CONTINUE, REVISE, RESTART, OR REPLACE: A CONTROLLED NULL ON PER-STATE INTERVENTION IN STALLED AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

An operator whose long-horizon agent has stalled has four options: let it *continue*, supply advice and let it *revise* in place, restart it with a clean context (*fresh*), or replace it with a successor built from a handover package (*new*). Which is best cannot be read off agent logs: a log records one intervention at a state and never its counterfactual, so replacement always arrives confounded with better advice and with a cleaner context. We separate the three factors with a 2×2 design in Craftax in which all four arms resume from one evaluator-only snapshot of a genuinely stalled run, under the same remaining budget and the same frozen executor, with a byte-identical 512-token suggestion shared by the advised arms and a byte-identical $1{,}024$-token handover package shared by the rebuilt arms. The result is a null: across two executors differing $3.6\times$ in task competence and $14{,}712$ paired branches, choosing among the four actions per state yields no measurable gain, the cross-fitted oracle ceiling being $+0.007$ achievements (95% CI $[-0.063, +0.081]$, permutation $p = 0.43$) with the weak executor and -0.001 ($[-0.119, +0.121]$) with the strong one. One contrast resolves, in the weak regime only: advice without context repair harms ($R - C = -0.156$, $[-0.247, -0.065]$), and rebuilding the context cancels that harm. A variance decomposition makes the null interpretable rather than empty, since 76% of apparent between-cluster variance is repeat noise and the minimum detectable effect is 0.129 achievements in the weak regime and 0.190 in the strong one, so smaller effects are unresolved rather than absent. We also withdraw a $+22\%$ oracle ceiling we had reported internally — it was almost entirely max-of-k bias — and give the permutation-null diagnostic that catches it.

1 INTRODUCTION

This paper reports a null result. On two executors differing $3.6\times$ in competence, we could not measure a gain from choosing state by state among four standard responses to a stalled agent. We believe the design is sound and the effect, if any, smaller than it can resolve; both are worth pressing on.

The decision. When a long-horizon agent stalls late in its budget, deployed systems do all four Guo et al. (2026); Rombaut (2026): let it *continue* (C); hand it a targeted suggestion and let it *revise* in place (R); rebuild it from a compact handover package, a *fresh* start (F); or replace it with a *new* session built from that package and given the same suggestion (N). The choice is heuristic, or appeals to context degradation Liu et al. (2024).

Logs cannot settle it. A trajectory records what happened after one intervention, not what a *different* intervention would have produced at the same state. When a successor takes over and the task completes, three accounts stay inseparable: the successor helped, the suggestion handed over with it helped, or discarding the predecessor's reasoning helped. Traces and multi-agent leaderboard gaps Cemri et al. (2025) inherit the confound.

What we built. A 2×2 factorial — the C/R/F/N grid, keep or rebuild context against withhold or supply advice — on *one real stalled state*. All four arms resume from one evaluator-only snapshot of

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

a real predecessor run, under the same budget and frozen executor; none resets the world. R and N share a byte-identical suggestion, F and N a byte-identical handover package (Section 3). We also fit a budget-aware manager, scored held out against five baselines.

What we found. Experiment 1 uses a weak executor (354 map clusters, mean 4.0 of 67 achievements); experiment 2 a strong executor (397 clusters, two to six repeats per checkpoint, mean 14.4). Across both, $N - C$, $N - F$ and $F - C$ are null. The cross-fitted ceiling — an oracle picking the best action per state — is $+0.0073$ $[-0.0630, +0.0808]$, $p = 0.429$, in experiment 1 and -0.0013 $[-0.1188, +0.1213]$, $p = 0.509$, in experiment 2. On 160 held-out test maps no baseline separates from always continuing.

One contrast resolves, and only in the weak regime: advice without context repair harms, $R - C = -0.1562$ $[-0.2467, -0.0648]$, and rebuilding the context cancels it, $N - R = +0.1395$ $[+0.0624, +0.2150]$. Not two findings: $N - R = (N - C) - (R - C)$ by construction, $N - C$ is unresolved at -0.0167, and both follow from the R arm scoring low. Only $R - C$ is pre-registered primary; $N - R$ is exploratory. On the infrastructure-clean subset $R - C$ is -0.1033 $[-0.2108, +0.0055]$, no longer excluding zero; we do not claim it survives contamination removal. In the strong regime $R - C$ is $+0.0578$ $[-0.0622, +0.1757]$; its opposite sign is not a finding: the estimate shrank monotonically toward zero as clusters accumulated (Section 6).

Why the null is interpretable. 76% of the apparent between-cluster variance in paired differences is repeat noise (within-checkpoint sd 1.566 achievements against a true between-cluster sd of 0.450), putting the minimum detectable effect at 80% power at 0.129 achievements in experiment 1 and 0.190 in experiment 2. $R - C$ clears the first bar; nothing observed clears the second. Effects under 0.129 are outside this instrument's reach, and we do not claim they are zero.

Numbers we withdraw. An earlier internal version reported an oracle ceiling of $+1.02$ achievements ($+22\%$) and a learned manager capturing 4.1% of it. Both are withdrawn, along with a claim that no management baseline beats doing nothing: the first was almost entirely the upward bias of a maximum over four noisy arms, the second was scored in-sample, and the third was never supported in either direction (§6.6).

Where a reviewer should push. At the seven pressure points we list in Appendix K.1: the capability floor, the single environment, B5 as a weak context-management opponent, the untrained manager, the $162/400$ contaminated experiment-1 prefixes, the test split touched before the freeze, and the weak executor's disabled reasoning field. Section 7 states each as a threat with its numbers.

Scope. No component here is new Sutton et al. (1999); Packer et al. (2023); Vezhnevets et al. (2017), and management is not worthless: only per-state choice among these four actions, under this instrument, did not pay.

2 Related Work

Every component of this instrument is borrowed and **nothing in this section is claimed as new**; our only unborrowed contribution is the null measurement in §6.

The harness as a learned object. Optimising the scaffold around a frozen model, and learning that controller from logged interaction, are established (Shao et al., 2026; Yi & Song, 2026); neither crosses the interventions nor pairs data at a state, so neither separates *replacing* the executor from the accompanying advice and clean context.

In-run correction. Mid-run correction is the standing rival account for any handover benefit (Shinn et al., 2023); the R arm is that account under matched cap and charge, and its negative effect (§6.1) is a fact about one generator in one regime, not a verdict on the technique.

