Success requires an actual edit, all tests passing, and unchanged AST outside the target function. Whole-file writing can satisfy this endpoint. The frozen prediction requires positive task-level means for $D_T = \mathrm{TL} - \mathrm{TG}$ and $I = D_T - (\mathrm{CL} - \mathrm{CG})$.

In metadata, L/G repairs succeed on 207/256 and 241/256 responses (80.86%/94.14%); the corresponding ordinary-text arms succeed on 220/256 and 224/256 (85.94%/87.50%).

Both predictions fail: observed $D_T = -13.28$ pp and $I = -11.72$ pp, with Bonferroni-adjusted one-sided $p = 1$ for each. The descriptive ordering reverses the predicted direction. All 893 edits use replacement editing; 892 pass, while 131 first actions inspect or test without editing. All 1,024 responses preserve non-target AST. The result concerns first-action inspection versus modification, not a shift between editing interfaces, identified locality, or whole-episode quality. It motivates revising the directional account; it is not a successful prior prediction (Appendix M).

4.3 Controlled surfaces also leave observable text differences

Three original Flash studies vary surface with prompt (160 responses), composition at fixed callable count (80), and tool richness (160). Saved predictions yield surface balanced accuracy 0.9000 in the first two. Paired requests isolate the tools field; the fixed-count study shows that callable count alone does not determine the text difference. Appendix C retains the original readout definitions.

A Pro factorial study crosses surface, tone, and deliberation strategy over 24 algorithm tasks and four repetitions (768 first responses). Character TF-IDF and logistic decoders train on 12 tasks and freeze before evaluating 12 others. The primary view masks exact intervention echoes. Minimal/Standard surfaces contain two/nine tools; tone changes *Solve* to *Please solve*, and strategy changes *Think quickly* to *Think carefully*.

Decoder information is log-loss improvement over equal-probability prediction. Surface minus tone is $+0.030481$ nats, with an individual approximate 97.5% task-bootstrap interval $[+0.018735, +0.043595]$, positive on all 12 evaluation tasks. Strategy minus surface is -0.001420, interval $[-0.022125, +0.019458]$. Thus the selected surface contrast exceeds this tone contrast, while the strategy–surface ordering remains unresolved. These intervals condition on frozen decoders and retain seven evaluation truncations.

4.4 A COMPLETE-EXECUTION TEST SEPARATES ACTION DIRECTION FROM UTILITY

We next freeze two predictions for new Qwen tasks: line-first presentation increases initial inspection, and its success advantage is larger when relevant facts must be retrieved than when they are supplied. Four arms cross supplied/missing facts (S/M) with L/G. Only the first call changes presentation; subsequent calls use a common surface within six responses. The formal sample comprises 192 tasks from four families and one trajectory per arm (768 total).

Both predictions remain unsupported. The average inspection-rate difference is -2.60 pp (corrected one-sided lower bound -7.69), and the success interaction $(ML - MG) - (SL - SG)$ is -1.04 pp (lower bound -7.72); Bonferroni-adjusted $p = 1$ for each. With facts missing, both presentations inspect first on 188/192 trajectories; with facts supplied, the counts are 118/192 and 128/192. These descriptive state differences do not rescue the failed predictions. Final successes are 180/176/172/170 in SL/SG/ML/MG. The result limits the proposed transfer from first-action tendencies to complete-task utility. Development cohorts and feasibility decisions are reported separately in Appendix T.3.

5 RQ2: CROSS-CALL EFFECTS AND THEIR PREDICTIVE LIMITS

5.1 WITHDRAWAL REVEALS LOCALIZED HISTORICAL ACTION EFFECTS

Three Pro, Flash, and Qwen campaigns each collect 1,184 responses on the same eight tasks. Sixty-four common prefixes each generate four randomly assigned branches: two receive M and two S at $h = 0$, then all receive M at $h = 1$–3. Here M contains read, replacement editing, and testing; S adds glob, grep, and write. The endpoint is any later attempt to invoke a removed tool. Prefixes can already contain expanded exposure, so the contrast concerns the additional $h = 0$ intervention.

Pro and Flash show attempt-rate differences of $+17.19$ and $+21.88$ percentage points; Qwen's is -1.56 points (Table 10). These are exploratory because of recovery deviations. Separate demand-timing studies use Pro (1,616 responses) and GLM (224 responses). Pro's prespecified timing interaction is $+7.8125$ percentage points (SE 3.125), with exploratory approximate 95% interval $[+1.6875, +13.9375]$; GLM records no removed-tool attempts across its 64 branches. The designs differ in task construction and timing conditions (Appendices D, H). Requirements, cumulative checks, and verification state change together, so the Pro contrast does not identify a semantic trigger or decay rate.

Figure 18 in the appendix summarizes these historical action effects.

5.2 A PROSPECTIVE DEMAND-CONDITIONED EXTENSION DOES NOT PASS

A frozen follow-up uses 64 new generated tasks from four fixed families and Flash/Qwen. Eight probes per task–model cross earlier M/S history, immediate/delayed observation, and a new edit demand/verification of the old specification. Delay consists of two read-only background checks. Every target probe starts from the same canonical old code and public tests, cleared external read/test state, and the common four-tool surface; native histories remain different.

The readout is the same replacement-edit/write/other first-proposal score. Let $D_{\ell,d}$ be the S-minus-M difference at interval ℓ and demand d. Frozen predictions require positive task means for

$$C_E = D_{\text{immediate,edit}} - D_{\text{immediate,verify}}, \qquad I = C_E - C_L, \tag{5}$$

where C_L uses the delayed interval. Both endpoints must pass per model, with Bonferroni correction across four one-sided task tests. All 1,024 formal probes complete, but none of the four tests passes (Appendix Q).

Qwen's immediate/delayed edit differences are descriptively $+0.28125/-0.046875$, not confirmed decay. This follow-up changes tasks, readout, and external-state control relative to the Pro timing study. It limits the proposed directional generalization rather than exactly replicating or negating the earlier localized effect (Appendix Q).

5.3 BROADER TEMPORAL AND KERNEL PREDICTIONS REMAIN UNSUPPORTED

Several complementary tests constrain the propagation account. A Qwen long-window study (4,416 responses, eight tasks, $h = 0$–7) passes none of three corrected tests in its frozen 279-dimensional representation (Figure 6). An automatic baseline/pulse/sustained study on 48 new three-task bundles, two repetitions, and Flash/Qwen supplies 4,118 responses plus 298 predetermined absorbing positions; none of eight tests in its frozen sixteen-dimensional readout passes. A Pro kernel study freezes predictions before 1,024 test responses: LTI, matched-current FIR, and LTV increase loss by 12.53%, 22.65%, and 61.50% over the current-time baseline, and none of six comparisons passes. A separate history-by-workspace study also fails its three incremental tests (Appendix D).

Text readouts do not recover a general withdrawal signal: ten corrected comparisons over 7,392 reused responses do not pass. A separate Qwen sustained-versus-pulse text comparison is exploratory ($+0.005512$ nats, Holm $p = .040$, but a descriptive task-cluster interval crossing zero); pulse-versus-baseline tests do not pass (Appendices I, R). Collectively, the data support separating current exposure from history; they do not establish a general persistence-and-decay sequence or reliable response kernel.

6 RQ3: INSTRUCTION-DEPENDENT EXPOSURE COSTS

6.1 STATE-DEPENDENT CONFIGURATION ON COMPLETE TASKS

We first test `completion_core`, a public-state rule. Let $\mathcal{C}$ contain read, replacement editing, testing, and write, and let $g(z_t) = 1$ when the current code has passed all public checks:

$$S_t^{\text{completion}} = \begin{cases} \mathcal{C}, & g(z_t) = 0, \\ \varnothing, & g(z_t) = 1. \end{cases} \tag{6}$$

Withdrawal still permits a normal closing response. The rule uses public state, without a fitted response kernel or history predictor. Full exposes seven tools throughout; static core always exposes $\mathcal{C}$.

The Qwen evaluation adapts MBPP to single-ticket editing and testing: four development and 192 evaluation tasks exclude 442 previously used project IDs. Each task has one public and two private assertions; success requires all three, within at most eight responses. All 576 evaluation trajectories complete. Completion core succeeds on 167/192 tasks, versus full's 162 and core's 164.

Against full, pooled whole-task tokens decrease by **34.74%**. Mean paired saving is 3,086.1 tokens (corrected lower bound 2,110.9); the success-difference lower bound is -3.28 pp. This meets the frozen cost, 10% pooled-saving, and -5 pp noninferiority requirements. Against static core, saving is 23.05%, but the success lower bound is -5.76 pp, so the joint criterion does not pass. The four one-sided endpoints use Bonferroni $\alpha = .0125$ (Appendix K). Earlier retention policies that failed their joint criteria remain in Appendix J.

6.2 EXPLICIT CLOSING ATTENUATES THE EXPOSURE COST DIFFERENCE

To isolate exposure at the same execution state, an initial experiment clones all 188 static-core prefixes with a valid first public pass before the response budget expires. Retain/withdraw branches share code, native history, feedback, and remaining budget. Withdrawal saves 67.30% of pooled continuation tokens and meets its code-test noninferiority criterion (Appendix K). The original system prompt already asks for a response without tools when work is complete, but does not equate public pass with completion.

We then predict that a more explicit closing instruction will attenuate the cost difference. Before new collection, we freeze a four-arm experiment on those same 188 states, crossing retain/withdraw with original/explicit-closing instructions. The added instruction is: *End this ticket now with a final response. Do not call any tools or modify any files.* It imposes no length requirement. All 752 formal suffixes are newly sampled in randomized adjacent order; the old suffixes are not controls. At a fixed instruction, the first requests differ only in tools. This is a prospective response test on existing prefixes.

Let T_{iqs} be total suffix tokens for prefix i, instruction $q \in \{O, C\}$ (original/closing), and surface $s \in \{R, W\}$ (retained/withdrawn). The frozen interaction is

$$I_T = \mathbb{E}_i\big[(T_{iOR} - T_{iOW}) - (T_{iCR} - T_{iCW})\big] > 0. \tag{7}$$

The mean exposure difference falls from 1,812.15 tokens under the original instruction to 409.85 under explicit closing. The interaction is **1,402.30 tokens**, with corrected lower bound **1,133.17**. Under closing, withdrawal saves **28.62% of pooled continuation tokens**, with cost lower bound 292.95 tokens and success-difference lower bound -2.15 pp. Every arm succeeds on the same 163/188 tasks. The cost–noninferiority conjunction and attenuation interaction pass their three prespecified one-sided tests with Bonferroni $\alpha = .05/3$ (Appendix O).

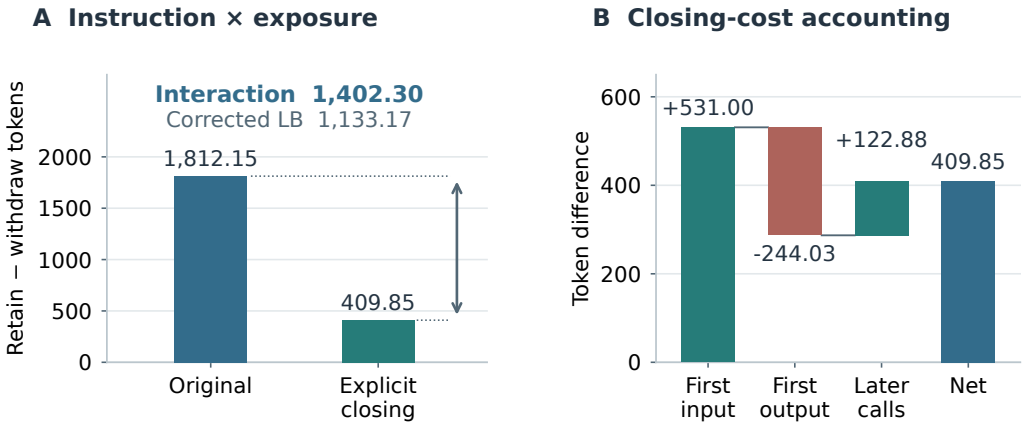

Figure 2: **Closing instructions attenuate the cost of retaining tools.** (A) Mean retain-minus-withdraw suffix cost at 188 matched public-pass states; LB is the corrected one-sided lower bound for their interaction. (B) Explicit-closing accounting: first input, first output, and later calls sum to 409.85 tokens. The first-input difference is exactly 531 for every prefix under both instructions and cancels from the interaction. All four arms pass the same 163 tasks. This excludes a fixed first-request surcharge as the entire explanation, not all length effects.

6.3 A FIXED FIRST-REQUEST SURCHARGE CANNOT EXPLAIN THE INTERACTION

Finding 2: the cost of tool exposure depends on the closing instruction. A simple alternative attributes the entire exposure effect to the fixed input cost of tool declarations. Exact accounting shows a first-request retain-minus-withdraw input difference of 531 tokens for *every prefix under both instructions*. Subtracting this charge therefore leaves each interaction unchanged:

$$\begin{aligned}&[(T_{iOR} - T_{iOW}) - 531] - [(T_{iCR} - T_{iCW}) - 531] \\ &\quad = (T_{iOR} - T_{iOW}) - (T_{iCR} - T_{iCW}).\end{aligned} \tag{8}$$

The interaction survives this exact cancellation without a new statistical test. Across the 188 states, its total 263,632 tokens comprise zero first-input interaction, 31,591 first-output interaction, and 232,041 later-response interaction. Thus an instruction-independent fixed first-request surcharge cannot be the complete explanation. The directional interaction was prespecified; this accounting argument is a subsequent interpretation of that result.

Action replay clarifies the continuation. Retention produces 181 tool proposals with the original instruction and 16 across 11/188 prefixes with explicit closing. The latter include eight first reads of the current code version, seven tests of unchanged passing code, and one same-content write. None changes code. Under closing, withdrawal reduces input by 121,262 tokens but increases output by 44,210, yielding the net 77,052-token saving (Appendix P). The result excludes a fixed additive accounting explanation, not all length-dependent generation mechanisms. It also does not identify a positive behavioral saving after deducting that charge or a reduction in internal reasoning.

6.4 DIRECT STOPPING BOUNDS THE EXECUTION CLAIM

Retrospective truncation at first valid public pass preserves all original code-test labels across 576 complete trajectories. At the 188 shared states, direct stopping preserves the same 163 successes with no suffix calls or tokens. It omits a final model response, whose utility we did not score (Appendix N). The gains therefore concern the specified continuation baselines; they do not establish superiority to direct stopping or a success benefit from dynamic tool control.

7 DISCUSSION

What the controls establish. The shared-interface and closing experiments retain their distinct positive prediction results. Their controls exclude forced interface substitution and a fixed first-input surcharge, respectively. The description-order study adds a boundary condition: with capabilities and input length matched, its first-action repair ordering reverses the predicted direction. This observed reversal motivates revision; it does not retrospectively validate the failed prediction.

What revised TSCTD contributes. The framework defines contrasts conditional on presentation, retained state, and task context. A stable mapping from a description's semantic emphasis to an action direction, repair quality, or universal decay is not part of the revised account. The closing interaction supports instruction-conditioned cost, while historical action evidence remains localized and exploratory. RQ1–RQ3 do not form an identified mediation chain, and failed tests do not establish zero effect.

Action tendencies do not substitute for task value. The completed information-supply experiment separates a first-call presentation intervention from its later consequences under a common policy. Neither its predicted inspection direction nor its quality interaction passes. The proposed constant log-odds shift remains an unfitted candidate (Appendix B.6); these results cannot validate it. In contrast, the closing-instruction cost interaction is supported under its own matched-state design.

Scope. Returned text and interface names do not measure internal reasoning quality. Expanded-inventory studies partly confound count, descriptions, and length; the matched study controls these quantities but does not identify a tool-specific mechanism apart from request position and format. The MBPP comparison uses only three assertions, and generated tasks cover four fixed families. Project-new tasks need not be absent from pretraining. Shared prefixes, recovery deviations, reused responses, and differing model configurations retain their original status in the appendices. Within-study corrections do not remove adaptivity across the research program.

8 CONCLUSION

Tool surfaces can redirect shared actions before feedback and interact with closing instructions to change continuation cost. Capability-matched repair and complete-execution tests constrain the direction and utility that can be inferred from exposure sensitivity. The revised account separates presentation effects, retained-state propagation, and execution utility. Public-state configuration reduces complete-task cost against full exposure, while static-core and direct-stop comparisons delimit its value. Future control rules require both state-specific predictions and direct evaluation of task utility.

AI USE STATEMENT

Generative AI tools, including OpenAI Codex, assisted with hypothesis and experiment design, literature search, writing, figures, and typesetting. The authors remain responsible for verifying all AI-assisted work and for the accuracy and integrity of the submitted content.

REPRODUCIBILITY STATEMENT

The main text defines the interventions and principal measurements; the appendices provide model and agent settings, task construction, complete results, statistical procedures, and study lineage. The

anonymous supplementary archive contains a core package for the earlier principal endpoint statistics, token accounting, and figures, plus added-study observations and code for the metadata-order and information-state tests. A study inventory distinguishes new collections from reused observations and development cohorts. These packages recompute statistics from reduced observations and saved predictions; they do not retrain classifiers or replay every raw model interaction and private test. Archived raw records and frozen protocols support internal traceability. The supplied package does not imply independent third-party reproduction.

REFERENCES

Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Wang. Agent S: An open agentic framework that uses computers like a human. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/394c7c30ea87b5c3521b4d9e9d419071-Abstract-Conference.html.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/abs/2108.07732.

Rahul Suresh Babu and Laxmipriya Ganesh Iyer. ToolChoiceConfusion: Causal Minimal Tool Filtering for Reliable LLM Agents, 2026. URL https://arxiv.org/abs/2606.06284v1.

Thierry Blankenstein, Jialin Yu, Zixuan Li, Vassilis Plachouras, Sunando Sengupta, Philip Torr, Yarin Gal, Alasdair Paren, and Adel Bibi. Biasbusters: Uncovering and mitigating tool selection bias in large language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 102460–102489, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a79875cc0d046ce7ce65f03f3affaa9e-Paper-Conference.pdf.

Shiyang Chen. Looking Is Not Picking: An Attention-Segment Account of Tool-Selection Failures in LLM Agents, 2026. URL https://arxiv.org/abs/2606.16364v2.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Kazem Faghih, Wenxiao Wang, Yize Cheng, Siddhant Bharti, Gaurang Sriramanan, Sriram Balasubramanian, Parsa Hosseini, and Soheil Feizi. Tool Preferences in Agentic LLMs are Unreliable. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20954–20969, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1060. URL https://aclanthology.org/2025.emnlp-main.1060/.

GLM-5 Team. GLM-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Zhicheng Guo, Sijie Cheng, Hao Wang, Shihao Liang, Yujia Qin, Peng Li, Zhiyuan Liu, Maosong Sun, and Yang Liu. StableToolBench: Towards stable large-scale benchmarking on tool learning of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11143–11156. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.664. URL https://aclanthology.org/2024.findings-acl.664/.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/edac78c3e300629acfe6cbe9ca88fb84-Abstract-Conference.html.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL https://arxiv.org/abs/2307.13702.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3102–3116. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.187. URL https://aclanthology.org/2023.emnlp-main.187/.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024a. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Weiting Liu, Jieyi Bi, Wanqi Zhou, Jianfeng Feng, Yining Ma, Ai Han, and Wenlian Lu. ToolAnchor: Anchoring Counterfactual Context to Boost Agentic Tool-use Capability, 2026. URL https://arxiv.org/abs/2607.14145v1.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e9df36b21ff4ee211a8b71ee8b7e9f57-Abstract-Conference.html.

Lennart Ljung. System identification. Technical Report LiTH-ISY-R-2809, Linkoping University, Division of Automatic Control, 2007. URL https://rt.isy.liu.se/research/reports/2007/2809.pdf.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.nips.cc/paper_files/paper/2024/hash/e4c61f578ff07830f5c37378dd3ecb0d-Abstract-Conference.html.

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.

Saanvi Paturi, Arsen Kenzhebayev, Arham Sethi, Vyas Raina, Ivaxi Sheth, and Vatsal Raina. When Tools Get in the Way: The Effect of Unnecessary Tool Availability on LLM Answering, September 2026. URL https://arxiv.org/abs/2609.14157v1.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/28e50ee5b72e90b50e7196fde8ea260e-Abstract-Conference.html.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language

models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ed3fea9033a80fea1376299fa7863f4a-Abstract.html.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering, 2024. URL https://arxiv.org/abs/2405.15793v3.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Z.ai. GLM-5.3 model card. Hugging Face, 2026. URL https://huggingface.co/zai-org/GLM-5.3. Accessed September 26, 2026.

Jiaru Zou, Ling Yang, Yunzhe Qi, Sirui Chen, Mengting Ai, Ke Shen, Jingrui He, and Mengdi Wang. AutoTool: Dynamic Tool Selection and Integration for Agentic Reasoning, 2025. URL https://arxiv.org/abs/2512.13278v2.

A Evidence map and study lineage

Tables 2 and 3 index the available evidence by scientific question and sampling unit. The accompanying anonymous evidence_inventory.csv gives 47 individual collection, archival, development, and reanalysis entries, their evidence status, manuscript locations, and source-file hashes. These entries are not 47 independent replications. We retain each study's original endpoint, correction, and interpretation rather than pooling heterogeneous effects or counting positive results as votes for the framework.

Independence and provenance. Recovery directories continue the same experiment and preserve earlier accepted observations; they are not extra batches. Repeated arms, common-prefix donors, and deterministic absorbing positions are not independent tasks. The two public-pass suffix studies reuse the same 188 states from the whole-task study, while generating new suffixes under their own frozen designs. A new readout of old responses creates a new analysis, not new trajectory data. Earlier archived reports retain their original verification level; this synthesis does not claim a fresh replay of every historical response. Technical development stops without complete endpoints do not count as null scientific results.