Context management. Rewriting and merging the working context explains anything a rebuilt session buys (Yi et al., 2026), hence F as its own cell. Our comparator $B5$ is an **explicitly labelled**

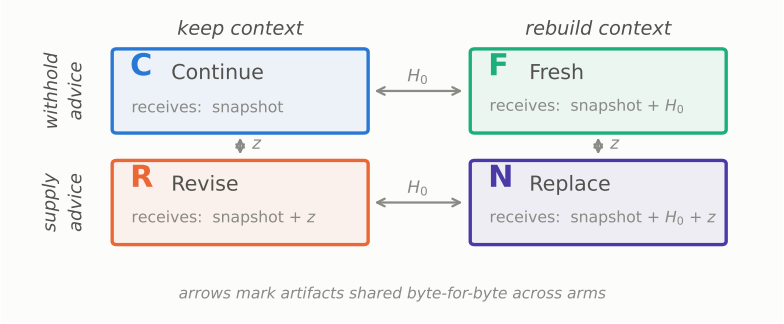


Figure 1: **The four management actions.** Columns cross keeping against rebuilding the working context; rows cross withholding against supplying a targeted suggestion. Each cell lists what that arm receives. Arrows mark the artifacts shared byte-for-byte — one H_0 between F and N, one z between R and N — which is what makes a between-arm difference attributable to the intervention rather than to a differently worded handover. No arm resets the world.

weak substitute — an untrained summarisation rule, as no trained editor exists in this checkout (§G.3) — so its last place in Table 1 is not evidence against learned context management.

Handover cost. Inherited accounts are incomplete and expensive to reconstruct (Cemri et al., 2025); hence provenance discipline in H_0 and an equal re-read price, a channel our one environment never exercised (§3.2).

Successor selection, value estimation, agent logs. Choosing among candidate successors is a separate problem we do not address: G_0 and H_0 are fixed, so a reviewer who locates the gain in routing is not contradicted here. A state value plus per-action advantages (Baird, 1993; Wang et al., 2016), temporally extended actions (Sutton et al., 1999) and an external manager over a frozen worker (Vezhnevets et al., 2017; Packer et al., 2023) are standard; our manager (§4) is an ordinary instance whose result is withdrawn as uninformative. Logged traces and leaderboard gaps (Cemri et al., 2025; Liu et al., 2024) all share one confound: succession arrives bundled with advice and a cleaner context, and the trace never records the counterfactual. Practice does all four interventions (Guo et al., 2026; Rombaut, 2026) by heuristic, and we found no prior work measuring the four from a single restorable state.

3 The Four-Cell Design

This section describes an instrument, not a result: choosing among the four actions bought no measurable gain in either executor (§6); what follows is the machinery that makes that a fact about the interventions, not about mismatched branches. One incumbent executor is in contact with the environment at a time; the manager never acts in it, wakes at *management opportunities*, and picks one of four interventions. Weights, tools and protocol stay frozen: an intervention changes only the executor's *working context* and whether it carries a targeted suggestion. Appendix E gives the opportunity and sampling rules.

3.1 Four interventions, two factors

The actions cross *keep or rebuild the working context* against *withhold or supply a targeted suggestion* (Figure 1). **C** leaves the session untouched; **R** appends one suggestion z to it; **F** opens a new session seeded with a handover package H_0; **N** opens a new session with the *same* H_0 and the *same* z. Both artifacts come from fixed prompt-and-format generators, not trained components: a negative result about R is a result about *this* generator.

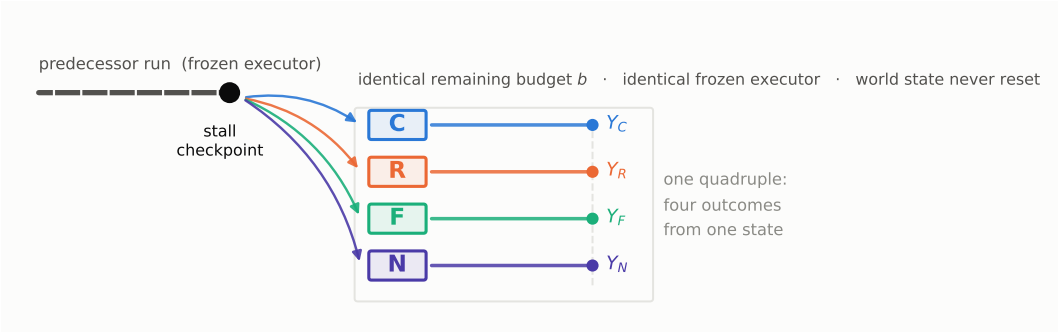


Figure 2: **How one quadruple is collected.** A predecessor run reaches a management opportunity; the evaluator snapshots that state and branches it four ways. Every branch inherits the same remaining budget b, the same frozen executor and the same un-reset world, and draws from one environment RNG stream, so Y_C, Y_R, Y_F, Y_N differ only by the intervention. Repeats re-sample z and the model stream.

3.2 What each sharing constraint excludes

Each constraint kills one rival reading of a between-arm difference (Figure 2), is enforced in code, and verified in every complete quadruple — all 1,376 of experiment 1 and all 2,302 of experiment 2 (Appendix E).

- **One z, byte-identical for R and N**; independent repeats re-sample it.
- **One H_0, byte-identical for F and N**.
- **Symmetric billing**: a branch that *uses* a shared artifact pays its generation cost in its own ledger, and no management action advances an executor turn.
- **The world is never reset**: F and N rebuild the *session*, not the episode, and keep the predecessor's protocol.
- **No free re-read**: the log stays retrievable by every arm, H_0 only points at it, and pulled-back text is billed as input tokens. *Untested here* — the Craftax executor had no log-retrieval tool, so no branch exercised the channel.
- **One environment RNG stream per quadruple**, with a distinct model stream per branch: common random numbers across arms.

In experiment 1 every handover returned at exactly the 640-token budget, so H_0 was truncation-bound throughout. This cuts against our own nulls, plausibly weakening F and N — the only arms receiving H_0 — and pushing $F - C$, $N - C$ and $N - F$ downward.

3.3 Estimands and the ceiling estimator

Let $Y(u)$ be the realized utility of running $u \in \{C, R, F, N\}$ to termination from one checkpoint. The unit is one (checkpoint, repeat) quadruple; clusters are originating maps. The pre-registered primary family is $N - C$, $N - F$ and $R - C$; $F - C$ and $N - R$ are secondary. Intervals are 95% paired cluster bootstraps over maps, 10,000 resamples, with *no* multiplicity correction — so "$R - C$ excludes zero" is an uncorrected statement about one of five contrasts. Appendix E gives the weighting and the interaction I, descriptive only.

The ceiling asks how much is available to *any* per-state policy. We first took the max over a quadruple's four arms minus C; we withdraw that number: the maximum of k noisy arms exceeds the common mean even when all four expectations are identical, and the noise here is large — within-checkpoint repeat standard deviation 1.566 achievements against between-cluster 0.450. The estimator we report instead is cross-fitted — pick the argmax on one repeat, score it on another — and is printed beside a permutation null shuffling action labels *within* a quadruple.

4 THE MANAGER, AND WHY ITS RESULT DOES NOT INFORM

Withdrawal. We previously reported that the learned manager captured 4.1% of the oracle ceiling. That number was computed in-sample, on the manager's own training quadruples; held out it is $+0.00006$ achievements. It should not be cited. The quantity it was built to capture is itself null — the cross-fitted ceiling on per-state choice is $+0.0073$ achievements ($p = 0.429$) for the weak executor and -0.0013 for the strong one (§6) — and since no deployed policy can exceed that oracle, a null ceiling bounds what *any* per-state policy could have won. The ceiling is an interval, not a point: we do not claim the available gain is zero.

Parameterisation. The manager scores C, R, F and N *before* any of them is run, from a capped record x of the state. With b the remaining budget, the value function is one trunk plus three gains measured against Continue:

$$Q_\theta(x, b, C) = v_\theta(x, b), \qquad Q_\theta(x, b, u) = v_\theta(x, b) + g_\theta(x, b, u), \quad u \in \{R, F, N\}. \tag{1}$$

The C column is v_θ with an inserted hard zero, so $Q_\theta(x, b, u) - Q_\theta(x, b, C) \equiv g_\theta(x, b, u)$ is an identity of the module, not an approximation after training (Baird, 1993; Wang et al., 2016). Appendix F gives the information set, the objective, and the deployment rule.

Deviation: the adapters were never trained. The design specifies a Qwen3-4B backbone with LoRA adapters on the attention query and value projections, adapters and value head trained and base weights frozen (Hu et al., 2022; Yang et al., 2025). LoRA training ran out of memory at 62 GB. We froze the adapters and trained only the value head: $726{,}532$ trainable parameters against a frozen 4.03B backbone, fit on $1{,}139$ checkpoint-examples. Eq. 1 is unchanged, but the learned component is a small head over fixed features, and every manager result here is a result about that smaller model. Two further gaps — the outcome-only ablation and risk-only arm were implemented but never trained, and the four-action manager has no test-split campaign row — are in Appendix F.

5 EXPERIMENTAL SETUP

Two experiments were run, both on Craftax, and both return a null. Normalized utility is *distinct achievements* / 67 over our own text interface, not comparable with Craftax RL leaderboards. Appendix G gives the interface, budgets, opportunity rule, arms and statistics.

5.1 TWO EXECUTORS

Experiment 1 used `nvidia/qwen/qwen3-32b` as executor, advice generator G_0 and summariser H_0: mean 4.0 of 67 achievements, normalized utility 0.0597. Across all 400 campaigns only 16 distinct achievements were *ever* unlocked, and campaigns end on budget exhaustion or in-game death by starvation or thirst. That is a capability floor, not a stall: an agent that has never smelted iron in 400 runs is not one turn of better advice away from it. Experiment 2 re-ran the protocol with `azure/anthropic/claude-opus-4-6` in all three roles: mean 14.4 of 67, utility 0.2145, a factor of 3.6. The two regimes bracket the range we could afford; they do not span it. Both used one base model across the three roles and the *train* split (Appendix G.8).

Registered deviation: reasoning disabled in experiment 1. The weak executor is a reasoning model, run with reasoning disabled for throughput; on identical maps this costs utility, 0.082 disabled against 0.108 enabled. Part of the capability floor is self-inflicted, and we cannot say whether the $R{-}C$ effect survives that change.

5.2 COLLECTED, AND WITHDRAWN

Experiment 1 yields 699 checkpoints from 400 campaigns, $5{,}504$ branches and $1{,}376$ complete quadruples over 354 map clusters; experiment 2, 765 checkpoints from 400 campaigns, $9{,}208$ branches and $2{,}302$ quadruples over 397 clusters, collected in two stages at six and at two repeats per checkpoint.

Three numbers from earlier internal reports are withdrawn and restated with their corrections in §6.6: a $+1.02$-achievement oracle ceiling that was max-of-k winner's curse, a learned manager said to capture 4.1% of it that was scored in-sample, and a claim that no management baseline beats doing nothing that the campaign intervals are far too wide to support either way. Finally, $R{-}C$ joined the primary family after it had resolved in experiment 1, so it is exploratory there and confirmatory in experiment 2 (Appendix G.2).

5.3 Power

We decompose the paired differences into within-checkpoint (repeat) and true between-cluster variance:

$$\mathrm{sd}_{\text{within}} = 1.566 \text{ ach}, \qquad \mathrm{sd}^{\text{true}}_{\text{between}} = 0.450 \text{ ach},$$

so 76% of the apparent between-cluster variance is repeat noise. The minimum effect detectable at 80% power is 0.129 achievements for experiment 1 (354 clusters, 2 repeats) and 0.190 for experiment 2 (397 clusters, 2–6 repeats). In experiment 1 only $R{-}C$ and $N{-}R$ clear 0.129; in experiment 2 *nothing observed clears* 0.190. So the failure of $R{-}C$ to replicate is *not* evidence of absence in the strong regime: experiment 2 establishes only that no per-state choice among the four actions produces an effect *larger* than 0.190 achievements there. The decomposition is estimated from experiment 1 and assumed to carry over (Appendix G.5).

6 Results

This is a null result. Across two executors differing $3.6\times$ in task competence, choosing among Continue, Revise, Fresh and Replace per state buys nothing we can measure: the cross-fitted per-state ceiling is $+0.007$ achievements in the weak regime ($p = 0.429$ against a within-quadruple permutation null) and -0.000 in the strong regime ($p = 0.509$). An earlier version of that quantity, $+1.02$ achievements, is withdrawn in §6.6. Two contrasts resolve, both in the weak regime and both one phenomenon (§6.1).

Every number comes from the frozen script `analysis/preregistered_fourcell.py`, hashed before any experiment 2 branch existed (§E.6). Contrasts are in achievements (utility $\times$ 67) with 95% paired cluster bootstrap intervals clustered on the originating map. Appendix H holds the cost ledger and the per-arm token accounting.

6.1 Experiment 1: weak executor

Executor `qwen3-32b`, reasoning disabled: 400 prefix campaigns, 699 checkpoints, $5{,}504$ branches, $1{,}376$ complete quadruples over 354 map clusters. The regime is a capability floor: mean 4.0 of 67 achievements, 16 distinct achievements ever unlocked. Two intervals exclude zero (Figure 3; exact values in Table 4): $R{-}C = -0.1562$ $[-0.2467, -0.0648]$ and $N{-}R = +0.1395$ $[+0.0624, +0.2150]$. They are one finding: $N - C$ is unresolved at -0.0167, so both are carried by the low R arm. $N - R$ is exploratory, outside the pre-registered primary family; $R - C$ is inside it. R and C differ in exactly one respect (§3.2).

Infrastructure contamination. 162 of the 400 prefix campaigns carry `termination_reason=infra_error`. On the clean subset (910 of $1{,}376$ quadruples, 232 of 354 clusters) $R - C$ is -0.1033 $[-0.2108, +0.0055]$ and no longer excludes zero, against a subset minimum detectable effect of about 0.16 achievements. The point estimate is stable under the removal; the contrast does not survive it (§7).

6.2 Experiment 2: strong executor

Executor `claude-opus-4-6`: 400 prefix campaigns, 765 checkpoints, $9{,}208$ branches, $2{,}302$ complete quadruples over 397 map clusters, identical pipeline. It was collected in two stages — 193 checkpoints at six repeats, then 572 at two. Capability gap: 14.4 of 67 achievements against 4.0, 0.2145 utility against 0.0597.