Scope of aggregation. The 34.94% result averages relative savings across twelve Flash task bundles. The 34.74% main result is the ratio of pooled complete-task costs for 192 Qwen tasks. The 67.30% and 28.62% figures instead concern suffix costs at 188 previously observed states, under different instruction comparisons. These estimates answer different questions and are not pooled. The same rule applies to immediate text information, proposed interfaces, unavailable-tool attempts, and final code-test outcomes: each retains its original denominator and does not replace another endpoint.

Table 2: Generation and propagation evidence, organized by endpoint. A task or bundle is the inference cluster; repeated responses and branches do not increase that count. Counts retain their study-specific meaning. Development cohorts and feasibility decisions appear in Appendix T.

Study family	Models	Evaluation unit and scale	Evidence and detailed location
Feedback-free text	Flash, Pro	Earlier studies: 8 tasks each; frozen decoder: 12 held-out tasks, 384 evaluation responses	Specified surface contrasts are distinguishable; the surface–tone contrast passes, but the full factor ordering does not. Appendices C, S.
Shared editing interfaces	Flash, Qwen, GLM	Same 24 tasks/model; 576 evaluation responses; GLM reuses materials	Qwen passes its original corrected test; GLM's interval spans both signs. Interface choice does not establish edit locality. Appendix L.
Metadata order	Qwen	128 tasks; 4 arms and 2 repetitions; 1,024 responses	Both original positive-direction tests fail; the observed first-repair order reverses. Appendix M.
Information state	Qwen	192 tasks; 4 arms; 768 complete episodes	Both frozen inspection and final-quality predictions fail. Appendix U.
Earlier history readouts	Flash, Pro	8-task legacy studies; separate stage-cue study: 36 tasks, 288 endpoints	Time-specific decoding and stage confounding limit temporal claims; the cue contrast is not detected. Appendices D, S.
Sequence prediction	Pro	Known 8-task sets; 1,024, 1,024, and 2,048-response studies; later 1,024 frozen test responses	Small exploratory gains do not yield consistent replication; all six later frozen kernel comparisons fail. Appendix D.
Pulse and withdrawal	Pro, Flash, Qwen	8-task sets; 592-response pulse; 1,184/model withdrawal	Localized Pro/Flash action effects are exploratory; Qwen does not reproduce that endpoint. Appendices D, F.
Demand timing	Pro, GLM	Pro: 1,616 responses; GLM: 8 bundles, 224 responses	Pro's exploratory timing interaction is positive; GLM has no removed-tool events in 64 branches. Designs differ. Appendices D, H.
Long window and state restoration	Qwen, Pro	8-task sets; 4,416 and 1,744 responses, respectively	Frozen extended-window and incremental history/workspace tests do not pass. Appendix D.

B Estimands and conditional implications

B.1 Cross-replicate mean separation

Fix a task bundle, deployment, and observation time. Define $\Delta_r = \phi(Y_r^a) - \phi(Y_r^B)$ for repetitions $r = 1, 2$, and a prespecified nonnegative diagonal weight matrix W. Under cross-repetition independence, equal target distributions, and finite second moments,

$$\mathbb{E}[\Delta_1^\top W \Delta_2] = \mathbb{E}[\Delta_1]^\top W \mathbb{E}[\Delta_2] = \|\mu^a - \mu^B\|_W^2. \tag{9}$$

The first equality follows from independence. Although the target is nonnegative, the finite-sample estimate need not be; clipping negative estimates changes its properties. Provider sampling independence is an assumption, not an externally verified property. A shared baseline and a shared pulse/sustained initial response induce dependence between contrasts. The analysis constructs whole contrasts within task clusters rather than treating correlated positions as independent tasks.

The older Qwen long-window estimator uses two branches per condition. For a block with squared distance d, it computes

$$\widehat{D} = \frac{1}{4}\sum_{i=1}^{2}\sum_{j=1}^{2} d(S_i, M_j) - \frac{1}{2}\{d(S_1, S_2) + d(M_1, M_2)\}. \tag{10}$$

Four normalized block estimates are averaged. This frozen estimator is retained rather than replaced with the newer sixteen-dimensional representation after observing outcomes.

Table 3: Additional propagation and execution evidence. New suffixes on old prefixes are distinguished from new tasks. Descriptive audits reuse the original responses and add no independent treatment observations.

Study family	Models	Evaluation unit and scale	Evidence and detailed location
Automatic B/P/E	Flash, Qwen	48 three-task bundles/model; 4,118 responses plus 298 absorbing positions	Eight frozen tests fail; later direct/text readouts are exploratory reuses. Appendices E, R.
History, demand, interval	Flash, Qwen	64 tasks/model from 4 families; 1,024 formal probes	All four corrected directional tests fail. Appendix Q.
Frontier and retention	Flash	8 small tasks; 12 three-ticket bundles; later 96 new two-ticket bundles	The three policy studies fail their respective joint criteria; the early 34.94% saving has a different estimand from the main result. Appendices J, K.
Complete-task state gate	Qwen	192 new tasks; 3 policies; 576 episodes	34.74% pooled whole-task saving and success noninferiority pass against full exposure; the joint comparison with static core fails. Appendix K.
Matched completion branches	Qwen	Same 188 public-pass prefixes; 376 then 752 entirely new suffixes	Conditional withdrawal savings and the closing-instruction cost interaction pass. Appendices K, O.
Text, action, cost reanalyses	All four	Existing responses only; text readout reuses 7,392 responses in 5 datasets	No general corrected text carryover; editing scope, fixed-charge cancellation, and stopping are separate descriptive audits. Appendices I, R, P, N.
Earlier controller development	Flash, Pro	Small pilots and 8–24-task studies; overlapping development data	Archived MPC, residual, re-anchoring, and guarded-contraction results do not establish general advantage over strong state policies. Appendix T.

B.2 Sufficient restoration equalizes future distributions

Suppose X_t contains every variable governing the future interaction kernel. If restoration gives two branches identical distributions of X_t, and both subsequently use the same surface policy and transition kernel, applying that kernel gives identical next-state distributions. Induction gives identical distributions of later observables. This conditional implication does not establish that a practical partial reset satisfies its premise. Hidden provider state, distinct control fields, or unstable sampling may violate the assumptions. The completed history/workspace intervention does not establish empirical reset equivalence or an internal neural-memory mechanism.

B.3 Propagation does not imply monotonicity

A file changed by an initial action may influence feedback much later. A new requirement may make an earlier behavioral difference observable only at a subsequent decision. Neither process requires the effect to decline every round. Additional contraction assumptions would be needed for an appropriate decay bound.

Input–state–output models provide a standard system-identification starting point (Ljung, 2007). For illustration, a linearized model with zero initial difference,

$$\delta x_{t+1} = A\delta x_t + B\delta u_t, \qquad \delta y_{t+1} = C\delta x_{t+1} + D\delta u_t, \tag{11}$$

gives immediate coefficient $CB + D$ and lag-k coefficient CA^kB for $k \geq 1$. These are algebraic consequences of the assumed model, not evidence that an agent satisfies it. Stability alone also does not guarantee monotonicity in every observed direction. The frozen predictive results in the paper directly test particular approximations instead of treating this derivation as empirical validation.

B.4 Conditional contraction and observable manifestation

For probability distributions ν, ν' use $\mathrm{TV}(\nu, \nu') = \sup_A |\nu(A) - \nu'(A)|$. Let ν_t^P, ν_t^B be pulse and baseline distributions of a sufficient complete state at a shared starting context c, and set

$\delta_t = \mathrm{TV}(\nu_t^P, \nu_t^B)$. Write $\tau_{t,c}^f(P, B)$ for their difference in expected readout $f(Y_t)$. After withdrawal, assume both regimes use the same possibly time-varying Markov kernel K_t on that state, and a common output kernel G_t. Let

$$\alpha_t = \sup_{x,x'} \mathrm{TV}(K_t(x,\cdot), K_t(x',\cdot)), \qquad \beta_t = \sup_{x,x'} \mathrm{TV}(G_t(x,\cdot), G_t(x',\cdot)). \tag{12}$$

For every measurable $0 \le f \le 1$,

$$|\tau_{t,c}^f(P, B)| \le \beta_t \delta_t \le \beta_t \delta_1 \prod_{j=1}^{t-1} \alpha_j. \tag{13}$$

Proof. The oscillation of $K_t g$ for $0 \le g \le 1$ is at most α_t. Integrating against the signed measure $\nu - \nu'$ gives $\mathrm{TV}(\nu K_t, \nu' K_t) \le \alpha_t \mathrm{TV}(\nu, \nu')$. Iterate and apply the same inequality to G_t, then use the bounded-readout characterization of total variation. This is a conditional mathematical result, not an empirical estimate of the coefficients.

Deterministically appending complete history may preserve disjoint histories and hence make the worst-case coefficient one. A smaller observable state requires its own sufficiency argument. Even a decreasing upper bound does not force the realized effect to decrease: changing task demands can change output sensitivity. Thus elapsed calls alone do not identify a decay rate. In addition, $D^E > D^P$ is a comparison of squared distances to baseline; it is distinct from testing direct sustained–pulse separation. The completed H4 results preserve the former target.

B.5 History information and the value of a retention rule

Suppose a policy can condition on public state z and history h. The class of all policies using (z, h) includes policies that ignore h, so its optimal expected utility is at least that of the smaller class using z alone, under a common utility and environment. This set-inclusion argument gives no such guarantee to a fixed heuristic that must retain recently used tools. The frontier-plus-lease comparisons therefore test incremental value directly, including success and token cost. The theory does not derive the two-response lease from a fitted half-life.

B.6 State-conditioned action distributions and execution value

Write the surface as $S_t = (\mathcal{U}_t, \Pi_t)$, separating executable capabilities $\mathcal{U}_t$ from their presentation Π_t, including descriptions and ordering. Capability matching does not make two presentations interchangeable. For two surfaces s, s' with $\mathcal{U}(s) = \mathcal{U}(s')$, a matched-context contrast is

$$\begin{aligned}\Delta^{\psi}(s, s'; h, c) = \mathbb{E}[\psi(Y_t) \mid do(S_t = s), H_t = h, C_t = c] \\ - \mathbb{E}[\psi(Y_t) \mid do(S_t = s'), H_t = h, C_t = c].\end{aligned} \tag{14}$$

Here ψ is an externally specified action readout. Execution readouts also require matched initial workspace and execution rules; for next-response readouts alone, Y_t can be replaced by (V_t, A_t). Fixed rules can distinguish inspection, actual modification, and declared finishing, retaining other or invalid outputs separately. Their probabilities describe recorded actions, not hidden cognitive modes. Declared finishing is not verified success. The metadata study (Appendix M) records the first proposal and its execution, not a latent transition between reasoning stages.

This distributional formulation alone does not predict the sign of an intervention. One restricted candidate uses the action probabilities $p_k(s, z, h)$ at public state z and history h:

$$\ell(s, z, h) = \log \frac{p_{\mathrm{inspect}}(s, z, h)}{p_{\mathrm{modify}}(s, z, h)} = b(z, h) + \gamma(s). \tag{15}$$

The log-odds expression applies when both probabilities are positive. Probabilities use the complete action denominator, retaining declared finishing, invalid, and other outputs; the analysis must not discard them to select an inspection/editing-only sample. The functional form of b and the presentation terms γ would be fixed using development data, then evaluated on independent tasks against a state-only predictor. The model requires $\ell(s, z, h) - \ell(s', z, h) = \gamma(s) - \gamma(s')$ to be constant across the specified states, a restriction that can fail. Allowing an unrestricted presentation-by-state term after evaluation would remove that restriction.

This probability ratio does not determine the full-denominator inspection rate when other actions also change. The additive candidate is proposed, not fitted or validated here, and identifies no neural decision threshold. An unconstrained score representation merely restates observed probabilities. The prespecified line-first repair prediction remains unsuccessful after introducing the broader notation.

Execution value is a separate target. Let Q^π indicate success on the complete task under surface policy π, and let Cost^π record its total execution cost. A state-conditioned evaluation may seek lower expected cost subject to a prespecified success constraint relative to a strong static comparator π_0:

$$\begin{aligned} \min_{\pi \in \mathcal{P}} \quad & \mathbb{E}[\mathrm{Cost}^\pi \mid z] \\ \text{subject to} \quad & \Pr(Q^\pi = 1 \mid z) \geq \Pr(Q^{\pi_0} = 1 \mid z) - \varepsilon. \end{aligned} \tag{16}$$

The public state z, candidate policies $\mathcal{P}$, comparator, cost definition, and noninferiority margin ε must be fixed before evaluation. This is a criterion for assessing a controller, not a proof that a tested heuristic attains it. An action-distribution difference cannot substitute for the success constraint or a cost comparison.

A further hypothesis concerns information sufficiency. Let $Q^{s;\rho}$ be complete-task success when only the first call uses presentation s and all later calls follow the same surface policy ρ. For independently specified inspection-favoring and modification-favoring presentations s_I, s_M with the same capabilities, define

$$\Delta Q(z) = \Pr(Q^{s_I;\rho} = 1 \mid z) - \Pr(Q^{s_M;\rho} = 1 \mid z) \tag{17}$$

under a common execution budget. One candidate prediction is $\Delta Q(z_{\mathrm{missing}}) > \Delta Q(z_{\mathrm{sufficient}})$: the success advantage of the inspection-favoring presentation is larger when relevant task information is missing. This does not require a sign crossing and does not follow from the log-odds equation. Costs must be reported separately. The metadata study alone does not test this complete-execution interaction; the subsequent frozen experiment in Appendix U does, and neither of its two positive-direction predictions is supported. Inspection may merely delay an otherwise identical repair; only complete execution can distinguish that possibility from a change in task success or cost. That follow-up freezes presentation labels, the information manipulation, direction, future policy ρ, and both primary contrasts before its collection. Its result constrains this candidate without retrospectively changing the earlier repair prediction. Maintaining different presentations on later calls would instead define a separate sustained intervention.

These extensions leave the historical estimands and conditional restoration/contraction results above unchanged. Neither the existence of a history pathway nor the ability of a broad state model to accommodate an observation establishes predictive validity.

C TOOL SURFACES, READOUTS, AND MODEL SETTINGS

Table 4: Surface definitions are study-specific. The same informal label does not imply identical tools across experiments.

Study	Surface contrast	Main measurement
Frozen opening	Minimal/Standard: 2/9 tools	Masked-text decoder information
Withdrawal/long window	M: read/edit/test; S adds glob/grep/write	Removed-tool attempts or 279-dimensional separation
MBPP opening	Common: read/edit/test/write; expanded adds glob/grep/todo	First proposed call gives qualifying implementation
Automatic B/P/E	Same 4/7-tool pair	Fixed response/action/public-state readout

Here edit, test, and todo abbreviate `str_replace_editor`, `run_tests`, and `todo_write`. The original fixed-count composition study exposes five callables in both conditions: a recorder and four proxies with different functions. Richness varies zero, two, five, or nine proxies plus a common recorder; intermediate inventories are nested prefixes in a fixed order. This jointly changes inventory size, length, and content. The saved-prediction fixed-count information score is 0.112581 nats; a

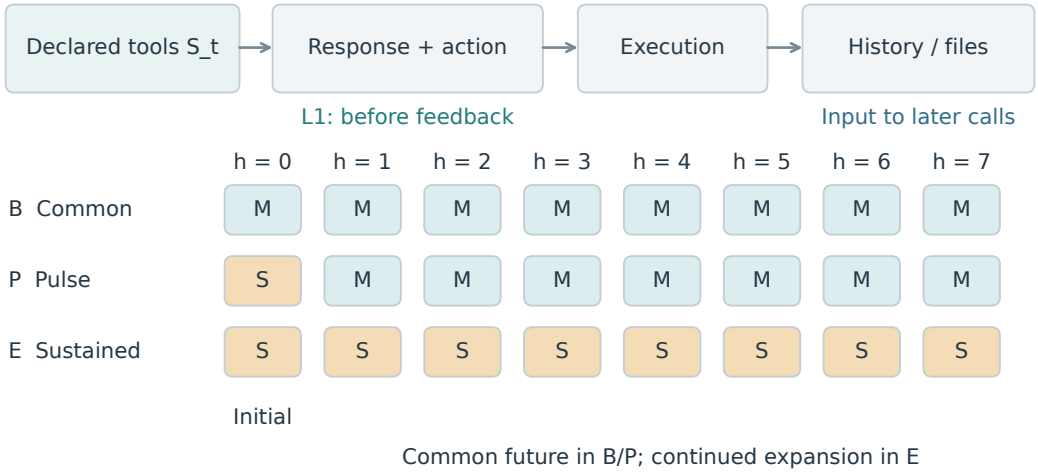

Figure 3: Three intervention targets. Baseline B, pulse P, and sustained exposure E separate current exposure, history after withdrawal, and continuing input. The schedules define comparisons; they do not assume persistence or decay.

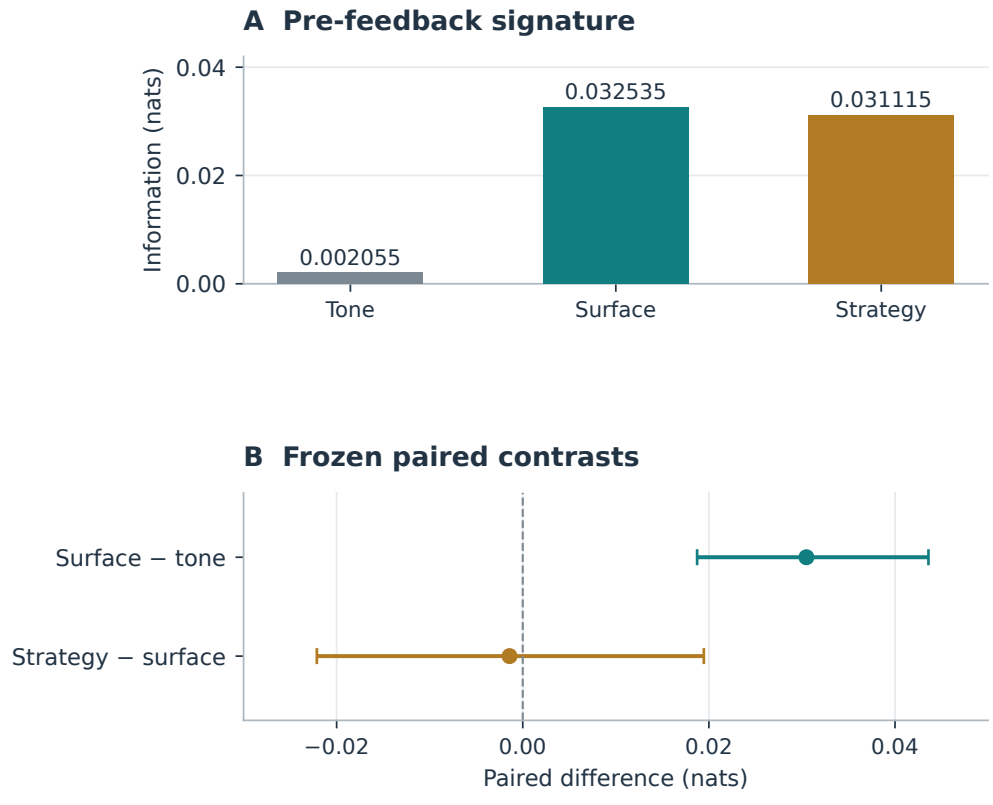

Figure 4: Pre-feedback text distinguishability. (A) Frozen decoder information on 384 evaluation responses. (B) Paired contrasts with individual approximate 97.5% task-bootstrap intervals. The surface contrast exceeds the selected tone contrast; decoder-training uncertainty is not included.

separate earlier refit obtained 0.112122 with the same balanced accuracy but different probabilities. The current manuscript uses the saved-score value and does not call the refit an exact reproduction.

The older temporal readout consists of 256 masked text-hash coordinates, nine response-structure coordinates, eight action/feedback coordinates, and six task-state coordinates. Block squared errors are normalized using the frozen scales, then averaged equally. Effect magnitudes from this representation are not directly comparable with the newer sixteen-dimensional readout.

Generation limits are study-specific. The common-prefix withdrawal studies use a maximum output of 4,096 tokens and a thinking budget of 1,024 for Pro, Flash, and Qwen; Pro and Flash additionally use maximum reasoning effort. The shared-interface study uses 4,096/1,024 for Flash and 8,192/1,024 for Qwen and GLM, where each pair gives maximum output/thinking budget. Qwen's long-window, completion-core, public-pass, and closing-factorial studies use 8,192/1,024. Qwen and GLM impose no additional output-effort constraint. Cross-model comparisons therefore need not match inference settings or sampling and do not isolate a model-family effect. Exact request identifiers and version-resolution records remain in the reproduction notes.

Model provenance. The DeepSeek-V4 report describes Pro and Flash (DeepSeek-AI, 2026); the Qwen release identifies the 35B-A3B version (Qwen Team, 2026). For GLM-5.3, we cite its exact model card and the GLM-5 family report recommended by that card (Z.ai, 2026; GLM-5 Team, 2026). Pro and Flash denote the recorded API identifiers `deepseek-v4-pro` and `deepseek-v4-flash`. Reported model names and study-specific request settings define the experimental configurations.

Original opening diagnostics. Saved-prediction recomputation gives surface information 0.143969 nats in the original surface-by-prompt study, versus 0.009384 for its prompt contrast. The fixed-count score is 0.112581 nats with balanced accuracy 0.9000; the richness study gives mean task-level Spearman correlation 0.842459. Richness jointly changes count, schema length, and functional content. These are saved-score checks, not independent reproductions of all training and uncertainty procedures.

D Additional completed evidence

D.1 Temporal prediction and a distance-based pulse test

Two earlier temporal batches each contain 1,024 responses on eight known tasks. An FIR history model gives relative prediction gains of $+0.6597\%$ against K0 and $+0.2936\%$ against K0_time in the first batch, with 6/8 positive tasks in each comparison. The prospective repetition yields $+0.1092\%$ and $+0.0327\%$, each with 4/8 positive tasks, failing its descriptive consistency rule. A 2,048-response staged study yields -0.0090% mean gain against K0_cell. These are new-sequence tests on known tasks, not task-population generalization.

The 592-response common-prefix pulse test gives delayed four-block separation $+0.00097513$, nominal one-sided $p = 0.4287$. A post hoc discovery of more later removed-tool attempts motivated the targeted withdrawal endpoint, evaluated on separate responses without pooling the discovery batch. Positive targeted behavior does not replace the inconclusive distance result.

An older 80-trajectory decoder curve begins at 0.086435 nats, falls to 0.055965, then rebounds to 0.084015. Its first-lag ratio 0.647477 fails the original below-one-half rapid-decay threshold. Different decoders at different times prevent treating this curve as a fixed-coordinate response kernel. Historical sustained-exposure experiments also co-vary some stage prompts with surface switches; they motivate distinguishing ongoing exposure from withdrawal rather than proving a tools-only washout mechanism.

D.2 Frozen kernels and explicit pathway contrasts

Figure 5 visualizes the frozen comparisons. The later 2,384-response prediction study uses 80 prefix, 1,280 identification, and 1,024 held-out test responses. Single-pulse and baseline sequences identify five fixed models. All predictions for held-out SSMM, SMSM, SSSM, and SSSS sequences are saved before test authorization. The support rule requires at least 1% relative gain and a positive corrected

task-cluster lower bound. Six comparisons use Bonferroni correction; no fitting or selection uses test responses. All 105 truncated test responses remain included.

Table 5: All frozen-kernel comparisons. Positive gain favors the candidate; no comparison meets its joint support rule.

Candidate	Baseline	Relative gain	Corrected loss-difference lower
Strict LTI	Current-time	-12.5259%	-0.01174926
Strict LTI	Exposure-count	+2.7977%	-0.00154095
Current-time FIR	Current-time	-22.6465%	-0.01760424
Current-time FIR	Exposure-count	-5.9447%	-0.00572348
LTV	Current-time	-61.5040%	-0.04663467
LTV	Exposure-count	-39.5106%	-0.03481954

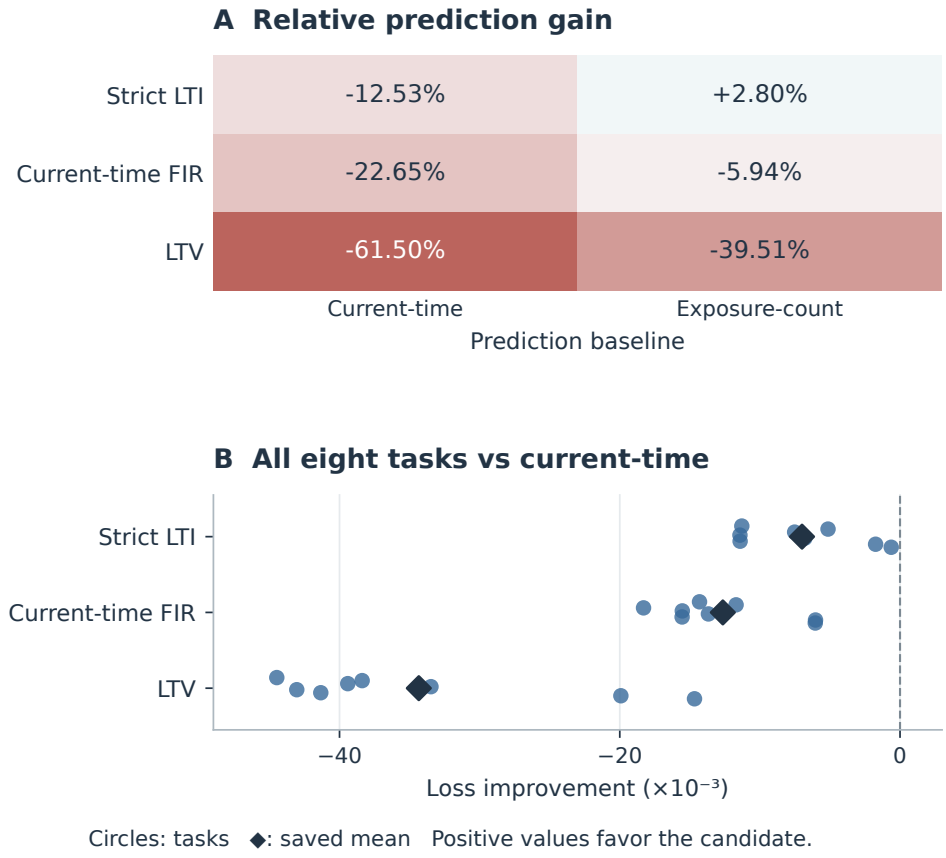

Figure 5: Frozen prediction tests on 1,024 held-out responses. (A) All six relative-gain comparisons; positive values favor the candidate and negative values favor its baseline. (B) Loss improvements against the current-time baseline for all eight tasks (circles), with task means (diamonds). Every task favors the current-time comparator for these three candidates. The small positive aggregate LTI gain against exposure count does not meet the joint support rule.

The 1,744-response pathway experiment independently retains or resets history and workspace after an initial donor interaction. Relative to resetting both, history-only, workspace-only, and both-retained increments are -0.00421799, -0.01093463, and -0.00884689. Corrected lower bounds are -0.02821085, -0.04003185, and -0.03689926; adjusted $p = 1.000$ in all three. The batch has no protocol deviations, but still uses eight reused tasks and a fixed representation. These explicit restoration contrasts are not natural mediation percentages.

D.3 Demand timing and the Qwen observation curve

The demand-timing study contains 32 common starting points and 384 branches. Its interaction averages two matched differences: the S-minus-M effect at $h = 1$ under early versus late demands, and the S-minus-M effect at $h = 3$ under late versus early demands. The conservative Neyman standard error is 0.03125; the approximate normal interval is exploratory because events are sparse and some variances can be zero. Requirements, cumulative tests, private checks, and verifier state change together at a stage transition.

Table 6: Descriptive Qwen long-window means in the original frozen representation. Negative finite-sample estimates are retained. Primary tests are those in Table 11, not a selected part of this curve.

Lag	0	1	2	3	4	5	6	7
Mean $\times 10^3$	0.747	5.451	-0.598	7.293	-1.237	-2.608	-3.339	1.397

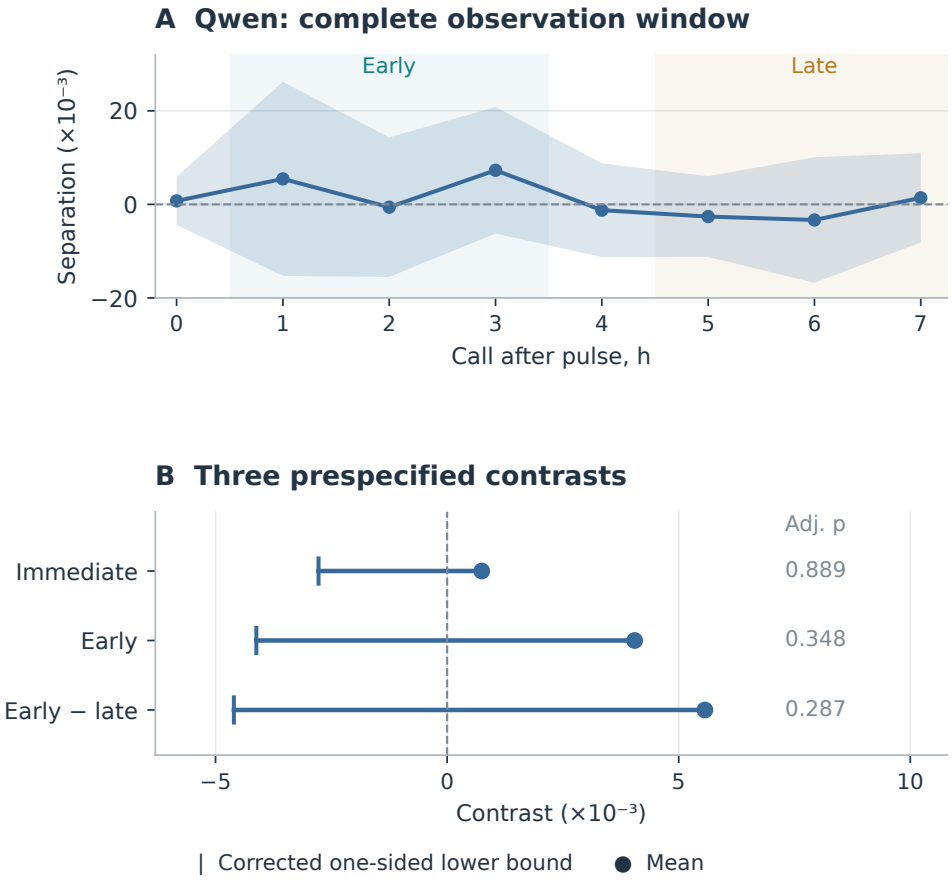

Figure 6: Qwen long-window evidence from 4,416 responses on eight tasks. (A) Means with archived approximate 95% simultaneous Bonferroni t bands across eight calls; connecting lines do not fit a decay model. Shading marks the early and late windows. (B) Prespecified mean contrasts (dots), corrected one-sided lower bounds (ticks), and adjusted p values. Segments run from lower bound to mean, not a two-sided interval. All three lower bounds lie below zero; negative finite-sample estimates remain visible.

D.4 Execution counts and all joint comparisons

The first policy study separates four development tasks from eight synthetic evaluation tasks, with three repetitions per evaluation task and policy. The state comparator is an explicit heuristic. Development uses audited completeness and policy disagreement, not efficacy. Development and evaluation total 112 episodes and 461 responses; the evaluation has 96 episodes.

Table 7: Initial four-policy evaluation. Success counts are descriptive, and minimal's 24/24 is not proof of population superiority.

Policy	Successes / 24	Tokens	Responses	Tool errors
Full	23	218,291	93	0
Minimal	24	243,884	103	1
State	21	187,229	89	0
History	23	282,329	108	1

State versus full has success difference -8.33 percentage points and mean task-relative saving $+10.34\%$. History versus full, state, and minimal has success differences 0, $+8.33$, and -4.17 points and savings -29.44%, -45.96%, and -31.13%. All four joint criteria fail. A zero-variance success contrast is inconclusive rather than automatically noninferior.

In the MBPP study, history versus frontier, count control, and full has a $+4.17$-point success estimate in all three comparisons. The respective corrected success lower bounds are -7.58, -7.58, and -16.79 points. Mean token savings are -27.14%, -3.39%, and $+34.94\%$, with corrected lower bounds -61.46%, -37.26%, and $+16.65\%$. Its twelve convenience task bundles are small for noninferiority inference. The provisional staged design preceded inspection of the earlier small-task final effects, but its finalization and collection followed that inspection. This timing is disclosed as part of the exploratory design history.

The complete MBPP campaign includes 24 development episodes, 144 evaluation episodes, and the separate 144 first-response episodes: 312 episodes and 1,819 accepted/authorized responses, reporting 4,107,817 logical tokens. Each phase completes its own frozen audit and analysis. Schema/history byte lengths are proxies, not identified per-segment billing or measures of reasoning quality.

E Completed automatic trajectory validation

E.1 Allocation and interventions

A separate study uses 168 public tasks not overlapping the earlier MBPP selection by task ID, grouped into eight development and forty-eight evaluation bundles of three tasks. Semantic deduplication and pretraining absence are not claimed. Two deployments have one development repetition and two evaluation repetitions, yielding sixteen development and 192 evaluation units. All three requirements, starter files, and public checks are provided at $h = 0$; no new requirements arrive later.

Baseline B uses M throughout, pulse P uses S only at $h = 0$, and sustained E uses S throughout. The common inventory contains read/edit/test/write, with glob/grep/todo added by S. P and E share one real initial response and execution, then clone native history and workspace bytes without replaying the initial tool. Every response executes at most one legal call. Invalid or unexposed calls and calls exceeding the execution limit receive recorded rejection feedback.

Responses without tool calls terminate only when the stop reason is end_turn or stop_sequence. A max_tokens response without calls is not normal completion. After termination, no more model calls occur; predetermined absorbing observations fill later positions. A unit requires at most 23 actual responses, giving maxima of 368 development and 4,416 evaluation responses. This campaign uses neither human annotation nor model-based semantic judges.

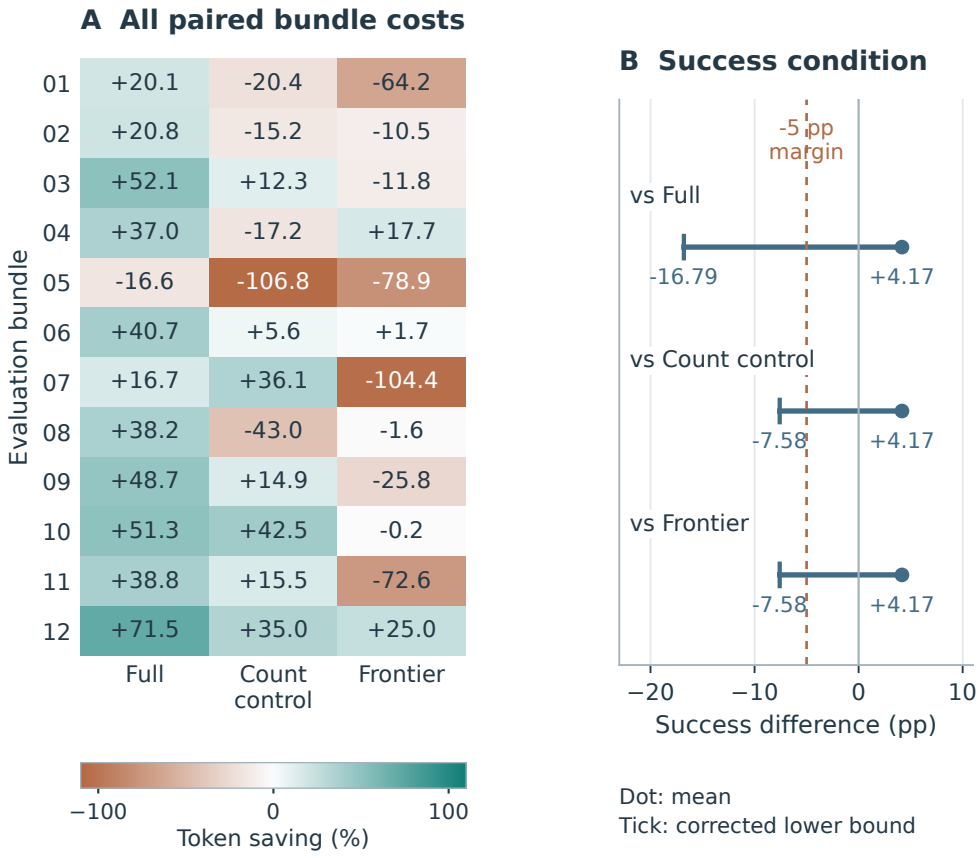

Figure 7: Bundle-level execution results and the success condition. (A) All 12 paired bundle token savings for frontier plus lease, in fixed evaluation order; cells report percentages to one decimal, with a symmetric color scale. Negative values indicate increased cost. (B) Success-difference means (dots) and corrected one-sided lower bounds (ticks), with the -5 pp noninferiority margin. All lower bounds fall below the margin. Success and cost use the original six-comparison Bonferroni correction and 12 bundle clusters.

Table 8: Sixteen-dimensional automatic readout. Each block receives one third of total weight; coordinates within a block are equally weighted.

Block	Coordinates	Weight / coordinate
Response (4)	Visible characters / 20,000 capped at 1; proposed calls / 4 capped at 1; nonempty returned thinking; truncation	$1/12$
Action (7)	read, search, edit, test, plan; no proposed call; any tool error	$1/21$
Public state (5)	Source changed; public test executed; fraction of current files passing public checks; repeated unchanged read; termination	$1/15$

E.2 READOUT AND DEVELOPMENT GATE

Visible text excludes signatures, schema text, tool JSON, and tool results. Private acceptance never enters the readout. Absorbing observations carry the public-pass fraction and set termination to one, with all other coordinates zero; they are not fabricated model responses. Length, action, and public-state differences cannot be equated with validated semantic reasoning quality.

The frozen development gate required each deployment to have at least two features with pooled nonzero rate in $[0.10, 0.90]$, including an action/state feature. Shared $h = 0$ is counted once. Actual responses must fill at least 20% of positions in both $h = 1$–2 and $h = 6$–7 windows. No positive treatment effect is required. Both deployments passed after sixteen units and 354 responses. The subsequent evaluation completed all 192 units with 4,118 actual responses and 298 predetermined absorbing positions. Full replay audit passed, verifying 384 clones; evaluation reported 12,195,826 logical tokens. Absorbing positions are bookkeeping after termination, not additional API responses.

E.3 FROZEN COMPARISONS

Each evaluation bundle uses two repetitions for the cross-replicate estimator in Appendix B. Four comparisons per deployment test immediate separation, mean early pulse separation ($h = 1$–2), early minus late pulse separation ($h = 6$–7), and late sustained-minus-pulse separation. Eight one-sided tests use $\alpha = 0.05/8$, forty-eight bundle clusters, and df=47 t bounds. Zero variance is inconclusive. Persistence and attenuation must both pass before an observed residual can be described as weaker in the late window. Block-specific curves and termination rates remain descriptive; a favorable block cannot replace the primary composite after inspection.

E.4 COMPLETE RESULTS

Table 9 reports all eight frozen comparisons. None meets its support criterion. H4 compares $D^E - D^P$, each relative to baseline, not $\|\mu^E - \mu^P\|^2$. Negative estimates remain included, and the primary representation and weights are unchanged. These results do not establish equivalence; they also do not support the proposed immediate–persistence–attenuation chain in this representation. Figure 8 visualizes the complete contrasts.

Table 9: Automatic B/P/E validation: 48 bundle clusters per deployment. H1: initial pulse separation; H2: early pulse separation; H3: early minus late pulse separation; H4: late sustained minus pulse distance from baseline. Lower bounds use $\alpha = 0.05/8$ and df=47.

Deployment	Test	Mean	Corrected lower bound	Supported
Flash	H1	-0.004376	-0.023472	No
Flash	H2	+0.000613	-0.024654	No
Flash	H3	+0.019421	-0.015655	No
Flash	H4	+0.003632	-0.032546	No
Qwen	H1	+0.011113	-0.014466	No
Qwen	H2	+0.003162	-0.017234	No
Qwen	H3	-0.003900	-0.034262	No
Qwen	H4	+0.002445	-0.036230	No

F SAMPLING DEVIATIONS AND AUDIT SCOPE

Completed-study audits reconcile saved requests and responses with native histories, cloned workspaces, tool execution, and reported outcomes. Interrupted runs resume from saved state without replacing accepted responses. Failed requests with unknown usage remain separate from observed token totals. For example, the Qwen long-window study retains all 4,416 accepted responses and records five additional attempts with unknown usage; its reported total is 13,980,977 logical tokens.

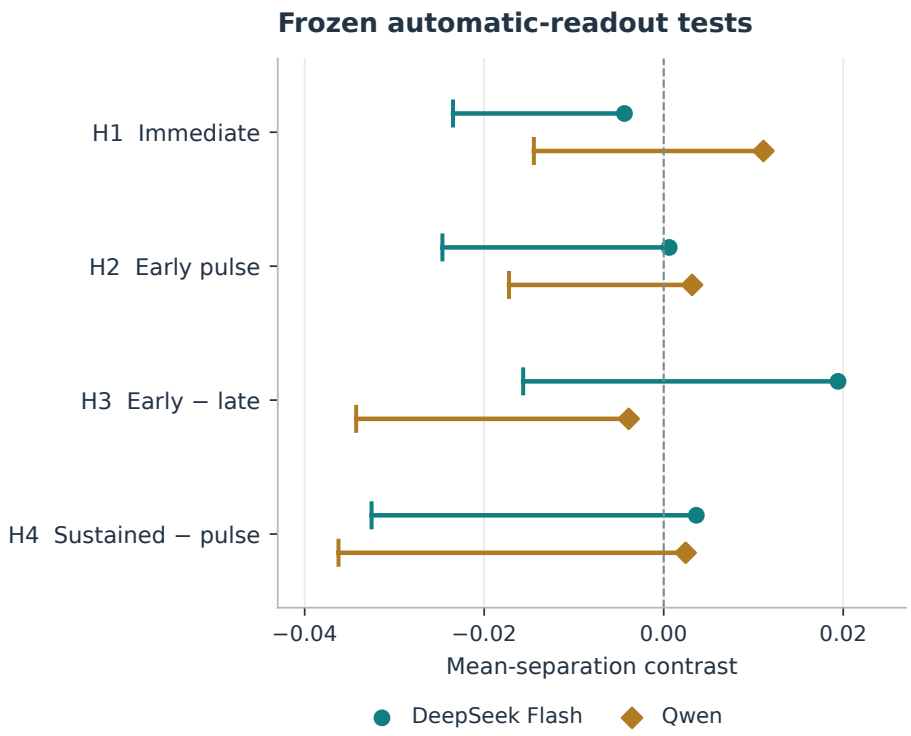

Figure 8: All eight automatic validation contrasts. Circles denote Flash estimates and diamonds Qwen estimates; ticks denote corrected one-sided lower bounds, and the vertical line marks zero. Segments are lower-bound-to-mean intervals, not two-sided confidence intervals. The measurement is fixed response/action/public-state mean separation.