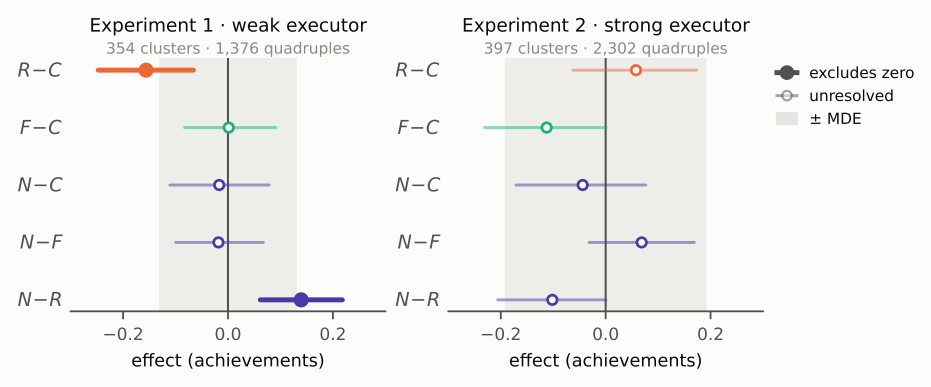


Figure 3: **All five paired contrasts, both regimes**, in achievements. Point estimate with 95% paired cluster bootstrap interval over maps; shading is ± the minimum effect detectable at 80% power. Filled markers exclude zero, hollow do not. Only $R - C$ and $N - R$ resolve, only in the weak regime, and they are one fact twice. In the strong regime every interval is comparable to or wider than the MDE band: intervals that fail to exclude zero, not measurements of zero.

Nothing resolves (Figure 3, right). All five intervals contain zero and the cross-fitted ceiling is -0.0013 $[-0.1188, +0.1213]$, permutation null mean $+0.0003$, $p = 0.509$. The frozen cross-fit uses two of the 6×5 available ordered repeat pairs, which is wasteful (Appendix H).

6.3 What this design can and cannot resolve

The within-checkpoint standard deviation of the paired differences is 1.566 achievements against a between-cluster standard deviation of 0.450: 76% of apparent between-map variation is one map disagreeing with itself across repeats, which is why the in-sample ceiling of §6.6 was so inflated. That fixes the minimum detectable effect at 80% power — the shaded bands of Figure 3 — at 0.129 achievements in experiment 1 and 0.190 in experiment 2. $R - C$ at 0.156 clears the first bar; nothing in experiment 2 clears the second. The experiment 2 nulls are therefore weak evidence about 0.1-sized effects, because that regime cannot see them; the experiment 1 nulls license only "no management action beats Continue by more than about 0.13 achievements in this regime". Null here means *unresolved*, not zero.

6.4 The advice contrast across regimes

$R - C$ is -0.156 (resolved) in the weak regime and $+0.058$ (unresolved) in the strong one. The opposite sign invites a capability story we do not tell. Experiment 2 was collected in two stages, and $R - C$ moved monotonically toward zero as clusters accumulated — $+0.123$ at 99 clusters, $+0.074$ at 288, $+0.058$ at 397 (Figure 5). A real effect holds its point estimate as n grows; an artefact shrinks toward zero. This one shrank, it is below experiment 2's MDE of 0.190, and the regimes differ in more than capability (Appendix H). We make no capability-moderation claim.

6.5 Campaign-level comparison

Table 1 runs whole policies end to end over 160 held-out test maps, one seed, hyper-parameters frozen on validation before any test map was read (`manifests/freeze_before_test_v3.json`).

Doing nothing ranks first; every arm-versus-C interval spans zero at four times the largest point estimate. $B5$ is an untrained model-free summarisation stand-in whose last place rules out nothing about learned context management (Appendix H). We draw no conclusion from the ordering: at 160 maps and one seed this comparison has no resolving power.

6.6 Withdrawn claims

Three claims in our own internal reports are wrong; we print them with their corrections.

Table 1: **Campaign-level outcome on** 160 **held-out test maps**, in utility units, weak executor, one seed; arms as in Table 3. Best to worst spans 0.0027 utility and every contrast is null. The table supports no ordering of the arms; it is printed so the absence of separation is on the record.

Arm	Policy	Utility
C_only	never intervene	0.07127
$B3$	RiskFresh, threshold 0.08	0.07118
$B2$	PeriodicFresh, $K = 16$	0.07099
$B4$	PromptManager	0.07090
$B1$	StrongSingle	0.07015
$B5$	ContextManager, fraction 0.5	0.06856

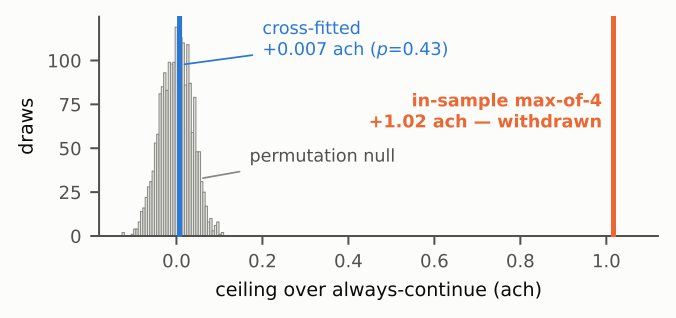


Figure 4: **Why the ceiling was withdrawn** (experiment 1). Grey: the ceiling statistic under a null shuffling action labels *within* each quadruple, preserving common random numbers. Orange: the in-sample max-of-four estimator we first reported. Blue: the cross-fitted estimator, argmax on one repeat and scored on another, sitting inside the null.

"Oracle ceiling $+1.02$ **achievements** $(+22\%)$**."** Withdrawn. That estimator took the per-checkpoint maximum over the four measured arms against C; with two noisy repeats and within-checkpoint noise of 1.566 achievements, the maximum is biased upward even when all four arms have identical expectations, and a within-quadruple permutation null reproduces essentially the whole reported ceiling (Figure 4). The cross-fitted estimate is $+0.0073$ $[-0.0630, +0.0808]$, permutation null mean -0.0003, $p = 0.429$: under one percent of the withdrawn figure.

"The learned manager captures 4.1% **of the ceiling."** Withdrawn: computed in-sample, on the manager's own training quadruples. Held out it is $+0.00006$ achievements, and the ceiling it was a percentage of is withdrawn too, so the claim fails twice.

"No management baseline beats doing nothing." Withdrawn as unsupported. The intervals behind Table 1 are too wide to support a claim in either direction, and the phrasing invited a null to be read as a finding. Nothing separates the arms.

7 Threats to Validity

Each threat below is checkable against a released artifact; Appendix I gives each in full. The three numbers we withdraw are stated with their corrections in §6.6; the estimator that replaced them was frozen before experiment 2's data existed and forbids max-of-k ceilings and in-sample policy scores by name. Appendix K says where a reviewer should push.