Audit repairs changed technical file matching, not the scientific definitions or sampled outcomes. The separate reproduction notes retain the operational record. Record consistency does not verify model weights or eliminate the stated inferential assumptions.

The earlier withdrawal studies had collection interruptions, and the Flash study also had substantial truncation. These deviations, the eight reused tasks, and the exploratory analysis history limit the evidence to the stated settings. We retain all truncated responses and preserve the original predictions and analyses. The frozen-kernel study likewise retains its 105 truncated test responses. These studies are not upgraded to confirmatory evidence by passing a consistency audit.

G Numerical tables accompanying the main figures

The following tables retain the exact counts and primary contrasts visualized in the main text.

H Demand timing under a common continuation surface

The GLM demand-timing study uses eight constructed programming bundles with two independent repetitions, forming 16 units. Each unit generates M/S initial responses and early/late continuation branches under the common surface, for 224 accepted responses in total. The removed tools are `write`, `glob`, and `grep`. Each condition has 16 branches; none attempts any removed tool at $h = 1, 2, 3$. The initial-surface by demand-timing interaction is zero with zero estimated variance, so a positive interaction is not supported and an informative interval cannot be computed. The study is

Table 10: Post-withdrawal attempts on eight reused tasks. Entries count branches with at least one attempt at $h = 1$–3. All attempts to removed tools are rejected. Risk differences average common prefixes; p values are nominal directional randomization results.

Deployment	M pulse	S pulse	Difference (pp)	p
Pro	7/128	29/128	+17.1875	0.0001
Flash	10/128	38/128	+21.8750	0.0001
Qwen	3/128	1/128	-1.5625	0.9385

Table 11: Qwen long-window primary contrasts. All 4,416 responses pass the independent completion audit. None of the three corrected lower bounds is positive.

Contrast	Mean	Corrected lower bound	Adjusted p
Immediate ($h = 0$)	+0.00074670	-0.00277607	0.889430
Early ($h = 1$–3)	+0.00404842	-0.00411912	0.347561
Early minus late ($h = 5$–7)	+0.00556532	-0.00460372	0.287197

Table 12: MBPP interaction evaluation: 12 bundles, two repetitions, six policies. Tokens are reported logical input and output usage. The method's history label denotes frontier plus the two-response lease.

Policy	Successes / 24	Tokens	Responses	Tool errors
Full	19	771,523	290	12
Minimal	21	649,627	299	6
Frontier	19	380,407	179	1
Frontier + lease	20	460,238	188	0
Count control	19	503,016	203	3
Fixed schedule	21	720,641	283	14

exploratory; its task and timing design differs from the Pro timing study and does not support a direct model comparison.

Table 13: Demand-timing branch outcomes for GLM. Success is descriptive and does not replace the frozen removed-tool endpoint.

Condition	Branches	$h = 1$ events	$h = 2$ events	$h = 3$ events	Successes
M / early	16	0	0	0	11
M / late	16	0	0	0	7
S / early	16	0	0	0	12
S / late	16	0	0	0	3

All 224 responses pass the final replay audit. Two interrupted requests have unknown usage; continuation from saved history and workspace preserves the original tasks, endpoint, and analysis. Zero observed attempts neither establishes equivalence nor implies absent internal computation.

I Exploratory classification of post-withdrawal text

I.1 Data and leakage controls

We reuse 7,392 responses: 1,024 each from Pro, Flash, and Qwen withdrawal; 4,096 from the Qwen long window; and 224 from the GLM demand-timing study. These are existing responses, not a new collection. Features use only the current assistant's returned thinking text. Unicode normalization, case folding, removal of ten exact tool names, and whitespace normalization reduce literal name

echoes. Neither prompts, histories, tool schemas, tool-call JSON, feedback, task identifiers, nor labels enter the feature vector. Empty thinking and truncated responses remain included. The main classifier uses character TF-IDF 3–5 grams (minimum document frequency 2; at most 8,000 features; sublinear term frequency) and logistic regression ($C = 1$, liblinear; 2,000-iteration limit).

One shared post-withdrawal classifier is trained per fold and evaluated at every held-out lag. The first scope withholds one of four task-template families in the older datasets and one of eight task bundles for GLM. The second scope withholds independent prefixes within each task (four folds in the older datasets; two repetition folds for GLM). All descendants and lags of a shared prefix remain together. The within-task scope was added after the first four initial results were known; it is explicitly post hoc. Hyperparameters were not tuned on held-out effects.

Information is $\log 2 + \log p(y)$ averaged equally over tasks; balanced accuracy is descriptive. Each of ten tests uses 199 allocation-preserving whole-trajectory label permutations with classifier refitting. Older datasets preserve two S labels among four branches per prefix; GLM swaps M/S within each unit, jointly across early and late calendars. TF-IDF is fit only on training-fold text and cached across label permutations because it is label-independent. The support gate requires positive information and Holm-adjusted $p \leq 0.05$ across all ten tests. The minimum permutation p resolution is 0.005. Correction does not erase the adaptive choice of the second scope.

I.2 All results and interpretation

Table 14 contains every main test; none passes. Flash's within-task result has the strongest nominal evidence, but its adjusted $p = 0.150$. Its shared-readout scores at $h = 1, 2, 3$ are $+0.012723$, $+0.004717$, and -0.003958 nats. This descriptive pattern motivates an independently testable hypothesis; no per-lag or attenuation test establishes decay here. Pro and Qwen short-window late positive scores vary with the generalization scope. Qwen long-window scores are negative at every lag in both scopes. Figures 9 and 10 show both scopes and all lags, without selected-time significance claims.

Table 14: All ten exploratory text-readout tests. Across: held-out task families/bundles. Within: same tasks, independent held-out prefixes. Scores are post-withdrawal task averages; adjusted p values use one Holm family of ten.

Dataset	Scope	Information (nats)	BA (%)	Nominal p	Holm p
Pro	Across	-0.000141	54.04	0.045	0.405
Pro	Within	-0.008503	51.82	0.620	1.000
Flash	Across	-0.008195	52.73	0.565	1.000
Flash	Within	+0.004494	55.21	0.015	0.150
Qwen withdrawal	Across	+0.001773	53.39	0.050	0.405
Qwen withdrawal	Within	-0.005393	51.43	0.495	1.000
Qwen long	Across	-0.020498	49.05	0.965	1.000
Qwen long	Within	-0.019303	48.10	0.935	1.000
GLM	Across	-0.009788	46.88	0.690	1.000
GLM	Within	-0.007996	46.88	0.600	1.000

For GLM, nonempty thinking occurs in 31/32 initial responses and 54/64, 17/64, and 19/64 subsequent responses. A first-scope auxiliary readout including returned prose also has negative overall information (-0.0080 nats); missing thinking alone is not an established explanation for the result. Raw-text, length-only, combined-text, initial-readout-transfer, and per-lag classifiers remain auxiliary analyses. Text decodability does not measure semantic reasoning quality or internal memory. This analysis provides no additional corrected support for textual carryover, attenuation, or the full TSCTD mechanism. The reanalysis preserves the input records and uses no new model responses or tool executions.

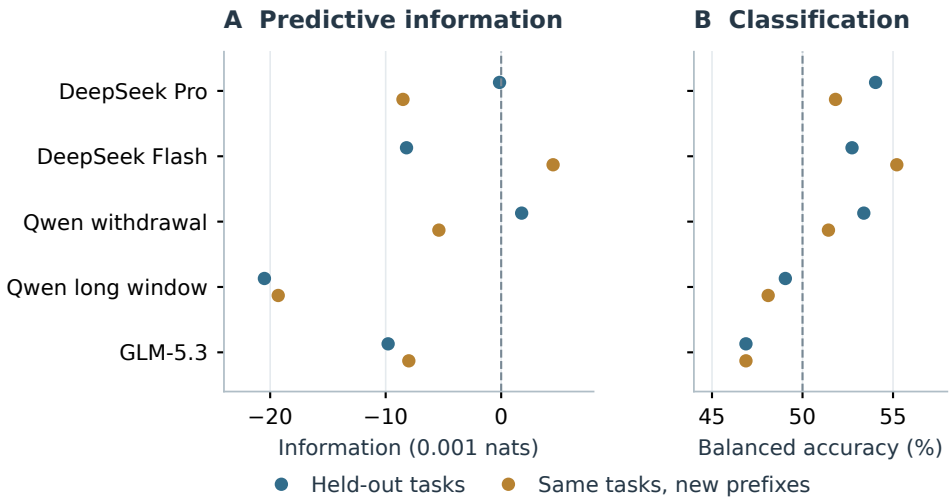

Figure 9: Exploratory classification under two generalization scopes. All ten tests use the same joint Holm correction. Information is decoder log-loss improvement; accuracy does not independently establish a positive information effect. No result passes the composite support gate.

J Earlier frontier and short-retention policy studies

J.1 A frontier-plus-lease policy

Let z_t include current source versions and public verification state. A task-state frontier $\mathcal{F}(z_t)$ exposes `write` when a pending candidate is a stub or has failed verification, and `run_tests` when a modified candidate awaits checking; an all-passing state has an empty frontier. This task-specific rule adapts precondition–effect filtering. To retain short-range action continuity (Figure 11), inspect the previous two responses in reverse recency order, collect distinct successfully executed common tools outside the frontier, and retain at most two:

$$S_t^{\text{lease}} = \mathcal{F}(z_t) \cup \mathcal{L}_t, \qquad |\mathcal{L}_t| \le 2. \tag{18}$$

The common inventory is `read`, `str_replace_editor`, `run_tests`, and `write`. The full inventory adds `glob`, `grep`, and `todo_write`. Final ordering follows a fixed canonical list. The selector requires no additional model calls or training. It is an explicit candidate policy, not a learned decay law. Both frontier and lease agents retain native conversation history; the comparator is not a memory-free agent.

J.2 Execution comparisons

We adapt Mostly Basic Python Problems (MBPP) into interactive code-editing and test-running tasks. Twelve development and 36 evaluation problems form bundles of three sequential tickets (four development and twelve evaluation bundles). Each bundle is one complete agent task in that earlier study; its token cost includes all calls across its three tickets. After an audited development gate based on policy disagreement rather than efficacy, two repetitions of six policies yield 144 evaluation episodes. Success requires all nine original assertions. Each ticket allows at most six responses and one executed tool per response. Flash uses max_tokens=4096, thinking budget 1024, effort=max, and no automatic retries.

The policies are full exposure, a fixed three-tool minimal core, the state frontier, frontier plus lease (`history`), an extra-tool-count control, and a fixed schedule using full exposure on the first call of each ticket and common tools thereafter. The count control uses its own recent history to determine extra count but rotates identities; it is neither wholly history-free nor exactly schema-token matched.

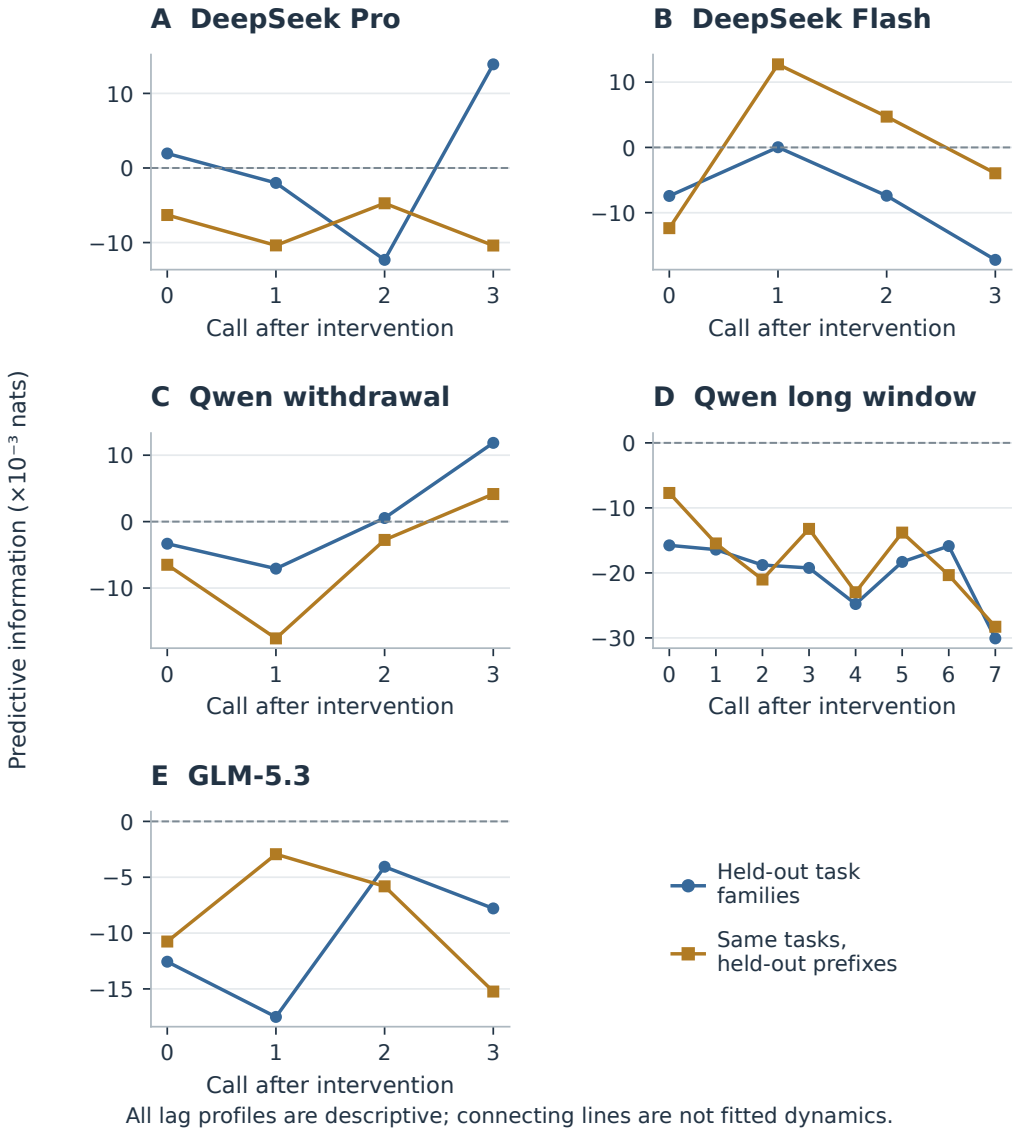

Figure 10: Complete descriptive lag profiles using each fold's shared post-withdrawal readout, with held-out predictions aggregated by task. Connecting lines do not fit a dynamical law. The classifiers differ across datasets and scopes; their scores are not coordinates of one universal response kernel.

Cost is the provider-reported logical input plus output tokens, including history and tool declarations as counted by the provider, rather than wall-clock time or billed currency. For bundle i, relative saving is $s_i = 1 - C_i^{\text{candidate}}/C_i^{\text{baseline}}$, averaged equally across bundles. This differs from the ratio of pooled totals in Table 12. Joint benefit requires a corrected success-difference lower bound above -5 percentage points, a positive saving lower bound, and at least 10% point saving. Six one-sided Bonferroni bounds cover success and cost for the three prespecified comparisons, using twelve bundle clusters and df=11.

J.3 Cost savings and the role of the comparator

Figure 12 summarizes the execution tradeoff. Against full exposure, the lease policy achieves **34.94% mean bundle-relative token saving**, with corrected lower bound $+16.65\%$. Its observed success count is 20/24 versus 19/24, but the corrected success-difference lower bound is -16.79 points:

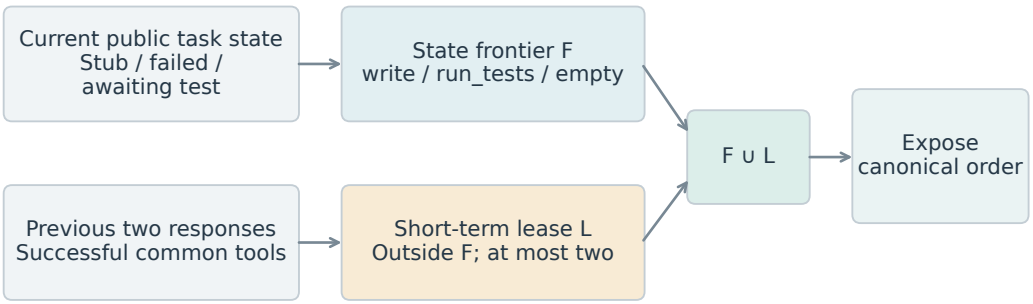

Figure 11: Frontier-plus-lease configuration. Public verification state selects the task frontier. Successfully executed common tools from the previous two responses supply at most two additional tools, in reverse recency order. The union is exposed in canonical order and recomputed at every response. This is a policy schematic, not evidence of a decay law.

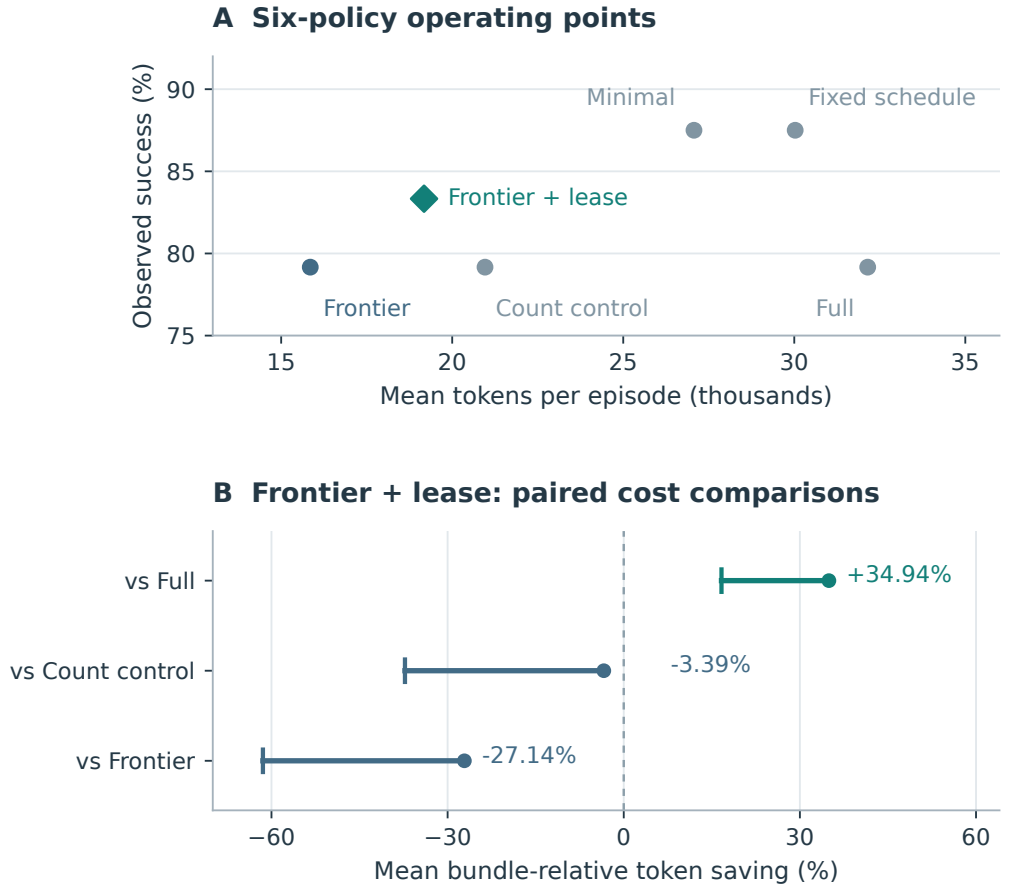

Figure 12: Execution value on the interactive MBPP adaptation. (A) Descriptive operating points for all six policies: 24 episodes per policy, clustered in 12 bundles. Costs are pooled tokens divided by 24; the success axis is expanded for readability. (B) Frontier-plus-lease savings relative to three baselines, averaged over paired bundle ratios. Dots show means and ticks show corrected one-sided lower bounds; segments are not two-sided intervals. The 34.94% saving is relative to full exposure. Success bounds and all bundle savings appear in Figure 7; joint success–cost support is not established.

success noninferiority is not established. Against frontier and count control, mean savings are -27.14% and -3.39%, with success lower bounds -7.58 points in both comparisons. All three joint criteria therefore fail. The supported cost component against full exposure is distinct from an incremental advantage of retaining history over an already selective frontier.

An earlier, separate four-policy experiment on eight synthetic evaluation tasks also fails all four joint criteria. Full, minimal, state, and history succeed on 23, 24, 21, and 23 of 24 episodes, respectively. History costs 29.44% more than full by the task-average ratio. Appendix D retains these results. Thus exposure selection is a concrete efficiency intervention worth testing, but these two studies do not establish the lease rule as a generally better controller. All 23 MBPP evaluation truncations remain included; the public-task adaptation and design history warrant exploratory interpretation.

K Completed whole-task and public-pass branch evaluations

K.1 Whole-task materials, policy, and allocation

The completion-core study uses a local archive of raw MBPP. It excludes the union of 442 IDs from five earlier material sets and exact duplicate old task prompts. Reference implementations must pass all three original assertions; starter and constant-None controls must fail the required checks. This compatibility selection precedes all new model responses. Four development tasks are selected from unused IDs 511–600 and 192 evaluation tasks from unused IDs in 11–510 and 601–974, using a frozen random seed. Because the evaluation pool includes the benchmark's original training range, the results describe a project-new interactive adaptation, not the standard held-out MBPP score. Each task is an independent single-ticket agent episode, unlike the older three-ticket bundles.

The three policies use one repetition per task. Task order and policy order within task are randomized with separate frozen random seeds; policy episodes for a task are adjacent. Full uses read/edit/test/glob/grep/write/todo, static core uses read/edit/test/write, and completion core presents the latter until the current source version passes every public check, then an empty list. Tool order follows the same canonical list. The controller has no model calls of its own. The empty list does not skip the model's closing response. The implementation supports restricted Python operations and a bounded evaluator; it is not a general repository environment.

Each task exposes the first original assertion; the other two are reserved. All three determine final success; the two private assertions never influence surface selection. All policies use Qwen3.6-35B-A3B with the inference settings in Appendix C. Each response executes at most the first tool proposal, and only if it is valid and exposed. Extra, malformed, or unexposed proposals receive recorded rejection feedback. A no-tool end_turn or stop_sequence ends the episode; a no-tool max_tokens response receives the same continuation message in every arm. All responses count toward the eight-response cap, including summaries and truncations.

The development gate requires complete replay integrity and at least two tasks on which completion core and static core actually expose different surfaces; it does not select on efficacy. All 12 development and 576 evaluation episodes complete, with 2,747 evaluation responses and 2,806 responses overall. One additional failed request has unknown usage. Collection resumes from saved state without repeating accepted outcomes, and final replay and accounting audits pass.

K.2 Frozen endpoints and complete comparisons

For task i, let C_i^b and C_i^c be baseline and candidate logical tokens, and let $Q_i^b, Q_i^c \in \{0, 1\}$ denote success. The primary cost estimate is $\bar{d} = n^{-1} \sum_i (C_i^b - C_i^c)$ with one-sided Student-t lower bound using $n - 1$ degrees of freedom. All tasks, including failures, remain in the denominator. Zero sample variance is marked inconclusive. Define $L = \sum_i \mathbb{1}\{Q_i^b = 1, Q_i^c = 0\}$. A conservative success-difference lower bound is $-U$, where

$$U = \mathrm{Beta}^{-1}(1 - \alpha; L + 1, n - L), \qquad U = 1 \text{ if } L = n. \tag{19}$$

Since the success difference is the gain probability minus the loss probability, bounding loss alone gives a conservative lower bound. It does not credit candidate-only successes to offset losses. This replaces neither the older experiments' frozen method nor their conclusions. For the whole-task study,

$n = 192$, $\alpha = 0.0125$ for each of four endpoints, and noninferiority requires $-U > -0.05$. At this size the criterion permits at most three baseline-only successes; the static-core comparison has four.

Table 15: Whole-task Qwen evaluation. All 192 tasks per policy are included. Pooled saving and mean per-task relative saving have different definitions.

Policy	Success / 192	Total tokens	Mean tokens	Responses
Full	162	1,705,473	8,882.7	949
Static core	164	1,446,324	7,532.9	977
Completion core	167	1,112,946	5,796.6	821

Table 16: Completion core versus each baseline. Cost and success bounds are one-sided 98.75%; pp means percentage points. Joint support also requires at least 10% pooled saving against full.

Baseline	Gain/loss	Success Δ (pp)	Lower (pp)	Token saving	Lower	Joint
Full	6/1	+2.60	-3.28	3,086.1	2,110.9	Yes
Static core	7/4	+1.56	-5.76	1,736.3	1,021.5	No

Pooled saving is $1 - \sum C_i^c / \sum C_i^b$: 34.7427% against full and 23.0500% against static core. Mean per-task relative saving, $n^{-1} \sum (1 - C_i^c / C_i^b)$, is respectively 33.5923% and 22.0263%. Both are descriptive companions to the primary absolute-cost estimate, not alternate primary tests. The old 34.944% mean bundle-relative saving belongs to a different policy and task design.

K.3 Diagnostic motivation and the prospective branch test

A subsequent, explicitly retrospective phase decomposition retains all 576 formal trajectories. It locates 323,823 of the 333,378 token difference between static core and completion core after the first public pass (97.13%). Static core has 94 reads, 59 later public tests, and two actual code changes in this phase, with no original-test repairs. This decomposition does not identify a causal mediation percentage because the arms generated different code and histories.

The next protocol is fixed before any new suffix is sampled. Its eligible set consists of every static-core episode with a first public pass before the eighth response and a still-valid public-pass state. It contains all four old development tasks and 188 of 192 old evaluation tasks; three evaluation tasks never pass publicly and one first passes at the final response. Prefixes include real tool feedback and native returned blocks, and reconstruct workspace bytes and verification hashes. No old private outcome or suffix cost is used to select among eligible prefixes. The inputs are nevertheless previously observed project states, making this a prospective response experiment on a fixed, reused prefix collection.

Each prefix is cloned into retain and withdraw branches. Retain exposes the four core tools; withdraw exposes an empty list at every subsequent call. Both preserve the original system prompt and receive exactly $8 - k$ remaining responses after a k-response prefix. Randomized adjacent branch order uses separate frozen development and evaluation seeds. Initial request payloads must be identical except for tools. Unexposed requests are rejected, not executed; no branch receives private feedback. The eight development branches must pass a technical replay audit before formal collection, without using cost or success direction.

Both branches' cost consists of newly sampled suffix responses, including the full history resent in each request and the normal closing response. Historical prefix-collection costs are not added again. The paired success difference is withdraw minus retain, and the same -5 pp noninferiority margin applies. One cost and one success endpoint use $\alpha = 0.025$ each; the absolute-cost bound uses df=187, and the exact conservative loss bound uses $n = 188$. No new repair, per-action, first-call, or subgroup significance tests are introduced.

The 25 gate-state failures yield zero observed original-test repairs in either arm; among 163 gate-state successes retention has one regression. These are descriptive counts, not a demonstration of zero repair probability. Withdrawal's code is unchanged by construction, while its cost includes all actual responses. Its pooled suffix saving is 67.2953%, and its mean per-task relative suffix saving is

Table 17: All new evaluation suffixes from 188 common prefixes. File changes count actual byte changes, not successful editor-return values.

Measurement	Retain	Withdraw
Gate-state test successes	163	163
Final test successes	162	163
Input tokens (including cache input)	442,323	112,107
Output tokens	49,994	48,904
Total suffix tokens	492,317	161,011
Model responses	357	189
Reads / additional public tests	101 / 60	0 / 0
Accepted edits / actual file changes	7 / 6	0 / 0
Episodes with changed final code	4	0
Original-test repairs / regressions	0 / 1	0 / 0
Natural ends / truncations	183 / 0	188 / 1

56.7696%. The paired absolute mean is 1,762.266 tokens (lower bound 1,504.567). The success difference is $+0.5319$ pp, with conservative lower bound -1.9430 pp; noninferiority is supported, not success superiority.

Formal collection produces 546 responses in 376 branches; including development there are 556 responses, no missing responses, and no unknown usage. Replay verifies the matched starting inputs, histories, tool actions, workspaces, and code-test outcomes. This identifies the effect of future tool exposure at these states; it does not isolate schema length from action availability, validate the history-retention heuristic, or establish a universal withdrawal rule.

K.4 An audited boundary regression

The sole success-discordant pair is MBPP 211, `count_Num(n)`. The shared code passes all three original assertions. Retention's first continuation merges its two boundary cases into $n \leq 1$, returning zero. It then reruns the public test at $n = 2$, which passes, and concludes. The private assertion `assert count_Num(1) == 1` now fails; the other private check at $n = 3$ still passes.

Shared public-pass code	Retain continuation's replacement
`if n <= 0: return 0`	`if n <= 1: return 0`
`if n == 1: return 1`	*special case removed*
`return 2 ** (n - 2)`	`return 2 ** (n - 2)`

Withdrawal leaves the code unchanged and passes all assertions. The benchmark reference also returns one at $n = 1$. Despite imperfect task wording, this is an integer boundary regression under the explicit frozen acceptance convention, not floating-point error. The historical core suffix only reads and summarizes; it is checked for provenance, not substituted for the new retention branch.

The retain suffix uses three responses and 5,912 logical tokens; withdrawal uses one response and 1,168 tokens. Replaying both branches with the original bounded evaluator confirms the regression. Repeated public-test success misses the boundary error; the case alone establishes neither general correctness harm nor an internal overthinking mechanism.

K.5 Earlier independent validation of short retention

Before the completion-core candidate, a separate Flash study tested a gated-history variant on 96 new two-ticket task packages, with full, frontier, history, and gated-history arms. The gate removed the retained tools once all public checks passed, but the unfinished-state tool rule still followed the earlier frontier/lease heuristic. All 384 formal episodes and 2,548 responses completed audit. Full succeeded on 78/96 packages, frontier and history on 77/96 each, and gated history on 76/96. Its mean package-relative token savings against full, frontier, and history were respectively $+16.96\%$, -43.64%, and -11.64%. None of the three frozen success–cost criteria passed.

The Qwen candidate therefore preserves all four core tools until public pass. It changes both policy and deployment, without validating the earlier Flash heuristic. Each separately frozen study retains its own task units, comparators, estimands, and multiplicity correction.

L CONTROLLED SHARED-INTERFACE CHOICE EXPERIMENT

L.1 DESIGN AND FROZEN ANALYSIS

This study follows an exploratory interface-choice observation. Materials are selected before Flash and Qwen collection, and the same materials are reused for GLM under a separately frozen protocol. Both surfaces supply `read`, `str_replace_editor`, `run_tests`, and `write`. The expanded surface additionally supplies `glob`, `grep`, and `todo_write`; the relative order and definitions of shared tools are unchanged. Added schemas jointly change count, length, and content.

Two development and 24 evaluation tasks are selected from a local official MBPP archive, excluding task IDs used in the earlier project experiments and exact prompt/reference duplicates. For each task, the initial state contains an unfinished implementation. The repair state contains a deterministic single-node mutation of a reference implementation that fails the real public test. The task, code, public assertion, and actual public failure are placed in a shared user prefix. This is a constructed state, not a model-generated trajectory or fabricated assistant/tool conversation. The two private assertions never enter the model input.

There are 24 development responses and 576 evaluation responses across Flash, Qwen, and GLM: eight development and 192 evaluation responses per model. Each evaluation crosses 24 tasks, two states, two repetitions, and two surfaces. Paired surface order is randomized and adjacent. Each branch yields one response, with no model continuation after execution feedback. Only the first native proposal is applied offline; other proposals, non-edit actions, errors, empty outputs, and truncations remain in the denominator. The model settings are Flash maximum output 4,096/thinking budget 1,024/effort=max and Qwen and GLM 8,192/1,024/empty output configuration.

The frozen score is $+1$ when the first proposal names `str_replace_editor`, -1 when it names `write`, and zero otherwise, including no proposal. Invalid arguments retain the named-interface score. Paired expanded-minus-common differences are averaged equally across the two states and two repetitions within each task, then across 24 tasks. Each model uses one-sided task-cluster sign flips, 19,999 Monte Carlo draws with fixed seeds, and $p = (1 + \#\{T^* \geq T\})/20{,}000$. The original Flash/Qwen family uses Holm correction at .05; its adjusted p values are .38675 and .04180, respectively. GLM is reported with its nominal p value and descriptive interval; we make no multiplicity-adjusted support claim for this estimate. Descriptive 95% t intervals use df=23 without multiplicity adjustment. States and repetitions are descriptive strata, not additional primary tests.

Table 18: First-proposal results on 24 tasks per model. Intervals are descriptive and unadjusted; p values are nominal, one-sided. The original Flash/Qwen correction and decisions are retained in the text.

Deployment	Mean $S - M$	95% interval	Nominal p
Flash	$+0.0208$	$[-0.0549, 0.0966]$	0.38675
Qwen	$+0.1146$	$[0.0162, 0.2129]$	0.02090
GLM	-0.0104	$[-0.0956, 0.0747]$	0.69255

All 600 accepted responses pass audit. Two prior interrupted Flash/Qwen requests have unknown usage and remain outside the accepted sample; resumption preserves completed outcomes. Qwen passes its original Holm-corrected test and Flash does not. GLM's score difference is close to zero, with a descriptive interval spanning both signs. Its common/expanded surfaces produce 79/85 read proposals and 3/2 responses without a tool proposal. These results do not establish a model interaction. The experiment measures immediate action choice, not later propagation, full execution efficiency, or semantic quality.

Table 19: Interface proposals and separate execution descriptions, 96 responses per surface and deployment. Partial/full replacement describes valid editor arguments, not semantic edit quality.

Deployment	Surface	Write	Editor	Full repl.	Partial repl.	Trunc.	Tests pass
Flash	Common	3	6	0	6	1	9
Flash	Expanded	2	7	0	7	1	9
Qwen	Common	7	45	32	13	3	50
Qwen	Expanded	3	52	33	19	0	53
GLM	Common	6	8	0	8	3	14
GLM	Expanded	4	5	0	5	2	9

L.2 Earlier first-edit endpoint

A separate MBPP opening experiment collected 144 first responses (36 tasks, two repetitions, two surfaces). Its frozen endpoint required the first proposed call to supply a legal implementation passing all three original assertions. Both conditions produced `read` first in all 72 responses, giving a zero-variance, inconclusive contrast. This endpoint floor is not 144 failed complete programming attempts or evidence of equivalent reasoning. That endpoint is preserved rather than replaced by the later independent interface score.

M Capability-matched metadata-order experiment

M.1 Four arms and the ordinary-text control

This experiment tests a prospective first-response prediction on Qwen3.6-35B-A3B, using a maximum output of 8,192 tokens and a thinking budget of 1,024. Every condition supplies the same seven tools in the same order: `read`, `str_replace_editor`, `run_tests`, `glob`, `grep`, `write`, and `todo_write`. Names, parameter schemas, executable behavior, and the shared editing interfaces are unchanged. Only the order of factual sentences in the `grep` description and a user-message reference block varies.

All descriptions start with the identical sentence `Search current text files for a literal substring; path may be dot.` They then permute three identical sentences: `Each match reports the line number.`, `Each match reports the line text.`, and `Each match reports the file path.` Order L is number–text–path, G is path–number–text, and N is number–path–text. Thus N denotes an interleaved order, not an empirically neutral description. The actual `grep` implementation returns all three fields. No condition has a different tool implementation or receives execution feedback before responding.

The user-message block always has the heading `Workspace interface reference (grep):`. It is ordinary interface documentation, not execution feedback or an instruction to prefer an editing action. Table 20 specifies the four arms. In the tool-description contrast, the user reference stays at N; in the ordinary-text contrast, the tool description stays at N. The same L/G sentence-order manipulation is therefore evaluated at two request locations.

Table 20: Metadata-order conditions. L, G, and N contain exactly the same factual sentences; only their order differs.

Arm	Tool description	User reference	Manipulated location
TL	L	N	Tool description
TG	G	N	Tool description
CL	N	L	Ordinary user text
CG	N	G	Ordinary user text

For each task and repetition, the system instruction, task, complete current code, real public-test failure, workspace, and inference settings match across arms. Each changing field separately has the same UTF-8 byte length, character multiset, word multiset, and factual-sentence inventory; the complete requests also match in byte length and character/word multisets. All 256 formal four-arm

blocks have exactly equal reported native total input tokens, including the corresponding cache input fields. This is whole-request equality, not an independent measurement of each field's token count. These controls rule out tool-count, executable-capability, and measured input-length differences within this experiment; they do not separate a tool-specific neural mechanism from other effects of request location or presentation.

M.2 Materials, execution, and frozen endpoint

The materials comprise eight development tasks and 128 evaluation tasks, with respectively two and 32 instances from each of four fixed generated families: record transformation, grouped aggregation, string parsing, and threshold boundaries. They use new parameter instances excluding the 68 earlier history–demand instances and within-study duplicates. This is instance-level validation within four families, not a new-domain or MBPP evaluation. Each task has a deterministic fault in a unique target function, at least two substantive non-target functions, three public checks, and 104 private checks. Initial programs fail the public checks; frozen reference implementations pass the complete checks. Private checks and reference implementations are not supplied to the model.

Development contains $8 \times 4 = 32$ responses and uses only a technical gate, with no success-rate, edit-rate, direction, or variance requirement. Evaluation contains $128 \times 2 \times 4 = 1{,}024$ responses. Task/repetition blocks and within-block arm orders are randomized before collection; the four arms of a block are collected adjacently. Two repetitions do not create two independent tasks.

Each arm produces one model response. Only its first native proposal is executed in a separate workspace; execution feedback is recorded but never returned to the model. Additional proposals, non-edit actions, invalid calls, and truncated responses remain in the denominator. A complete legal proposal can still be scored when its response is truncated. The binary endpoint is one only if the proposal legally edits the target file, actually changes it, passes every public and private check, and preserves the AST outside the uniquely identified target function. Source-location attributes are ignored in the AST comparison. Either editing interface can succeed: a whole-file `write` is eligible if it preserves the non-target structure. This endpoint is successful *first-response repair with non-target structural preservation*, not minimal editing, reasoning quality, or success after a complete agent trajectory.

M.3 Prespecified directional tests and complete results

Let $\bar{Y}_{ia}$ average the two repetitions for task i and arm a. The frozen contrasts are

$$D_{T,i} = \bar{Y}_{i,\mathrm{TL}} - \bar{Y}_{i,\mathrm{TG}}, \qquad D_{C,i} = \bar{Y}_{i,\mathrm{CL}} - \bar{Y}_{i,\mathrm{CG}}, \tag{20}$$

$$I_i = D_{T,i} - D_{C,i}. \tag{21}$$

The two primary predictions are $\mathbb{E}[D_T] > 0$ and $\mathbb{E}[I] > 0$. Each uses a one-sided paired-task Student t test with 128 tasks and df=127, with Bonferroni $\alpha = .025$ per endpoint. Both adjusted one-sided 97.5% lower bounds must exceed zero. This is approximate mean inference, not an exact randomization test. The interaction compares signed differences, not their absolute magnitudes. Zero sample variance would make the test unestimable and fail the joint criterion; neither observed primary contrast has zero variance.

Table 21: Both original positive-direction predictions fail. Values are proportion differences; the lower bound is the Bonferroni-adjusted one-sided 97.5% bound.

Contrast	Mean	SD	SE	Lower	Raw p	Adjusted p
D_T	−0.1328125	0.31675256	0.02799724	−0.18821398	0.99999723	1
I	−0.1171875	0.43237294	0.03821673	−0.19281151	0.99867609	1

The observed D_T is -13.28125 percentage points and I is -11.71875 points; both are opposite to the prespecified positive direction. The descriptive ordinary-text difference is $D_C = -1.5625$ points. The saved, unadjusted two-sided 95% intervals are $[-18.8214, -7.7411]$ points for D_T and $[-19.2812, -4.1563]$ for I. These descriptive intervals are not a post hoc reversal of the confirmatory alternative. The frozen joint decision remains `JOINT_NOT_SUPPORTED`.

Table 22: Complete first-proposal counts, 256 responses per arm. Repair is the joint primary endpoint. Tool columns describe the first proposal, not complete execution strategies.

Arm	Repair	Read	Grep	Tests	Editor	Write
TL	207	45	2	2	207	0
TG	241	13	0	2	241	0
CL	220	35	0	0	221	0
CG	224	31	0	1	224	0

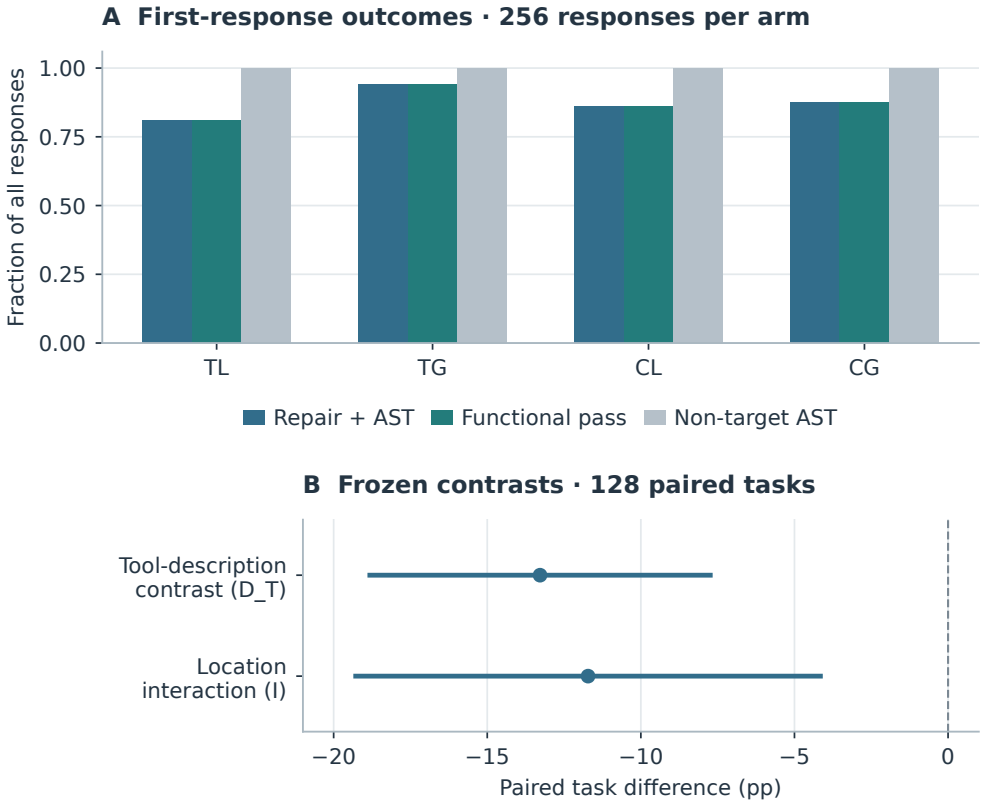

Figure 13: Saved metadata-order results. (A) First-response repair, functional pass, and non-target AST preservation on the complete response denominator. (B) The two frozen contrasts with their saved descriptive, unadjusted 95% task intervals. Both observed contrasts oppose the original positive-direction predictions; neither primary test passes. Structural preservation reaches a ceiling in every arm.