Infrastructure contamination in experiment 1. Of the 400 prefix campaigns behind experiment 1's checkpoints, 162 terminate with `termination_reason=infra_error` (HTTP 429 from the shared endpoint), truncating prefixes plausibly correlated with how long an agent survives. $R - C$ is -0.1562 $[-0.2467, -0.0648]$ on the full data; on the subset descending only from uncontaminated prefixes (910 of 1,376 quadruples in 232 of 354 clusters) it is -0.1033 $[-0.2108, +0.0055]$, which

does *not* exclude zero. We do not claim the contrast survives contamination removal, and experiment 2 does not resolve $R - C$ either ($+0.0578$ $[-0.0622, +0.1757]$).

The weak executor ran with its reasoning field disabled. On identical maps the same model scores 0.082 utility disabled against 0.108 enabled. All four arms share the setting, but the regime moves: a weaker agent may be more responsive to advice and to a rebuilt context. The reasoning-enabled replication was never run.

Power bounds the null, and the bound is not tight. Repeat noise is 76% of apparent between-cluster variance, and the minimum detectable effect at 80% power is 0.129 achievements in experiment 1 (354 clusters, 2 repeats) and 0.190 in experiment 2 (397 clusters, 2–6 repeats). Nothing in experiment 2 clears its bar, so its honest statement is "no effect of size 0.190 or larger".

B5 is a weak substitute. No trained AdaCoM-style context editor exists in this codebase, so B5 is an untrained, model-free summarisation rule, labelled `is_weak_substitute` in the baseline registry and the freeze manifest. It is last in the campaign table at 0.06856 against 0.07127 for doing nothing, which rules out one fixed heuristic and not learned context management as a class.

The manager's adapters were frozen. Only the value head trained: the LoRA path ran out of memory at 62 GB on the available MPS device, so the manager learns a value function over frozen features, and its campaign-table entry is a lower bound on a trained manager.

One environment. Only Craftax was collected; the FLE and SWE-bench Pro adapters are implemented and tested end to end but were never run for data. The weak-regime mechanism we do observe ($R - C = -0.1562$, $N - R = +0.1395$ $[+0.0624, +0.2150]$) could be specific to a context dominated by repetitive observation frames, and we have no cross-environment evidence.

The test split was read before the freeze, and the record said otherwise. Version 2 of the freeze manifest asserted that no test map had been read before the freeze. That was false: two smoke tests of the main-table harness ran with a 2-map argument, which resolves to the first two test seeds, and logged their utility numbers. They informed no tuned value, and the directory was deleted before the real table was collected. We superseded rather than amended the manifest, so both the false claim and its correction stand in the record (`manifests/freeze_before_test_v3.json`).

The failure mode this class of experiment has. Every defect the audit passes found was a silent wrong number, not a crash. The sharpest case is the truncated-allowance defect, under which $N - C$ resolved at -0.2478 $[-0.3423, -0.1548]$ and after the fix is -0.0167 $[-0.1114, +0.0781]$, unresolved, on the same checkpoints (Appendix J).

8 Conclusion

This paper reports a null. Choosing among replacement, advice, context rebuild and continuation from a real stalled agent state bought nothing we could measure: the cross-fitted ceiling on per-state choice is $+0.007$ achievements ($p = 0.429$) with the weak executor and -0.000 ($p = 0.509$) with the strong. One contrast resolves, in the weak regime only — advice without context repair harms, and rebuilding the context cancels that harm. We pre-registered that if advice or a clean context alone accounted for any apparent benefit, the contribution would be rewritten as context repair rather than agent succession. That clause fired.

What is *not* established is larger. Management is not shown useless: the intervals support neither direction, and effects below 0.129 achievements are outside this instrument's reach. Three of our own numbers are withdrawn, including a $+1.02$-achievement oracle ceiling that was almost entirely winner's curse (Figure 4).

The open question is resolution, and the instrument prices it: repeat noise is 76% of apparent between-cluster variance, so settling a 0.12-achievement effect needs roughly 500 clusters in the strong regime, a trained context editor, a manager whose adapters actually train, and the two environments we wired but never collected. We report the null rather than the adjustment that would have hidden it.

AI USE STATEMENT

Large language models are the object of study in this work: the executors and the fixed generators G_0 and H_0 are themselves LLMs (§5), and their outputs are the experimental data. Separately, and as a tool, we used an LLM-based coding assistant under author direction for software implementation and for drafting and editing text. The authors designed the experiments, specified and froze the analysis in advance, verified every reported number against the released records, and take responsibility for the final content of this paper.

REPRODUCIBILITY STATEMENT

Every number in this paper is produced by a frozen, hashed analysis script from released records. The pre-registered estimator is `analysis/preregistered_fourcell.py`, hashed before any experiment 2 branch was collected, and it refuses by construction the three analyses we had to withdraw (in-sample max-of-k ceilings, in-sample policy scores and post-hoc family selection); §E.6 and Appendix G.4 give the estimator, the clustering and the bootstrap. Hyperparameters were frozen on validation before any test map was read, and the freeze manifest (`manifests/freeze_before_test_v3.json`) records both that freeze and a false claim in its own superseded version 2, which §7 states as a threat. The branch-level records, checkpoint manifests and environment lock are released alongside the code, and Appendix H gives the cost ledger and per-arm token accounting needed to price a replication. Appendix J documents each defect the audit passes found, with the before-and-after estimates on the same checkpoints, so that a reader can check our corrections rather than take them on trust.

REFERENCES

Leemon C. Baird, III. Advantage updating. Technical Report WL-TR-93-1146, Wright Laboratory, Wright-Patterson Air Force Base, 1993.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-4082. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/b1041e52d3be19f0a9bc491657488e4a-Abstract-Datasets_and_Benchmarks_Track.html.

Jianyuan Guo, Zhiwei Hao, Chengcheng Wang, Cheng Fan, Tingzhang Luo, Hongguang Li, Ying Gao, Hefei Mei, Jiankun Peng, Rongjian Xu, Minjing Dong, Han Wu, Mengyu Zheng, Kai Han, Shiqi Wang, Chang Xu, and Yunhe Wang. From question answering to task completion: A survey on agent system and harness design. *arXiv preprint arXiv:2606.20683*, 2026.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35 (1):73–101, 1964. doi: 10.1214/aoms/1177703732.

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, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Michael Matthews, Michael Beukman, Benjamin Ellis, Mikayel Samvelyan, Matthew Thomas Jackson, Samuel Coward, and Jakob Nicolaus Foerster. Craftax: A lightning-fast benchmark for open-ended reinforcement learning. In *Proceedings of the 41st International Conference*

on Machine Learning (ICML), volume 235 of *Proceedings of Machine Learning Research*, pp. 35104–35137. PMLR, 2024.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

Benjamin Rombaut. Inside the scaffold: A source-code taxonomy of coding agent architectures. *arXiv preprint arXiv:2604.03515*, 2026.

Shuai Shao, Kangning Zhang, Qingyao Li, Shijian Wang, Hao Wang, Wenxiang Jiao, Yuan Lu, Yi Guo, Weiwen Liu, and Weinan Zhang. Harness-r1: Learning to edit executable runtime harnesses from agent failure trajectories. *arXiv preprint arXiv:2608.02276*, 2026.