M.4 What the result does and does not establish

The first-response repair rates are 207/256 (80.86%) for TL, 241/256 (94.14%) for TG, 220/256 (85.94%) for CL, and 224/256 (87.50%) for CG. Across all 1,024 responses, 893 legal edits all use `str_replace_editor`, and 892 pass the complete functional checks. The other 131 first proposals are `read`, `run_tests`, or `grep`; no first proposal uses `write`. Non-target AST structure is preserved in all 1,024 responses, including unchanged programs. No responses are truncated, and the saved tool-error count is zero.

The structural-preservation component therefore has a ceiling, while the joint endpoint varies and equals functional success on these observations. Its arm differences primarily reflect whether the first response directly edits or inspects, with one unsuccessful editor action. They do not demonstrate a choice between whole-file rewriting and local modification, nor a mechanism improving structural preservation. An initial inspection is a zero on this frozen first-response endpoint, not a demonstrated failure of the full programming task. No later execution was collected in this study.

The equal-capability controls sharpen the observation, but they do not rescue the failed directional hypothesis. A reverse first-action pattern is a candidate for a new, independently frozen prediction, not confirmation of the original one. Nonsupport also does not establish zero effect, equivalence of request locations, or a complete ordinary-text explanation. The study does not test withdrawal, long-term propagation, information-sufficiency interactions, or a beneficial dynamic controller.

All 32 development and 1,024 evaluation responses pass request, input-matching, workspace-replay, and accounting audits. Collection required explicitly recorded, bounded resumptions after interrupted requests: 1,059 physical attempts yielded 1,056 accepted responses, with three attempts of unknown usage retained separately. Previously accepted responses were preserved, the original deadline and scientific definitions were maintained, and physical-attempt limits were explicitly amended. These are disclosed operational deviations; unknown usage is not treated as zero. No intermediate effect analysis, alternate endpoint, additional test, or subgroup selection enters the reported result.

N Completion instructions and retrospective stopping

N.1 Actual prompt equality

The shared system instruction includes: “When the current work is complete, respond without a tool call. Use public tests to check your work.” This is a generic completion requirement, not an instruction to treat the first public pass as definitive success. The full, core, completion-core, and earlier two-arm branch requests retain it. That earlier two-arm experiment does not add a closing instruction at the gate or remove the original requirement.

The saved requests confirm identical system instructions across collection and branching. In all 188 evaluation and four development pairs, the first requests differ only in tools: native history, code, public-pass feedback, and inference parameters match. The observed branch difference therefore does not result from giving one arm an additional closing instruction.

The only other external user text is the initial task and the uniform continuation instruction after a no-tool max-token truncation: “Continue the current ticket. Use the available tools to complete and verify it.” Its occurrence is endogenous to the rollout, not a randomized closing-prompt arm. No model-generated completion phrase is recoded as an exogenous instruction. The audit adds no model responses or effect tests.

N.2 Deterministic truncation of saved trajectories

After the complete results are known, define a direct-stop reference at the first successful public-test response whose current code still has valid passing verification. Keep that response’s full reported cost and the resulting code; discard all later model calls. If no such state occurs, retain the complete original episode. Private outcomes do not select the stopping time. This construction changes no pre-gate input or tool policy and needs no unobserved model response. It does not simulate a model that was told a different rule earlier.

The analysis includes all 576 formal trajectories and their 2,747 response records. All 563 reached gates match the current code and its verification state. Every episode retains its original three-assertion label. This observed equality is neither a new noninferiority test nor a guarantee about untested inputs. The completion-core episode that succeeds without public feedback is retained normally.

Table 23: Post hoc stopping on each policy's own 192 observed trajectories. No new policy responses are sampled and no confidence bound is fitted.

Policy	Success unchanged	Observed tokens	Stop tokens	Saving	Responses
Full	162/192	1,705,473	1,225,939	28.12%	$949 \rightarrow 639$
Static core	164/192	1,446,324	971,316	32.84%	$977 \rightarrow 631$
Completion core	167/192	1,112,946	961,761	13.58%	$821 \rightarrow 634$

At the 188 paired gate states, direct stopping preserves 163 successes and requires zero new model tokens; it produces no final model explanation. The sampled withdraw branch has the same 163 successful labels, while retention loses one. Twenty-five gate failures remain failures in both sampled branches. The zero continuation cost excludes the already incurred common prefix cost. These counts do not establish withdrawal's superiority to direct stopping or compare final-summary utility, which was not measured.

N.3 Separate original inference from the added reference

Figure 14 preserves the original whole-task and sampled-branch comparisons, including the unsuccessful whole-task noninferiority comparison to static core. The new stopping reference adds no significance tests and changes none of those frozen results. Likewise, the 99.67% input share of the retain-minus-withdraw token difference is an arithmetic decomposition: input includes schemas and repeated history, which are not independently intervened on. It is not a schema-only causal effect.

These earlier two-arm data support exposure effects under the generic completion instruction but do not identify an exposure-by-new-closing-instruction interaction. Appendix O adds that randomized factor. Neither study identifies the value of final explanations, a predictive response kernel, or the incremental control value of history beyond current state. Model-generated completion language and post hoc subgrouping could not supply the instruction contrast absent from that earlier study.

O Factorial completion-instruction experiment

O.1 Frozen design and collection

Motivated by the completed two-arm and direct-stop analyses, this experiment freezes a new four-arm comparison before its first model call. It reconstructs the same four development and 188 evaluation static-core public-pass prefixes from their source responses and tool events. Eligibility remains first valid public pass with at least one of the original eight responses left: three evaluation tasks never pass publicly and one first passes on response eight. No private outcome, prior suffix cost, or case narrative selects prefixes. The prior suffixes do not enter the new controls or tests. This is prospective response collection on previously observed states, not an unseen-task evaluation.

The factors are original versus explicit-closing instructions and retained versus withdrawn tools. Retention supplies read, replacement editing, testing, and write in their original order; withdrawal supplies an empty list. The explicit instruction is exactly: "End this ticket now with a final response. Do not call any tools or modify any files." It is appended as one text block after the unchanged tool-result blocks in the final user message. All system text, code, original native history, public feedback, and remaining budget are preserved. It imposes neither brevity nor a special output format and does not declare public pass equivalent to correctness.

Within each prefix, the four arms are collected adjacently in shuffled order; task and arm order use separate frozen development and evaluation seeds. Every arm is newly sampled. Qwen uses the model settings in Appendix C. Each suffix may use only the original remaining response budget. At most the first tool proposal per response executes; extra or unavailable calls are rejected with native

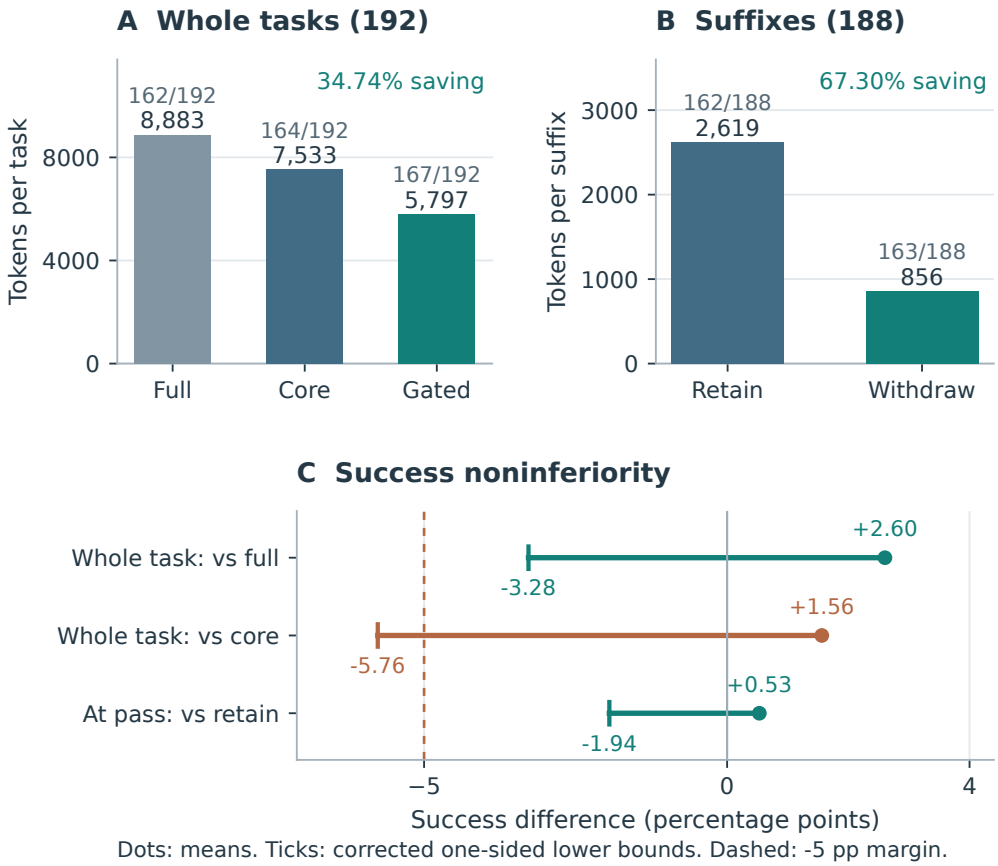

Figure 14: Two separate execution estimands. (A) Whole-task mean tokens and test-success counts on 192 project-new tasks per policy. (B) Suffix tokens and success counts from 188 previously observed public-pass prefixes, with two newly sampled branches each. Within each panel the percentage is the reduction in pooled tokens; panels have different denominators and cost windows. (C) Prespecified success-difference lower bounds against the -5 pp noninferiority margin. Cost bounds are positive for all three comparisons, but the whole-task static-core comparison fails its success condition. Error markers in C are one-sided lower bounds, not two-sided intervals.

feedback. Retention proposals are executed even if they conflict with the closing instruction, so the harness does not impose withdrawal on that arm.

A no-tool natural end terminates the suffix. After a no-tool max-token truncation, original arms receive the prior continuation message and closing arms receive the same closing instruction again. Thus the randomized factor is a closing-prompt policy; no truncation occurs in the completed formal sample. Four development quadruplets pass full technical replay before evaluation, without using effect direction or obedience rates as a gate.

All 752 evaluation branches produce 938 accepted responses; the 16 development branches bring the total to 958. Every branch completes the replay audit, with no missing response or unknown usage. Checks reconstruct requests, tool events, workspaces, code-test outcomes, and four-way initial-input equality apart from the two randomized factors. Private tests never enter model history or experimental control.

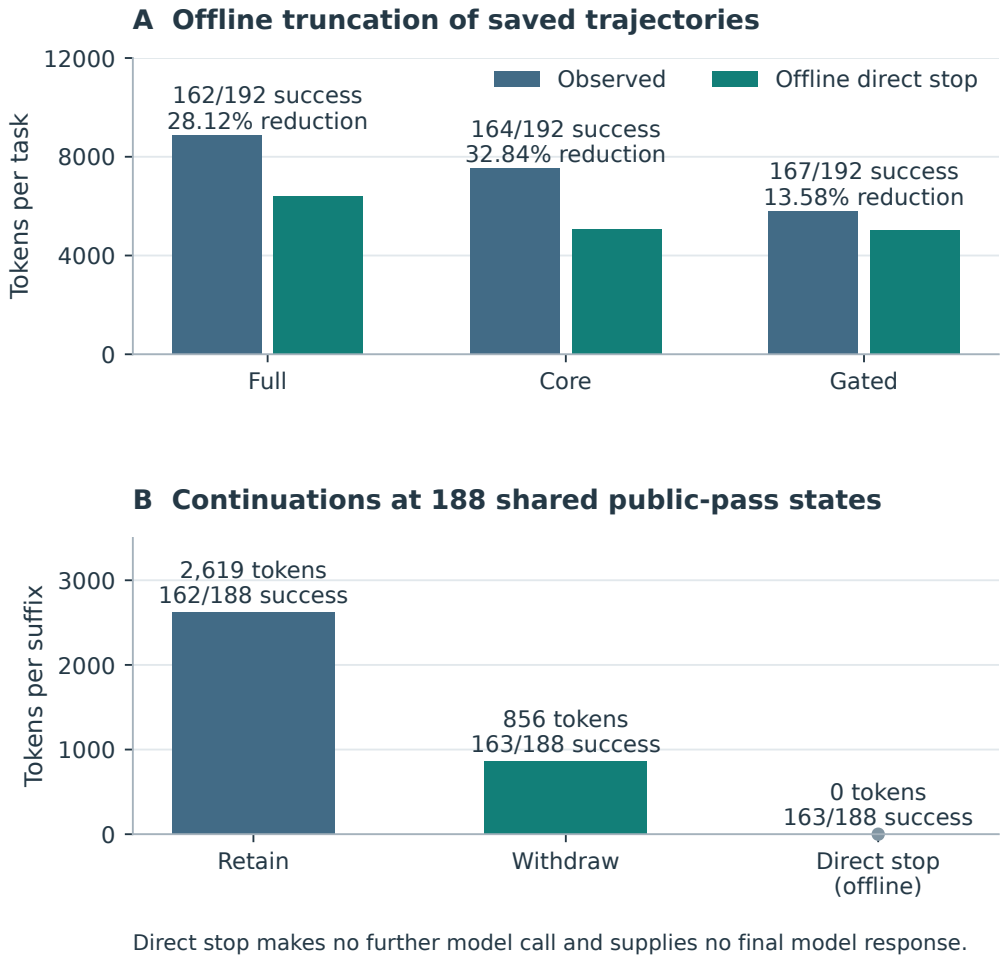

Figure 15: Observed continuations and retrospective stopping references. (A) Whole-task tokens before and after deterministic truncation of each saved trajectory at its first valid public pass; each policy includes all 192 tasks. Its success count is unchanged. (B) New retain/withdraw continuations from the same 188 prefixes, compared with preserving the gate code and making no further model call. Zero is continuation cost only. Direct stopping omits a final model summary; its utility is unmeasured. This reconstruction adds no significance tests.

O.2 Three prespecified endpoints

For prefix i, let T_{iqs} be reported suffix tokens, where $q \in \{O, C\}$ denotes original/closing instruction and $s \in \{R, W\}$ retained/withdrawn tools. Define

$$d_i = T_{iCR} - T_{iCW}, \qquad z_i = (T_{iOR} - T_{iOW}) - (T_{iCR} - T_{iCW}). \tag{22}$$

The first endpoint is the mean of d_i, testing residual savings under closing; the third is the mean of z_i, testing attenuation of the exposure-related cost difference. Each uses a one-sided Student-t lower bound over all 188 prefixes, df=187; zero variance is insufficient for a positive decision. The second endpoint uses the conservative loss-only success bound in Appendix K, now with $L = \sum_i \mathbb{1}\{Y_{iCR} = 1, Y_{iCW} = 0\}$ and the same -5 pp noninferiority margin.

The three endpoints share Bonferroni $\alpha = .05/3$, giving individual one-sided 98.3333% bounds. Residual cost–success support requires endpoints one and two jointly; the interaction is reported separately. The family is not redefined after observing results. Original-instruction simple effects, other prompt contrasts, ratios, actions, and natural endings are descriptive only. Every task, including failed or budget-exhausted suffixes, remains in the denominator.

Table 24: New factorial sample: 188 branches per arm. Input includes provider-reported cache input. Every final code-test label matches its gate label.

Instruction	Tools	Success	Input	Output	Total	Responses
Original	Retain	163	442,431	52,428	494,859	360
Original	Withdraw	163	111,543	42,632	154,175	188
Closing	Retain	163	234,937	34,293	269,230	202
Closing	Withdraw	163	113,675	78,503	192,178	188

Table 25: Frozen main tests, individual one-sided 98.3333% lower bounds. Cost units are tokens per prefix; success uses percentage points.

Endpoint	Estimate	Lower bound	Required	Result
Closing: retain–withdraw cost	409.85	292.95	> 0	Pass
Closing: withdraw–retain success (pp)	0.00	-2.15	> -5	Pass
Cost difference-in-differences	1,402.30	1,133.17	> 0	Pass

The descriptive pooled savings are 68.84% under the original instruction and 28.62% under closing; corresponding mean per-task ratios are 60.08% and 24.21%. The latter do not replace the absolute-cost primary test. Closing reduces total cost by 45.59% when tools are retained, but increases it by 24.65% when tools are withdrawn; these extra simple contrasts receive no new inferential tests. Code success is 163/188 in every arm, with zero label changes in all 752 suffixes, and the closing comparison has zero discordant losses or gains. Its conservative lower bound is -2.1543 pp, not zero uncertainty.

Original-retain produces 181 tool proposals, 101 reads, 68 public tests, 10 accepted edits, three file-change events, and 179 natural endings. Closing-retain produces 16 proposals, eight reads, seven tests, one accepted write, zero file-change events, and 186 natural endings. Both withdrawal arms have no tool proposals and 188 natural endings. These are descriptive counts. The next section audits the closing-retain events without selecting on effectiveness.

The direct-stop reference preserves the gate code with 163 successes and zero new model calls, but no final response. Nonempty visible text is measured mechanically, not scored for semantic quality. This comparison cannot establish the utility of summaries or superiority of a continuation policy over direct stopping.

P Action replay and cost accounting

P.1 Exhaustive action audit under explicit closing

We conducted a post-hoc descriptive audit of all 188 formal `closing_retain` branches, including branches without tools and those with unsuccessful code. All 11 branches with native proposals received detailed replay, without filtering by cost or outcome. No model calls, annotation, hypothesis tests, or revised endpoints were introduced.

We matched all 202 actual requests to their saved histories and verified that each contained the closing instruction: "End this ticket now with a final response. Do not call any tools or modify any files." At the first request, this text was appended to the final user block containing the genuine public-test feedback. For the 11 action-producing branches, the original restricted executor replayed 38 prefix events from starter code, all 16 suffix events, and the original code tests. The source records remain unchanged.

Complete action inventory. Eleven of 188 branches (5.85%) proposed tools; the remaining 177 proposed none. All 16 proposals were accepted executions: eight reads, seven public-test calls, and one write. Table 26 includes every action-producing branch. All eight reads accessed `solution_1.py`. Earlier reads concerned different versions; searching the complete preceding history confirmed zero same-hash matches. These are not repeated reads of unchanged content. Their utility for the final summary was not evaluated.

Episode	MBPP	Prefix	Actions	Responses	End	Gate/final
045	636	`core_556`	R	2	Natural	Pass/pass
143	652	`core_285`	R	2	Natural	Pass/pass
183	757	`core_401`	R	2	Natural	Pass/pass
194	661	`core_528`	R	2	Natural	Fail/fail
200	620	`core_506`	R, T	3	Natural	Pass/pass
272	640	`core_196`	R	2	Natural	Pass/pass
343	659	`core_509`	R	2	Natural	Pass/pass
404	617	`core_334`	T	1	Budget	Pass/pass
505	383	`core_394`	T, T, T, T, T	5	Budget	Pass/pass
638	626	`core_051`	W	2	Natural	Fail/fail
677	146	`core_288`	R	2	Natural	Fail/fail

Table 26: All 11 action-producing closing/retain branches, accounting for all 16 proposals. R: read; T: public test; W: write. Actions retain chronological order; responses include any final response without tools. Budget denotes exhaustion of the remaining response allowance. Gate/final uses the complete original code tests, not public tests alone.

The seven test calls, spread across three prefixes, reran the same public test on unchanged, already passing code; all passed again. The accepted write supplied exactly the existing 39-character file content. Neither it nor any other branch changed file bytes. Across all 188 prefixes, complete code-test success remained 163/188: zero repairs among 25 initially unsuccessful prefixes and zero regressions among 163 initially successful prefixes. Nine action-producing branches ended naturally; two exhausted their remaining allowance. Overall natural endings were 186/188. Budget exhaustion is distinct from a naturally terminating response.

Action-only illustrations. Two cases illustrate explicit post-hoc selection rules, without claiming representativeness. First, we selected the most repeated tests of unchanged code, breaking ties by smallest episode identifier. Episode 505 (MBPP 383) had passed its public test after prefix response three and retained five responses. After receiving the closing instruction, it called `run_tests` in each of those five responses. Every call passed without a code change; the branch then exhausted its allowance. Complete code-test success remained true. Second, we selected the smallest episode with an accepted write, the unique such branch. Episode 638 (MBPP 626) wrote the existing content unchanged, then ended naturally without another tool call. Its complete code tests failed both before and after the suffix despite the public test passing.

These records identify continued action despite an instruction prohibiting tools, without establishing internal motives or useless reads. The 11 branches form an outcome-defined subset; their behavior alone cannot explain the aggregate token contrast. The following accounting retains all 188 paired prefixes.

P.2 Complete token accounting

This post hoc audit retains all 188 tasks and all four new suffixes; it adds no confidence intervals, significance tests, semantic labels, or fitted models. Every one of the 938 formal responses is reconciled with provider usage, response bytes, recorded events, episode totals, and the frozen result. For each fixed instruction, all 188 first-request pairs match except for tools. Their reported input-token difference, including cache input, is exactly 531 in every pair, totaling 99,828. It is a provider-accounting increment for the tools field, not a separate measurement of an internal behavioral mechanism.

Write each arm's suffix cost as $T = I_1 + O_1 + L$, where I_1, O_1 are first-response input/output tokens and L is all later-response cost. With retain-minus-withdraw differences, the exact identity is

$$\Delta T = \Delta I_1 + \Delta O_1 + (L_R - L_W). \tag{23}$$

It holds separately for every task, without residual. Its aggregate components are:

Under closing, the 14 additional retention responses contribute 21,434 input and 1,668 output tokens. Across all responses, withdrawal reduces input by 121,262 but increases output by 44,210. Thus

Table 27: Descriptive accounting over all 188 prefixes. Positive components favor withdrawal. Later cost combines additional calls with their schemas, evolving history, and output; it is not an independently identified mediator.

Instruction	First input	First output	Later total	Net total
Original	$+99,828$	$-14,287$	$+255,143$	$+340,684$
Closing	$+99,828$	$-45,878$	$+23,102$	$+77,052$

the 77,052-token net saving is not a reduction in generated output. Reported output includes all provider-counted output and does not isolate internal reasoning. The invariant first-input difference, the generated-output response, and the extra-call component should not be collapsed into a claim of reduced reasoning effort.

Why a fixed first-request surcharge cannot explain the interaction. For every prefix, the retain-minus-withdraw first-input difference is 531 tokens under both instructions. Subtracting that fixed amount from each simple contrast therefore leaves its difference-in-differences unchanged:

$$[(T_{iOR} - T_{iOW}) - 531] - [(T_{iCR} - T_{iCW}) - 531] = z_i. \tag{24}$$

The total interaction is exactly $263{,}632 = 0 + 31{,}591 + 232{,}041$ tokens, where the terms are the first-input, first-output, and later-response interactions. The original mean interaction of 1,402.30 tokens and its corrected lower bound of 1,133.17 consequently also apply after this arithmetic subtraction; no new test is fitted. The directional interaction was specified before collection. Interpreting it as rejecting a fixed-surcharge-only explanation is an additional accounting argument, not a separately preregistered mechanism test.

This argument rules out only an instruction-invariant first-request surcharge as a complete explanation. It does not remove length effects on generation or later requests. Under explicit closing, the mean net saving of 409.85 tokens is smaller than 531: the accounting remainder is -121.15, not a positive saving attributable solely to behavior. The experiment therefore establishes a conditional cost interaction, not improved reasoning quality or a schema-free behavioral benefit.

The accounting does not independently randomize schema length, action availability, later history, or response count. Therefore it cannot assign a causal mediation percentage to these components, infer that all residual savings are a pure schema effect, or extrapolate a conditional average among the 11 action-producing prefixes to the full population. The randomized factorial contrast remains the main evidence; this audit specifies what its recorded costs contain.

Q Prospective history–demand–interval validation

Q.1 Prediction and evaluation population

This experiment tests a directional extension of the earlier Qwen shared-interface observation: whether an expanded initial surface leaves an interface-choice bias that is greater for a new editing requirement than for verification, and whether this demand difference is smaller after two background exchanges. These are new predictions about historical exposure, not a direct repetition of the earlier removed-tool endpoint. Materials, allocation, readout, and analysis were frozen before collection.

Four fixed task families cover record transformation, grouped aggregation, string parsing, and threshold boundaries. Each contributes one development instance and sixteen evaluation instances, giving four development and sixty-four evaluation tasks. Development and evaluation parameter combinations do not overlap; source and prompt duplicates are excluded by fixed offline rules. These are generated programming instances, not an MBPP sample or a random sample of task families.

Each task provides a correct implementation of an old specification, two public assertions, and a reserved flag whose new behavior is activated only by the editing requirement. Offline construction checks require the starter to satisfy the old specification and fail all eight new-requirement assertions; the reference implementation must satisfy both specifications. Independently constructed expected values and two fixed erroneous variants check the test materials. Private assertions, reference

implementations, and oracles never enter model requests or tool feedback. Both Flash and Qwen use the model settings in Appendix C.

Q.2 Eight branches with identical external starting states

For each task and deployment, two genuine preparation responses start from identical context and workspace, differing only in the declared surface. Surface M exposes `read`, `str_replace_editor`, `run_tests`, and `write`; S adds `glob`, `grep`, and `todo_write`. Only the first proposed call is eligible for execution; invalid or extra calls receive recorded error feedback. Native assistant blocks and corresponding tool results are preserved. Histories are retained regardless of preparation success, tool use, or termination.

Each M/S history is cloned into immediate and delayed prefixes. The delayed prefix receives exactly two fixed, read-only background checkpoints under M, each requesting the sum of supplied numbers in an independent workspace. The immediate prefix receives neither checkpoint. This manipulation changes intervening interaction and context length; it does not isolate elapsed wall-clock time. All background histories are constructed before the target branches, whose order is randomized.

Each history–interval prefix then forks into an *edit* demand activating the new rule and a *verify* demand retaining the old specification. Both conditions contain a current task. All eight target branches start with byte-identical canonical old code, public tests, and README, with empty read records, work plans, and test caches. The request explicitly announces a new independent workspace. Thus the initial historical intervention cannot carry forward a different target file or verification state. Previous surface declarations are not automatically appended to subsequent messages; historical differences reach the target through the preserved generated messages and genuine feedback.

All target calls expose the same four tools in M. Each branch allows at most three responses and evolves independently; its outcomes never update another branch or its parent history. Natural completion terminates the branch. A truncated response without a call receives a fixed continuation prompt within the same response limit. Errors, invalid arguments, extra proposals, no-tool responses, and truncations remain included. Allocation randomizes task–deployment order, preparation order, background order, and the four interval–demand pairs; M/S target branches are adjacent within each pair with randomized order.

Q.3 Frozen readout and four primary tests

The first `tool_use` in the first target response scores $+1$ for `str_replace_editor`, -1 for `write`, and zero otherwise, including no tool. Invalid arguments retain the proposed interface score, and later responses do not replace the first observation. The score measures interface choice; replacement editing can rewrite an entire file and does not imply locality or quality.

For task i, interval ℓ, and demand d, let $D_i(\ell, d)$ be the S-history minus M-history score. Define

$$C_{E,i} = D_i(\text{immediate}, \text{edit}) - D_i(\text{immediate}, \text{verify}), \tag{25}$$

$$C_{L,i} = D_i(\text{delayed}, \text{edit}) - D_i(\text{delayed}, \text{verify}), \tag{26}$$

$$I_i = C_{E,i} - C_{L,i}. \tag{27}$$

Each deployment tests $\mathbb{E}[C_E] > 0$ and $\mathbb{E}[I] > 0$. The four one-sided paired-task t tests use 64 task contrasts, 63 degrees of freedom, and a single Bonferroni family with $\alpha/4 = 0.0125$. A deployment meets the joint criterion only if both tests pass; replication of that joint pattern across the two deployments requires all four. Zero variance never yields a pass. The inferential units are generated parameter instances, conditional on the four fixed construction families; neither the 1,024 probes nor their individual assertions are independent task replicates. The t reference assumes independence across instances and is not an exact finite-sample randomization test.

Q.4 Complete results and interpretation

Table 28 and Figure 16 report all four tests. Neither deployment meets its joint criterion. Qwen's descriptive immediate/edit S-minus-M difference is $+0.28125$, versus -0.046875 after the background exchanges; its C_E and C_L means are $+0.265625$ and -0.046875. This is a mean sign reversal, not

Table 28: All four frozen history–demand tests. Intervals are unadjusted two-sided 95% intervals; lower bounds are Bonferroni one-sided 98.75% bounds. All corrected lower bounds are below zero and all tests fail their support criterion.

Deployment	Contrast	Mean	95% interval	Corrected lower	Adjusted p
Flash	C_E	0.218750	$[-0.109722,\ 0.547222]$	-0.158688	0.376089
Flash	I	0.046875	$[-0.374012,\ 0.467762]$	-0.436754	1.000000
Qwen	C_E	0.265625	$[\ 0.005187,\ 0.526063]$	-0.033637	0.091479
Qwen	I	0.312500	$[-0.054173,\ 0.679173]$	-0.108834	0.186957

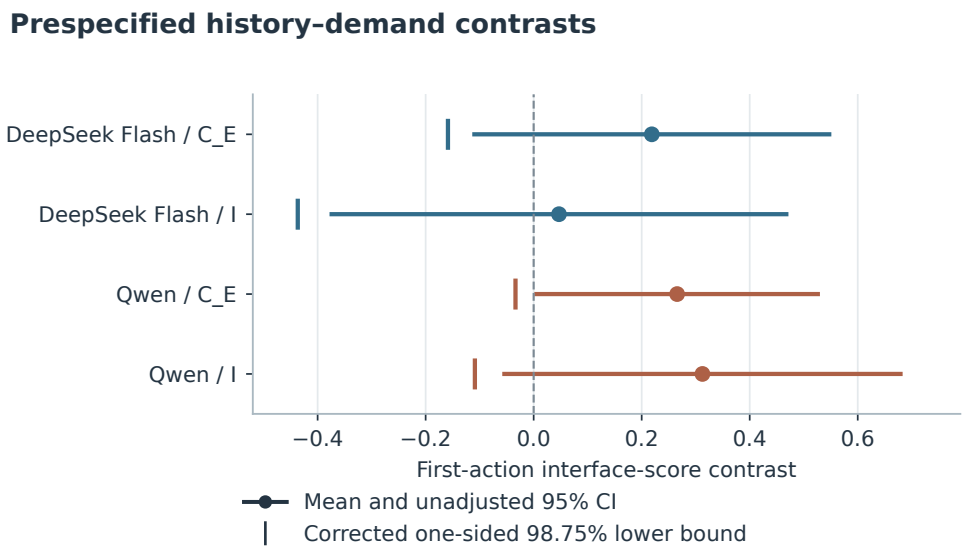

Figure 16: The four prespecified contrasts from sixty-four paired tasks per deployment. Dots are means, horizontal segments are unadjusted two-sided 95% intervals, and vertical ticks are Bonferroni one-sided 98.75% lower bounds. The unadjusted Qwen C_E interval excludes zero, but its corrected test does not pass.

confirmed attenuation. Flash's corresponding demand differences are $+0.21875$ and $+0.171875$. No deployment-interaction test was specified.

Even a positive C_E would not establish a positive immediate/edit simple effect. A positive I would establish a larger signed demand difference at the immediate checkpoint, not a smaller absolute historical effect, disappearance, or monotonic decay. Failure of the frozen tests does not establish zero effects or equivalence. The sample size alone does not guarantee adequate power, and neither nominal significance nor an exploratory subgroup can replace the corrected support rule. These results constrain the proposed extension to new tasks and the shared-interface endpoint while retaining the earlier withdrawal studies as evidence for their own different outcomes.

Private evaluation tests each final program against ten common regression assertions and eight assertions for its assigned demand. The functional score is the mean of the two pass fractions; complete success requires all eighteen assertions. Function, resources, simple history differences, and C_L are descriptive outcomes, evaluated on all sixty-four tasks per arm. Figure 17 displays the interface and functional means separately. No-tool or failed probes retain their final code and denominator. Near-ceiling functional scores do not establish equivalence or convert interface differences into execution benefits.

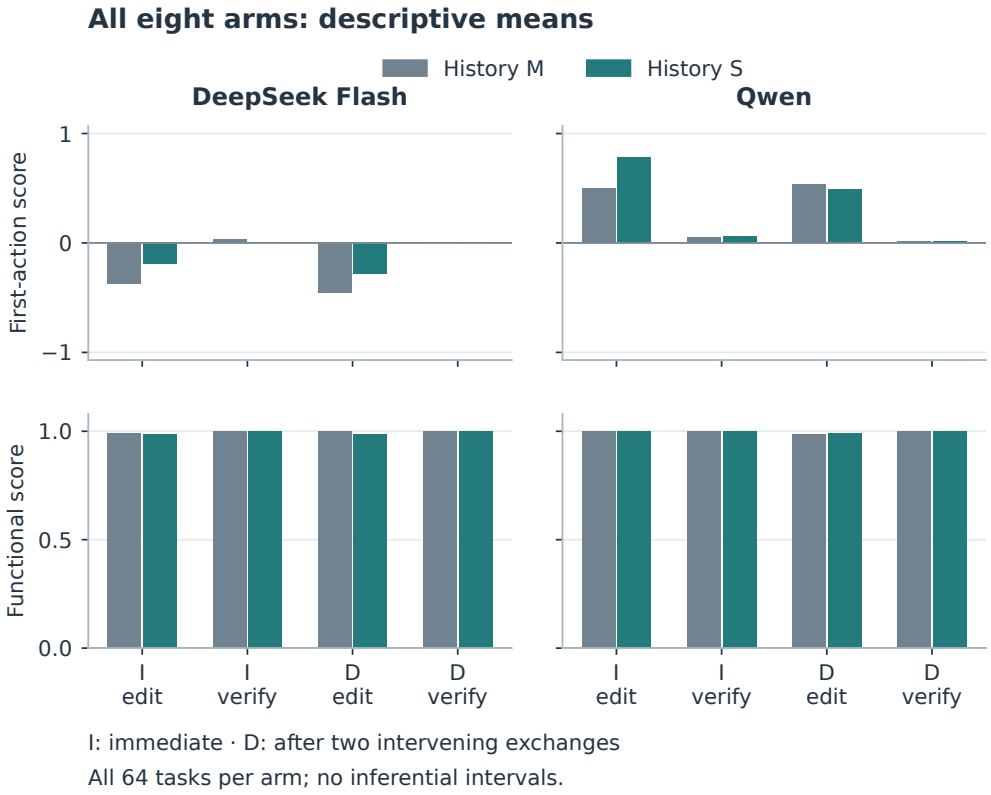

Figure 17: All eight target conditions per deployment, retaining sixty-four tasks in every arm. Upper panels show first-proposal interface scores; lower panels show the separately measured private functional scores. Bars are descriptive means without inferential intervals. The two outcomes are not interchangeable, and these panels add no tests to Table 28.

Q.5 Sample completeness

The technical development gate checks all sixty-four development probes, faithful request/history/workspace replay, common target starting states and future tools, and complete accounting and private scoring. It does not select on effect direction, interface frequency, or success. Development used 215 responses: 167 within target probes and 48 constructing preparation and background histories. Evaluation used 3,405 responses: 2,637 within the 1,024 target probes and 768 constructing those histories. All probes completed and passed replay and accounting audits before the frozen analysis ran.

Including development, the campaign accepted 3,620 responses and reported 8,523,001 logical input-plus-output tokens, including reported cache input. Four interrupted requests have unknown usage; these are not zero-cost observations. Saved native histories and workspaces permit continuation without replacing accepted outcomes. All continuations belong to one campaign and retain its original sampling and analysis rules. Reported tokens exclude unknown charges.

R Exploratory direct exposure contrasts and editing-interface audit

R.1 Reuse of the completed B/P/E evaluation

This analysis reuses the complete evaluation data from Appendix E: 48 task bundles, two repetitions, and two deployments, giving 192 units and 4,118 actual model responses. It introduces no new model calls or task executions. Its analysis plan was fixed before these additional computations, after the

original results were known; it is therefore exploratory and leaves all eight original primary tests unchanged.

Here B uses the common surface throughout, P uses the expanded surface initially and withdraws it afterward, and E continues expanded exposure. Both surfaces include `read`, `str_replace_editor`, `run_tests`, and `write`; the expanded surface adds `glob`, `grep`, and `todo_write`. The P/E branches share the actual $h = 0$ response, tool feedback, and workspace. Their first subsequent requests differ in current exposure; later differences can also include the resulting history and workspace changes. P/B instead compares downstream consequences of the initial intervention under common future tools. Neither comparison isolates an internal memory mechanism.

The additional family comprises four comparisons per deployment. The direct sixteen-dimensional comparison averages cross-repetition products of $(E - P)$ vectors over $h = 1$–7 using the original fixed weights. It estimates direct squared mean separation, distinct from the original H4 difference between two distances to B. The action-sequence representation combines two equally weighted blocks: one-hot first successfully executed tool categories at each of seven positions, and frequencies of adjacent category transitions. Its nine categories distinguish both editing interfaces, no execution, and absorption. Both distance tests use the 48 task bundles as clusters, one-sided t tests with df=47, and retain negative finite-sample estimates.

The two text comparisons decode E versus P and P versus B from current-response thinking at $h = 1$–7. Exact tool names are removed after NFKC normalization and case folding; schemas, accumulated history, action JSON, feedback, and $h = 0$ text are excluded. Four folds keep entire task bundles and both repetitions together. Character TF-IDF uses 3–5-grams, minimum document frequency two, at most 4,000 features, and sublinear term frequency. Logistic regression uses $C = 1$, balanced class weights, the liblinear solver, and at most 1,000 iterations, without tuning on held-out effects. One classifier per fold is shared across the later positions.

Information is $\log 2 + \log p_{\text{true}}$, averaged equally over bundles after averaging repetitions, both arms, and seven positions within each bundle. Empty thinking and absorbing positions receive probability $1/2$ and information zero; they are not treated as actual text responses. Balanced accuracy is separately described on real nonempty thinking. Each text test uses 199 paired whole-trajectory label swaps within bundle–repetition units, refitting within the same folds, and an add-one one-sided permutation p value. This diagnostic relies on paired exchangeability; it is not a finite-sample exact randomization test of the adaptive research program. The eight additional tests use a single Holm correction, with positive information or separation and adjusted $p \leq .05$ required for exploratory support.

R.2 All eight saved comparisons

Table 29: Complete exploratory comparison family. Intervals are unadjusted 95% bundle-cluster descriptive intervals. The distance estimates use the original representation scales; text information is in nats. Text p values come from paired label-swap diagnostics, whereas distance p values come from the specified cluster t tests. Positive text information measures predictive information, not an $E - P$ directional mean in a semantic attribute.

Deployment	Comparison	Estimate	Descriptive interval	Raw p	Holm p
Flash	Direct E/P, 16-dim.	0.003707	$[-0.008716, 0.016129]$	.27560	.49756
Flash	E/P action sequence	0.024926	$[-0.030640, 0.080491]$	.18572	.49756
Qwen	Direct E/P, 16-dim.	0.007186	$[-0.005046, 0.019418]$	.12160	.48640
Qwen	E/P action sequence	0.036706	$[-0.038582, 0.111994]$	.16585	.49756
Flash	E/P text information	-0.000246	$[-0.006134, 0.005642]$	.04500	.22500
Flash	P/B text information	0.002399	$[-0.010363, 0.015161]$	.02500	.15000
Qwen	E/P text information	0.005512	$[-0.001406, 0.012430]$	.00500	.04000
Qwen	P/B text information	0.006470	$[-0.002559, 0.015500]$	.01000	.07000

Only Qwen's E/P text comparison meets this exploratory support rule: information is $+0.005512$ nats, adjusted $p = .040$, and balanced accuracy on its 1,155 nonempty responses among 1,344 logical positions is 53.37%. Its unadjusted task-cluster interval nevertheless spans zero. The paired

label-swap reference and the cluster-mean interval address different references and assumptions; the former does not establish that the latter excludes a zero population mean. With only 199 swaps, the smallest attainable add-one p value is .005. The evidence is a weak exploratory signal under sustained versus withdrawn exposure, not an independent confirmation of historical persistence. A length-only auxiliary readout has information -0.000622; no contrast between the two readouts was tested, so a length-related explanation is not eliminated.

Neither P/B text comparison passes the family correction. Flash's E/P information is negative despite its nominal label-swap $p = .045$ and does not pass the positive-information requirement. All four direct-distance and action-sequence tests fail. No selected-lag or attenuation tests are added. These findings preserve the distinction between ongoing exposure differences, post-withdrawal effects, and the task utility of either.

R.3 Replacement editing does not establish local modification

The initial action summaries motivated a separate descriptive audit of editing parameters and workspace changes. It checks all 4,118 physical responses, the shared starting histories and files in all 192 P/E unit pairs, and 1,750 logical successful editing events. The event count includes the shared $h = 0$ in both logical regimes and is not an independent sample count. Table 30 instead excludes $h = 0$ and uses only $h = 1$–7. A full-old-text replacement means that, after trimming surrounding whitespace, `old_str` equals the entire previous file. A partial-old-text replacement does not by itself imply semantic locality or quality.

Table 30: Successful editing calls after the shared initial response, 96 trajectories per deployment and regime. Counts describe interface use and parameter coverage; they are not independent test observations.

Deployment	Regime	`write`	Replacement editor	Full old text	Partial old text
Flash	B	258	13	9	4
Flash	P	252	19	11	8
Flash	E	229	41	32	9
Qwen	B	46	233	233	0
Qwen	P	73	232	232	0
Qwen	E	20	284	284	0

Flash's $E - P$ count difference is -23 for `write` and $+22$ for replacement editing; 21 of the additional replacement calls match the entire old file. Qwen's corresponding differences are -53 and $+52$, with all additional replacements matching the entire old file. The replacement increase mainly occurs when editing the initial task template: 20 of 22 additional calls for Flash and 48 of 52 for Qwen. These are count decompositions rather than event-matched substitution rates. Replacing the whole old text does not mean every output character changed, just as writing an entire file does not imply every line changed.