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, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2):181–211, 1999. doi: 10.1016/S0004-3702(99)00052-1.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. FeUdal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3540–3549. PMLR, 2017.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1995–2003. PMLR, 2016.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Haiwen Yi and Xinyuan Song. Learning to control LLM agent harnesses with offline reinforcement learning. *arXiv preprint arXiv:2607.05458*, 2026.

Lu Yi, Runlin Lei, Liuyi Yao, Yuexiang Xie, Yuyang Li, Wenhao Zhang, Zhewei Wei, Yaliang Li, and Jian-Yun Nie. Learning agent-compatible context management for long-horizon tasks. *arXiv preprint arXiv:2605.30785*, 2026. URL https://arxiv.org/abs/2605.30785.

Index of numbers this paper does not have

The document contains **0** `\needsnumber` markers. Each names a quantity the surrounding claim would be stronger for, and that we did not compute. `README.md` gives, for every entry, the command that would produce it. None of them has been guessed at, and no number anywhere in this paper is an estimate standing in for a measurement.

A What Is in This Appendix, and What Used to Be

Earlier drafts of this appendix promised per-environment breakdowns for the Factorio Learning Environment, per-repository breakdowns for SWE-bench Pro, and a conditional-recall figure gated on an independent pass condition. **None of those experiments was run.** The adapters exist, are wired to the same opportunity rule, budget ledger and four-cell collector as Craftax, and pass their tests; no data was ever collected through them. The promises are deleted rather than deferred, because an appendix that lists an experiment reads, to a skimming reader, like an appendix that contains one.

Every number in this paper comes from Craftax, from two executors, from the collections described in §5. What follows is the protocol, the audit log, and the supporting material for those collections only.

B PROTOCOL, MANIFESTS AND VERSION LOCKS

The frozen protocol is reproduced from the manifests the pipeline emitted, not retyped, so that what is printed is what ran. It carries no results.

- `manifests/env_lock.json` — environment and package revisions (`craftax==1.6.1`, `Craftax-Symbolic-v1`, `auto_reset` disabled), served model revisions for both executors, and the budget tier table with every numeric field. The mid tier, used by both experiments, is recorded there together with the assertion that the realised caps matched the requested ones.
- `manifests/prereg_fourcell.json` — the pre-registration: the primary family $\{N{-}C,\ N{-}F,\ R{-}C\}$, α, the bootstrap and permutation resample counts, the RNG seed, the cross-fitted ceiling definition, and an explicit list of forbidden estimators (in-sample max-of-k ceilings, in-sample policy scores, post-hoc family selection, permutation nulls that shuffle across repeats). It carries the SHA-256 of `analysis/preregistered_fourcell.py` and the date it was written, which precedes the first experiment-2 branch.
- `manifests/freeze_before_test_v3.json` — the held-out-test freeze: $B2$'s K, $B3$'s threshold, $B5$'s fraction and the manager checkpoint, each with the validation scores it was selected on, and the record that two 2-map smoke tests had read test seeds before the freeze. This file supersedes `..._v2.json`, which asserted that no test map had been read; that assertion was false, and both files are kept so that the false claim and its correction are both in the record (§7).
- `manifests/e1_manifest.json` and the checkpoint manifests — the SHA-256 hash preselection over map seeds, performed before any method ran.

The generator and handover templates are reproduced verbatim from `succession/handoff/`, the manager input schema and its ordered compression rule from `succession/manager/input_builder.py`, and the training objective's hyperparameters from `succession/manager/train.py`. Each is printed at the revision recorded in the run manifest rather than at `HEAD`.

C THE FULL AUDIT LOG

Table 5 in Appendix J gives six of the audited defects. This appendix lists all of them, each with: what the code did, what it would have produced in the reported numbers, the fix, and the regression test that fails against the previous behaviour. They share one property, which is why they are worth a reader's attention rather than a changelog's — *not one of them raised an exception*. Each produced exactly the artifact a careful reader expects: a well-formed record, a plausible number, a run that completed.

The defect classes are: budget-tier arithmetic that cancelled the tier factor; a shared suggestion evicted from one arm's context but not the other's, biasing precisely the contrast it was shared to protect; budget exhaustion relabelled as an infrastructure fault and then entering the analysis at utility zero; a turn allowance whose token cap bound before its turn cap for the two arms that re-send a handover package; a map-aliasing check whose digest embedded the seed it was meant to distinguish, so that no input could make it fire; a percentile bootstrap that returned a zero-width interval and a floor p-value from a single cluster; a per-branch RNG that a `hasattr` guard skipped installing in silence; a budget-compliance check that `all([])` reported as satisfied with nothing to check; a cross-split overlap check that was never implemented; and an episode in which two duck-typed decision classes fell out of sync, every executor call raised, each raise was absorbed as an infrastructure error, and the test suite stayed green because it imported the class that was fine.

The first round of this log is released as `docs/AUDIT_FINDINGS.md`, with each entry naming the source location, the plan clause it violated, the measured consequence and the regression test that now locks it. The table this appendix owes a reviewer is the one that cross-references every defect against the collection it predates or postdates: the repository's audit log lists twelve numbered findings over four adversarial passes, each paired with the regression test that now pins it. Without it, a reviewer cannot tell how much of the reported data was collected under a defect that was later fixed, and neither can we. Two of the defects in Table 5 were found *after* data had been collected under them, so the question is not hypothetical.

D SUPPORTING MATERIAL

For Craftax only, and for the two collections reported:

- Termination-reason distributions per experiment, separating budget exhaustion, in-game death and infrastructure error, with the infrastructure-error rows counted and excluded before the four-cell computation.

- The sharing and charge audit output per (checkpoint, repeat) quadruple: identity of the shared suggestion across R and N, identity of the handover package across F and N, equality of generator and summariser charges, one shared environment RNG stream and four distinct model streams.
- The exclusion list with per-arm counts, so that a reader can check the exclusions are balanced across arms rather than concentrated in one.
- Per-branch residency of the shared suggestion z, measured in the actual context rather than by comparing reference strings — the check that caught the eviction defect.
- The bootstrap calibration simulation at the cluster counts actually realised (354 and 99), since a percentile interval over few clusters is the place this analysis is most likely to be over-confident.
- The campaign-level table's per-arm cost columns alongside its utility column, from `results/stageA/main_table.json`.

Two of these are more than bookkeeping and a reviewer should read them first. The exclusion balance decides whether the experiment-1 contrasts are comparable at all given that 162 of 400 prefix campaigns terminated on infrastructure error; and the bootstrap calibration decides whether the intervals we report are the intervals we think we report.

E THE FOUR-CELL DESIGN IN DETAIL

This appendix holds the material displaced from §3: the two shared artifacts, the check behind each sharing constraint together with its post-hoc audit count, the opportunity and candidate-sampling rules, and the deployment-time invocation order against which paired collection is defined. Where a constraint is enforced in code we name the check; where it is convention, or was never exercised by the environment we ran, we say so.