The interface-count direction is positive in both repetitions for both deployments, but its magnitude varies, and not all task bundles move in the same direction. No additional significance test was introduced for this selected count pattern. The earlier eight-call four-cell studies cannot serve as a matched replication of the shared-interface choice: their common surface omits `write`, whereas their expanded surface includes it. The subsequent independent first-proposal experiment in Appendix L supplies the appropriate shared-interface comparison. The defensible observation here concerns how already-available interfaces are used, not more local code changes, improved correctness, lower cost, or a universal temporal law.

S Additional archived generation and history-readout evidence

These studies preserve earlier evidence outside the main comparisons. We distinguish an archived result report, a recomputation of saved predictions, a completed confirmatory contrast, and a stopped development study. This revision does not retrain their classifiers or recreate their original uncertainty procedures.

DSH Bridge model-variant study. An archived DeepSeek Pro study uses eight code tasks, five repetitions, and two surfaces, yielding eighty feedback-free first responses. The pair surface exposes `bash` and `str_replace_editor`; the discovery surface exposes three discovery-related tools. Its masked-text, leave-one-task-out analysis reports balanced accuracy and AUROC of 1.0, information 0.387739 nats, a task-bootstrap 95% interval $[0.376562, 0.399504]$, and sign-flip $p = 0.0081$. The archived process and primary support gates pass. This adds historical first-response evidence, with its original inferential and provenance scope; it is not a new replication performed for this revision. Count, descriptions, length, and available capabilities differ together, and no continuation or task-success outcome is measured.

Persistent exposure within a common stage. The earlier Flash initial-by-persistent factorial contains 160 trajectory records over eight tasks and five repetitions, with two shared initial donors per block and 720 physical requests. Saved-prediction recomputation gives persistent-surface information of 0.157788 and 0.197130 nats during the two maintenance calls. Initial-surface information at those calls is 0.090215 and 0.031636 nats. These same-stage differences remain legacy evidence for observable exposure signatures; they should not be discarded merely because a later contrast is confounded. Once all branches use Standard, persistent-label information is 0.015172 and 0.011177 nats at the two washout calls. However, the checkpoint prompt changes with that transition. The cross-stage reduction therefore does not isolate schema withdrawal, and a nonsignificant later score would not establish equality with chance. These are checks of saved predictions, not an independent reproduction of all training and uncertainty calculations.

A separate stage-cue comparison. The v2.10 experiment examines how an explicit stage-ending sentence changes the decodability of earlier Minimal/Standard exposure when the current surface is fixed to Standard. Its thirty-six tasks yield 288 primary endpoints. The unique confirmatory target is the difference in decoder information between absent and present cue conditions, not absolute historical decodability. The task-average difference is $+0.005677$ nats, with two-sided complete-refit permutation $p = .515$ and approximate 95% percentile interval $[-0.026685, 0.033294]$. The condition scores, 0.010162 and 0.004485 nats, are descriptive. Thus the cue contrast is not detected; neither equivalence nor absence of historical information follows. Explicit history remains in the input, and the added sentence does not isolate a pure semantic reset or clear internal memory.

Stopped development does not supply negative effects. Three subsequent development efforts yield no complete scientific readout. Two accept no endpoint because requests are incompatible or interrupted. The third accepts one prefix but stops before continuation because another prefix violates the nonempty-returned-thinking rule. These technical or protocol-compatibility outcomes are not null treatment effects; empty returned thinking does not imply absent internal computation. They add no independent samples to the completed stage-cue or later text analyses.

T STUDY LINEAGE AND EARLIER CONTROL DEVELOPMENT

T.1 WHAT CONSTITUTES A SEPARATE SOURCE OF EVIDENCE

We preserve the distinction between a frozen prospective comparison, an exploratory evaluation, a development gate, an offline reanalysis, and an archival result report. Passing a replay audit establishes consistency between saved requests, responses, tool execution, and reported outcomes; it does not make an exploratory comparison confirmatory. Recovery directories contain continuations and copies of one experiment, rather than additional independent samples. Likewise, reanalyzing a saved trajectory with a new readout does not create another trajectory.

Table 31 maps the six completed execution studies used in the current evidence chain. Development observations are excluded from the evaluation estimates. Later candidates were motivated by earlier failures, but their rules and analyses were fixed before their own evaluation responses were collected. This makes them separate candidate tests, not a retrospective repair of earlier outcomes. The public-pass studies reuse observed static-core prefixes: their newly collected suffixes support conditional intervention comparisons on those states, not new-task replication. Project-new MBPP IDs also do not establish absence from model training data.

Table 31: Execution-study lineage. Counts are episodes or suffix branches, not independent model calls. Evaluation units are the task or bundle clusters used for inference; within-unit repetitions and arms are not additional clusters. All development gates are kept separate from efficacy estimates.

Study / deployment	Development	Evaluation	Unit and evidence scope
Small-task policies / Flash	16	96	8 synthetic tasks, 3 repetitions, 4 policies; exploratory.
Three-ticket MBPP / Flash	24	144	12 bundles, 2 repetitions, 6 policies; exploratory.
Gated history / Flash	16	384	96 new two-ticket bundles, 4 policies; frozen candidate validation.
Completion core / Qwen	12	576	192 new single-ticket tasks, 3 policies; frozen whole-task validation.
Public-pass fork / Qwen	8	376	188 existing core prefixes, 2 newly sampled suffixes each.
Closing factorial / Qwen	16	752	The same 188 prefixes, 4 entirely new suffixes each.

The three-ticket campaign also collected 144 separate feedback-free first responses for RQ1. They do not enter its execution denominator. The six campaigns contain respectively 461, 1,819, 2,658, 2,806, 556, and 958 accepted model responses, including development and, for the second campaign, that separate opening study. These counts describe collection, not a pooled statistical sample. The completion-core campaign includes one additional failed attempt with unknown usage; recovery does not convert that cost to zero. The two subsequent suffix experiments neither rebill nor count the old prefix responses as newly collected observations.

T.2 Different cost estimands answer different questions

Let C_i^b, C_i^c denote baseline and candidate cost for the relevant unit. The mean relative saving $n^{-1}\sum_i(1 - C_i^c/C_i^b)$ weights units equally, whereas pooled saving $1 - \sum_i C_i^c / \sum_i C_i^b$ weights the relative reductions by baseline cost. Neither is interchangeable with the mean absolute saving $n^{-1}\sum_i(C_i^b - C_i^c)$. For the three-ticket study, costs are averaged over the two repetitions within each bundle before computing the bundle ratio. Table 32 keeps the four headline percentages attached to their original estimands. The newer studies use absolute saving for the primary cost bound; their percentages are descriptive companions to that test.

Table 32: Four savings with different denominators. These values must not be averaged or treated as repeated estimates of one policy effect. The full success–cost results appear in Appendices J, K, and O.

Saving	Comparison and scope	Estimand and joint status
34.94%	History vs. full; 12 three-ticket bundles, Flash	Mean bundle-relative whole-task saving; cost component supported, success noninferiority not established.
34.74%	Completion core vs. full; 192 tasks, Qwen	Pooled whole-task saving; frozen joint criterion passes.
67.30%	Withdraw vs. retain; 188 public-pass prefixes, Qwen	Pooled suffix saving in the earlier two-arm study; joint criterion passes.
28.62%	Withdraw vs. retain under explicit closing; 188 prefixes, Qwen	Pooled suffix saving in the new four-arm study; closing-condition joint criterion passes.

All four initial small-task joint comparisons, all three three-ticket retention comparisons, and all three later gated-history comparisons fail their respective criteria. Completion core passes against full exposure, but its whole-task comparison with static core still fails success noninferiority despite lower cost. The common-prefix experiments do not retroactively change that result. Direct stopping remains a zero-suffix-call reference with the same observed code-test outcomes; no experiment here measures the semantic utility of the omitted final summary.

T.3 Development feasibility and evaluation entry

Development gates assess protocol feasibility separately from evaluation effects. In the information-state design, Qwen proceeds from 48 development episodes to 768 evaluation episodes; GLM completes 16 development episodes but does not enter evaluation. Both use four information-state/presentation conditions, seven tools, a common surface after the first call, and at most six responses per episode. Their gates pool presentations within each information state and require at least one state to have success strictly between zero and one, without selecting on the presentation effect.

GLM uses four existing development tasks, one per family, selected by fixed within-family ordering before collection without using Qwen outcomes. It uses a maximum output of 8,192 tokens and a 1,024-token thinking budget, with no temperature or additional output-effort setting. The 16 episodes require 94 responses and yield 8/8 successes in each information state. This ceiling triggers the frozen stop before the planned 32-task, 128-episode evaluation. Neither directional prediction is tested; the development result establishes neither equivalence nor model superiority. The anonymous supplement preserves all development records and the stopping decision, separately from evaluation observations and earlier demand-timing data.

T.4 Archival development results retained for completeness

The following earlier experiments belong to the development history of trajectory-aware control. This revision checked their archived result reports and preserved their stated evidence levels; it did not newly reconstruct their raw interaction records or rerun their uncertainty procedures. Their counts are therefore not merged with Table 31, and their original positive interpretations are not automatically adopted as current claims. In particular, no single favorable comparison establishes a generally useful history-based controller.

Periodic re-anchoring and midstream transfer. Two periodic re-anchoring studies each report 120 trajectories: eight tasks, five repetitions, and three conditions. Both compare a narrow versus broad read-only planning checkpoint after a shared prefix, with the same additional call budget. Their matched success differences have opposite signs ($+5$ and -5 percentage points), and both intervals span zero. The strict process gates fail because of four and one noncompliant tool choices, respectively; workspace protection remains intact. Comparisons with no checkpoint also add computation and cannot identify a narrow-surface benefit. A subsequent 160-branch, history-retain/reset by profile experiment passes its process gate, but neither the common-surface probe nor its reset interaction passes the transfer criterion. Its local re-anchoring effect therefore does not establish an effective periodic controller.

MPC and learned residual policies. An archived eight-task, three-repetition MPC evaluation reports a positive FIR-over-memoryless utility contrast, while the stronger capability-frontier baseline has higher observed utility and success than FIR. Subsequent residual-policy development fails its direction or utility gates. The expanded success-readout analysis contains 24 tasks and 72 pairs and still underperforms its frontier comparator; the event-triggered history policy likewise reports lower success than frontier. These results preserve the distinction between a history model altering decisions, outperforming a particular memoryless implementation, and adding value beyond a strong current-state policy. The earlier training/calibration examples are not independent confirmation data.

Late contraction and capability guards. An exact-prefix late-contraction report uses 96 blocks on 24 tasks, with its principal contrast restricted by a frozen eligibility rule. Although its cost and success-noninferiority gates pass, a separate gain-versus-harm direction gate fails, leaving its composite mechanism label false. A subsequent capability-guard study similarly reports lower last-step token cost and success noninferiority but fails its tool-error safety gate. The selected-block effects do not describe all tasks, and the composite failures cannot be replaced by the favorable cost component. Reused final calls in ineligible or identical-action records are not independent responses.

ATFR and CFR revisions. The archived ATFR confirmation has 12 tasks, three repetitions, and five conditions. The designated v1.3-versus-matched-frontier success contrast fails its -5-point safety threshold, and its total-token interval includes zero; a favorable exploratory v1.2 contrast does

not replace that endpoint. CFR15 search probes are small pilots or success-ceiling evaluations and principally establish route reachability. Different provider settings remain separate. The CFR17 evaluation explicitly reuses eight prior calibration tasks and is exploratory; its total-cost intervals against the reported late-contraction and always-standard controls include zero. CFR19 is an offline audit and replay result with only three route-divergent records among 342 inputs, including six synthetic examples. It receives a no-go decision for a new efficacy experiment; replaying old terminal labels cannot supply unobserved outcomes under the new controller.

Taken together, this lineage explains why the current paper centers the independently evaluated state gate and the controlled suffix experiments, while retaining unsuccessful history and retention policies. RQ3 establishes bounded execution-cost results without requiring a universal history-decay law. Conversely, those cost results cannot validate a failed RQ2 prediction or demonstrate that a learned response kernel improves tool-selection decisions.

U Independent information-state by presentation validation

U.1 Prediction and design

The metadata-order study motivated two prospective predictions: line-first presentation would increase initial inspection, and its final-quality advantage over path-first presentation would be larger when relevant facts had to be retrieved. We tested these predictions in a separate, frozen, complete-execution study using Qwen3.6-35B-A3B. The model used a maximum output of 8,192 tokens and a thinking budget of 1,024. This experiment tests the two directional predictions below; it does not fit the additive log-odds candidate in Appendix B.6.

Every condition exposes the same seven tools, in the same order, with identical implementations and parameter schemas: `read`, `str_replace_editor`, `run_tests`, `glob`, `grep`, `write`, and `todo_write`. At the first call only, the `grep` description places its identical factual sentences in order L (line number, line text, file path) or G (file path, line number, line text). All subsequent calls use one common order (line number, file path, line text), retaining the actual generated history and tool feedback. This common order is an experimental reference, not an empirically established neutral presentation.

The second factor supplies relevant external facts in the initial prompt (S) or leaves them to be retrieved from read-only workspace files (M). At the same prompt position, S includes two true records for the target profile, whereas M includes equally long, true but irrelevant audit records from the same files. The snapshot blocks match in bytes. Initial code, workspace, task objective, true facts, and final tests are otherwise identical. The missing-facts prompt does not explicitly require inspection, and the supplied-facts prompt does not prohibit it. Within each information state, the paired L/G first requests match in length and character and word multisets; their reported first-response input-token counts also match. Token equality is not assumed across information states or over complete generated episodes. This factual-record substitution changes content and potentially attention; it does not isolate a pure information mechanism from every such alternative.

Twelve development tasks (48 episodes) precede 192 evaluation tasks (768 episodes), with one episode per task and arm. Each of four fixed generation families—grouped aggregation, record transformation, string parsing, and threshold boundaries—contributes 48 evaluation tasks. These are new parameter instances, not new domains. Every task requires at least two external facts; automatically checked counterfactual task worlds share the missing-facts prompt and public assertions but require different answers on a specified input. The model can read the true fact files, while submitted programs cannot obtain the answer through file I/O. Private tests do not enter the model context. The development feasibility gate pools presentations within each information state and stops if both state-specific success rates are in $\{0, 1\}$; it does not select on a presentation effect.

U.2 Complete execution and frozen readouts

Episodes receive at most six accepted model responses. Each response executes only its first native tool proposal, with actual feedback appended to the history. Only the designated solution file may change. Invalid and unavailable calls remain recorded, and natural completion or exhaustion of the

fixed budget ends the episode. Final success requires all fixed public and private tests to pass and all protected files to remain unchanged.

The initial-inspection indicator A equals one when the first native proposal names `read`, `grep`, `glob`, or `run_tests`, regardless of argument validity or execution success. No-call, invalid, non-editing, failed, and truncated episodes remain in the full denominator. Thus A measures a proposed action category, not actual information acquisition or an internal cognitive state. The quality indicator Q measures the actual final code-test outcome under the common continuation policy. Table 33 reports all four arms; all 768 evaluation episodes are included.

Table 33: Information-state evaluation, 192 episodes per arm. Counts retain every episode. Tokens are mean reported input-plus-output tokens over each complete episode and are descriptive. S/M denotes supplied/retrievable relevant facts; L/G denotes first-call line/path-first presentation.

Arm	Initial inspection	Final success	Truncated episodes	Tokens/episode
SL	118/192 (61.46%)	180/192 (93.75%)	1	12,803.04
SG	128/192 (66.67%)	176/192 (91.67%)	2	12,959.23
ML	188/192 (97.92%)	172/192 (89.58%)	2	15,144.42
MG	188/192 (97.92%)	170/192 (88.54%)	3	15,487.08

The paired task is the inference unit. For task i, the two frozen contrasts are

$$A_{\text{inspect},i} = \tfrac{1}{2}\{(A_{SL,i} - A_{SG,i}) + (A_{ML,i} - A_{MG,i})\}, \tag{28}$$

$$I_{\text{quality},i} = (Q_{ML,i} - Q_{MG,i}) - (Q_{SL,i} - Q_{SG,i}). \tag{29}$$

Both predictions specify a positive mean. Each uses a one-sided paired-task Student t approximation with 191 degrees of freedom, with Bonferroni $\alpha = .025$ per test. Joint support requires both corrected one-sided lower bounds to exceed zero. We retain the original directions, endpoints, and correction.

Table 34: The two frozen directional tests. Effects and corrected one-sided 97.5% lower bounds are percentage-point differences. Neither test passes.

Contrast	Mean (pp)	Corrected lower (pp)	One-sided p	Adjusted p
A_{inspect}	-2.6042	-7.6889	.84316	1.00000
I_{quality}	-1.0417	-7.7152	.62074	1.00000

U.3 Results and interpretation

The complete study does not support either prediction (Table 34). The descriptive $L - G$ inspection differences are -5.2083 percentage points with facts supplied and zero with facts to retrieve. The corresponding final-success differences are $+2.0833$ and $+1.0417$ points. These simple contrasts are descriptive; they do not establish a quality benefit or an inspection-mediated effect. The saved, unadjusted two-sided 95% task intervals are $[-7.6889, +2.4806]$ points for A_{inspect} and $[-7.7152, +5.6319]$ for I_{quality}.

There are 4,013 evaluation responses, including all eight episodes with at least one truncated response. Protected facts remain unchanged in every episode. Complete request, history, workspace, and scoring replay passes before analysis. Interrupted collection resumes only missing observations, preserving the same evaluation sample; recovery is not another replication. Including development, the study contains 816 episodes and 4,258 accepted responses. We report complete-episode token differences descriptively and make no efficiency or noninferiority claim from this study.

These results constrain the proposed extension of the metadata observation: its selected presentation ordering does not establish an independent initial-inspection or final-quality interaction prediction under these task states and response budget. They do not prove zero effect or equivalence, identify a competing mechanism, or test a universal decay law. In particular, the broad state-conditioned formulation can accommodate these observations, but that flexibility is not predictive validation. The separately supported closing-instruction cost interaction retains its own intervention and scope.

V ADDITIONAL RELATED WORK

Tool-augmented agents and interfaces. ReAct interleaves reasoning and actions, while Toolformer learns API use (Yao et al., 2023; Schick et al., 2023). ToolLLM and Gorilla study retrieval and calling over large API collections (Qin et al., 2024; Patil et al., 2024); AutoTool and ToolAnchor learn tool selection or adaptation (Zou et al., 2025; Liu et al., 2026). SWE-agent and Agent S emphasize interfaces for executable interaction (Yang et al., 2024; Agashe et al., 2025). We instead intervene on exposure while preserving shared actions and the agent's execution rules.

Surface sensitivity and control. Competing descriptions and metadata can change tool preferences (Faghih et al., 2025; Blankenstein et al., 2026); unnecessary tools can suppress direct answers (Paturi et al., 2026). Attention-based accounts distinguish attending to a tool from selecting it (Chen, 2026). CMTF exposes a precondition–effect frontier and withdraws tools at goal satisfaction (Babu & Iyer, 2026). Our public-test gate applies this class of state-dependent control. We test choices among unchanged interfaces and cross exposure with closing instructions at identical prefixes, separating policy cost from a controlled exposure effect.

Observable reasoning and history. Chain-of-thought prompting generates intermediate reasoning (Wei et al., 2022), but this text need not faithfully identify the causes of an answer (Turpin et al., 2023; Lanham et al., 2023). Reflexion retains textual feedback across trials (Shinn et al., 2023); context position also affects performance (Liu et al., 2024a). TSCTD therefore distinguishes returned text, retained interaction state, and measured execution outcomes rather than identifying them with internal reasoning.

Executable evaluation. API-Bank and BFCL evaluate tool calls, while AgentBench and SWE-bench assess interactive task completion (Li et al., 2023; Patil et al., 2025; Liu et al., 2024b; Jimenez et al., 2024). StableToolBench addresses API and evaluation instability (Guo et al., 2024). Our MBPP adaptation uses local execution and paired workspaces to isolate surface interventions. Interface choice, test success, and cost remain separate endpoints; direct stopping is a distinct reference that also removes the final model response.

W HISTORICAL ACTION EFFECTS: SUMMARY VISUALIZATION

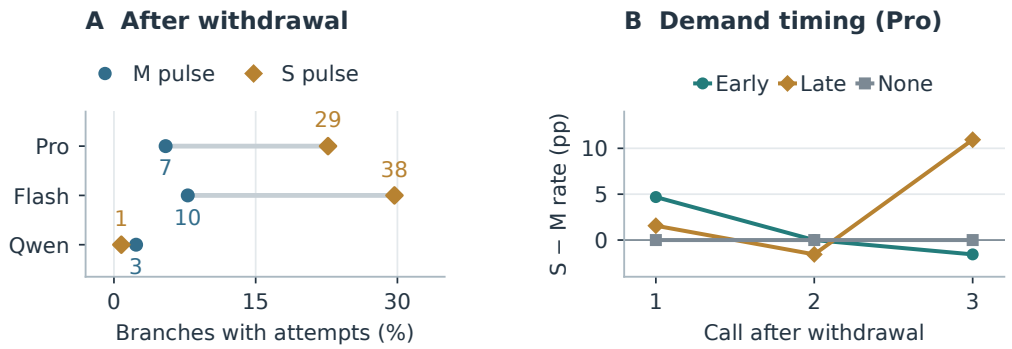

Figure 18: Exploratory historical action evidence. (A) Any removed-tool attempt at $h = 1$–3, with 128 branches per pulse arm and model. Lines join descriptive rates, not confidence intervals. (B) Separate Pro timing study: per-call S-minus-M attempt rates, 64 branches per cell; early/late demands arrive at $h = 1/3$. All unavailable calls are rejected.