E.1 THE TWO SHARED ARTIFACTS

z comes from a fixed generator G_0, which reads the visible observation, recent actions and results, the plan on record and the known failures, and emits at most 512 generated tokens: one premise that may be blocking progress, evidence for and against it, at most three next operations, and the conditions under which they apply. The cap is requested of the model client and re-measured on the returned text. The generator module holds no environment and no evaluator handle, so "grants no new tools" is structural rather than a prompt-level promise, and a template containing forced-opposition wording is refused at construction, so G_0 cannot become a contrarian that manufactures reasons to intervene.

H_0 is a fixed-format package of at most $1{,}024$ tokens — goal, completed work, visible state, evidence references, open questions, recent failures — in which verified facts and tentative interpretations occupy separate fields, routed by provenance: a statement renders as a fact only if a tool return supports it *and* that return is citable, otherwise it is marked unverified.

G_0 and H_0 are fixed prompt-and-format artifacts, not trained components. That is a real limit on what any arm here can show: a negative result about R is a result about *this* G_0.

E.2 WHAT EACH SHARING CONSTRAINT EXCLUDES, AND ITS AUDIT COUNT

Each constraint removes one rival explanation for a difference between arms. We report the post-hoc verification counts because the constraints are only worth as much as their audit.

One z, byte-identical for R and N, or "N did better" could just mean the successor got better advice. The artifact is generated once per (campaign, checkpoint, repeat) and handed to both branches as the same object, and `verify_paired_branches` asserts that N's guide reference is R's and that both were billed the same generator cost. In the collected data this holds in all $1{,}376$ complete quadruples of experiment 1 and all $2{,}302$ of experiment 2. Independent repeats re-sample G_0, so one replayed suggestion is not mistaken for a property of the intervention.

One H_0 for F and N, or $Y_N - Y_F$ confounds replacement with a better-written handover. Same construction, same check, same counts. One caveat that cuts against our own nulls: the summarizer is told to compress into a 640-token state budget, and in experiment 1 every package came back at exactly that budget, so the handover was truncation-bound in every case. F and N are the only arms that receive H_0. A handover that always had more to say plausibly weakens exactly those two arms, which would push $F - C$, $N - C$ and $N - F$ downward. We did not run a larger-H_0 arm.

Symmetric billing. Every branch that *uses* a shared artifact pays the same generation cost in its own ledger; C pays for neither and F pays no generator fee for a z it never receives. Verified in every quadruple of both

experiments. No management action advances an executor turn — manager, generator and summarizer calls spend model budget only — and this is asserted over every four-cell group.

The world is never reset. F and N rebuild the *session*, not the episode. The module realizing the four actions has no environment handle at all: it passes the task site through by identity and re-checks it on exit, and the worker fingerprints the live world at session start and raises if the adapter's world changes identity without an announced reset. A silently restarted episode would otherwise look like a spectacular handover benefit. **F and N keep the predecessor's protocol**; a rebuilt session returning a different one raises, so a context effect is not confounded with a change of working habit.

No free re-read: the raw log stays retrievable by every arm, H_0 only points at it, and text pulled back into a prompt is billed as input tokens on that branch's ledger — without this, "the successor simply had more evidence" would be an alternative reading of any positive $N - C$. *This one is untested in the data we report.* The Craftax executor was given no log-retrieval tool, so no branch ever exercised the channel and every branch ledger shows zero retrieval cost. The constraint is a property of the harness, not something our numbers confirm.

One environment RNG stream per quadruple. With `env_rng_mode="shared"`, the four branches of one (checkpoint, repeat) are started on the same derived external random stream, and each branch gets its own model stream. Verified: identical environment stream and four distinct model streams in all $1{,}376$ and all $2{,}302$ quadruples. This is common random numbers across arms, and the ceiling estimator of §E.6 depends on it.

Snapshots of the world exist but are held by the evaluator alone, so no method treats the world as free trial and error.

E.3 When a decision may be made, and which positions are sampled

Opportunities are a pure function of the executor's past turns: the first at the end of turn 16, then every 16; three consecutive same-type tool errors may bring one forward; two are never closer than four turns. None is emitted on a turn that reached a terminal state, and a state already considered solved is not turned into a handover to fill a quota. The periodic grid is absolute, so an early trigger does not re-phase it. The rule's input type carries no outcome, no hidden state and no sibling-branch result — there is nowhere to put future information — which is what makes the same set legal at deployment time, and every method with an external manager reads that same set: the periodic-restart baseline with $K \in \{16, 32, 64\}$ filters these ticks instead of running its own schedule, so a difference between managers is never a difference in when they were allowed to speak.

Candidates are opportunities with at least 16 executed turns that are not terminal. At most two are taken per predecessor run, without replacement: one drawn uniformly over candidates, one from the subset flagged as observably stalled, falling back to a second uniform draw when no stalled candidate remains. The stall flag is crude and fully observable — a repeated native action batch, repeated tool errors, or no gain on an observable progress scalar — over the same eight recent turns the manager sees anyway. Each pick records its draw probability, the exact marginal inclusion probability under the full scheme, the scheme name and the derived RNG reference, so the natural distribution and the stalled subset can be reweighted and reported separately. The outcome cannot reach the sampler: its signature has no parameter for a utility, a resolved/failed label or an evaluator snapshot, and its one free-form metadata channel is screened against outcome-like key names.

The period of 16 turns, the minimum gap of four, and the three-error trigger were set before collection and never tuned. We have no evidence that they are the right resolution. A reviewer who thinks intervention value lives at a finer or coarser granularity is objecting to something we did not test.

E.4 Deployment order, and how collection differs from it

At deployment the manager chooses first, from the visible record alone. G_0 is invoked only after R or N is chosen and H_0 built only after F or N is chosen; candidate inputs are passed as thunks and resolved lazily, so an unused candidate is never assembled, let alone paid for, and what was invoked is recorded per branch — for C, only the choice itself. The claim this licenses is correspondingly narrow: what is learned is the expected handover value of *this fixed* G_0, not a routing policy over arbitrary candidates or models.

Paired collection does the opposite: all four branches run from the same restored snapshot, and both shared artifacts are built once and charged into every using branch's ledger. The remaining interaction allowance is drawn once per repeat — uniformly from $\{32, 64, 128\}$ turns at the middle tier, rescaled to each tier's own units — with token and environment budgets scaled proportionally, so the four arms of a repeat differ only in the intervention, and the drawn allowance is stated to the executor rather than left implicit. Verified: the four arms of every quadruple in both experiments carry the same recorded allowance. A handover never refreshes an allowance; every branch inherits the prefix cost already incurred. Budget exhaustion is a legal termination and yields a usable label; only infrastructure failures are excluded, and a failed G_0 or H_0 call fails the whole repeat together rather than being retried for one arm, since a retry would hand N a different suggestion from R. Restoring a snapshot is charged on a separate diagnostic line, never as a deployment-time capability.

E.5 Estimands and ceiling estimator, in full

Let $Y(u)$ be the realized utility of running $u \in \{C, R, F, N\}$ to termination from one checkpoint under the stated remaining budget. The unit is one (checkpoint, repeat) quadruple; clusters are originating maps. The pre-registered primary family is $N - C$ (does replacement buy anything at all), $N - F$ (anything beyond the same clean context) and $R - C$ (does advice alone do anything); $F - C$ and $N - R$ are secondary. Intervals are 95% percentile intervals from a paired cluster bootstrap over maps with 10,000 resamples, each per-contrast: there is *no* multiplicity correction, so "$R - C$ excludes zero" is an uncorrected statement about one of five contrasts. Repeats are not averaged within a checkpoint first, and every contributing checkpoint contributed the same number — two in experiment 1, six in experiment 2 — so no checkpoint is reweighted.

We also report the interaction

$$I \;=\; \big(Y_N - Y_F\big) - \big(Y_R - Y_C\big), \tag{2}$$

descriptive at this resolution and nothing more: not pre-registered, and a difference of two differences, so noisier than any component. It is $+0.1381$ $[+0.0300, +0.2486]$ in experiment 1 and -0.0838 $[-0.2856, +0.1233]$ in experiment 2; we draw no inference from it. A commitment fixed before the data were read: if $R - C$ or $F - C$ carries an effect, the contribution is context repair, not succession. That is the case that applies — in the weak regime $R - C = -0.1562$ $[-0.2467, -0.0648]$ and $N - R = +0.1395$ $[+0.0624, +0.2150]$.

E.6 The ceiling estimator, and the one we withdrew

We wanted an upper bound: how much is available to *any* per-state policy over $\{C, R, F, N\}$? We first took the max over a quadruple's four arms, minus C. That statistic is wrong and we withdraw the number it produced. Under noise the maximum of k arms exceeds the common mean even when all four expectations are identical, and the bias grows with per-arm noise, which here is large: within-checkpoint repeat standard deviation 1.566 achievements against between-cluster 0.450, so 76% of apparent between-cluster variance is repeat noise. That version returned an "oracle ceiling" of $+1.02$ achievements, or $+22\%$, reported internally and almost entirely bias. It must not be used, and we say so here rather than in an appendix because we published it to ourselves and believed it.

The pre-registered statistic instead selects and scores on different repeats: per checkpoint with at least two complete repeats, take the argmax arm on the first and score *that arm minus C* on the second, average over both orderings and over checkpoints, and bootstrap over map clusters. Selection and scoring noise are then independent, so the statistic is centred at zero under the null. Beside it we report a permutation null that shuffles the four action labels *within* each quadruple (2,000 draws). Within-quadruple is the correct exchangeability, since those four utilities share one environment stream; shuffling *across* repeats or checkpoints breaks the common random numbers and inflates the null for any max-like statistic. An internal audit failed in exactly that way. The frozen outputs are $+0.0073$ $[-0.0630, +0.0808]$, permutation null mean -0.0003, $p = 0.429$ (weak executor), and -0.0013 $[-0.1188, +0.1213]$, null mean $+0.0040$, $p = 0.509$ (strong executor); those null means are the empirical check that the statistic is centred. It is not literally a ceiling but the realized value of one feasible selection rule, argmax on an independent repeat; we keep the word for continuity with the number we withdrew.

What this estimator cannot do. It bounds per-state choice only among *these four* actions at *these* opportunities, and uses two repeats per checkpoint, so in experiment 2 four of six are discarded — part of why that interval spans $[-0.1188, +0.1213]$. The unregistered all-pairs variant gives $+0.0073$ $[-0.0618, +0.0810]$ and $+0.0648$ $[-0.0452, +0.1793]$, neither of which resolves; we flag it for a reviewer to demand rather than substituting it. Above all, resolution binds every null here: at 80% power the minimum detectable effect is 0.129 achievements with 354 clusters and two repeats, and 0.190 with 397 clusters. Only $R - C$, at 0.156, clears its threshold; nothing in experiment 2 clears 0.190, and an effect of 0.12 would need roughly 500 clusters. Our nulls mean "smaller than these thresholds", not "zero" — with the caveat that both cross-fitted point estimates sit at zero, which is not what a merely-underpowered estimator looks like. Two further limits: the opportunity grid (first decision at the end of turn 16, then every 16) was assumed, not calibrated, so a reviewer who locates intervention value at a finer granularity objects to something we did not test; and the instrument ran in one environment, Craftax alone, on 400 prefix campaigns with the weak executor and 100 with the strong. The other two adapters are wired and tested but never collected, so "two capability regimes" means two executors in one world, not two worlds.

F Manager Implementation

This appendix holds the manager specification displaced from Section 4: the information set, the objective, the deployment rule, the two remaining deviations between the specified design and what ran, the limits of the null-ceiling argument, and the questions we would most like a reviewer to push on. Nothing here is evidence for the method; the manager's only empirical claim is withdrawn in Section 4.

Rendered section	Content
Goal	task description at this checkpoint
Current legal observation	the environment's official observable view
Recent turns	at most 8 turns, oldest first
Current plan	the agent's stated plan and its known failures
Work and evidence	completed work, references for re-reading logs
Error statistics	error turns, total turns, distinct action signatures
Remaining budget	the branch's actual remaining allowance
Candidate actions	the mechanical contract of C/R/F/N, no ordering or prior

Table 2: The manager-visible record x. The list is closed: a field outside it is never rendered, and a field naming evaluator-only material is refused rather than dropped.

F.1 What the manager is

At a management opportunity the manager must choose among Continue (C), advise (R), fresh restart (F) and new-executor-with-handover (N) *before* any of them is run, from what an operator could see at that moment. The estimator is direct conditional value management; subtracting the Continue return is ordinary advantage comparison, not a new reinforcement-learning principle (Baird, 1993; Wang et al., 2016). No bootstrapping, target network, policy gradient or switching bonus appears anywhere in it.

Information set. The manager reads a record x with a closed content list in a fixed order (Table 2), capped at $4{,}096$ tokens and compressed, when over cap, by a fixed ordered rule that calls no model. Four classes are excluded and the builder raises on them rather than filtering silently: the evaluator-only snapshot reference and any field marked hidden or future; the final outcome, termination reason and cost of the episode being judged; the other three arms' outcomes at this checkpoint; and the predecessor's model identity or protocol label, which would act as a bare success prior. Campaign, task and turn identifiers travel outside the text as grouping keys only.

Budget features. In Eq. 1 the gain layer is initialised to exactly zero, so an untrained manager predicts that switching buys nothing. The budget enters as its own feature block: normalised remaining caps, a logarithmic turn feature, indicators at the sampled continuation allowances, and the minimum fraction over the caps that exist, since a run stops as soon as any single cap does. A cap a domain lacks is flagged, never imputed as full. Holding the C column at a hard zero keeps the model readable in the units of the experiment and makes the paired loss term below a direct supervision of the gains.

Objective. Each checkpoint contributes two targets, both fit with a Huber loss (Huber, 1964) whose transition point is a tenth of the utility range. The outcome term regresses $Q_\theta(x, b, u)$ on $\overline{Y}(u)$, the *mean* terminal utility over the repeated continuations of u — their mean, not the individual rollouts and not the best of them. The paired term regresses $Q_\theta(x, b, u) - Q_\theta(x, b, C)$ on $\overline{Y}(u) - \overline{Y}(C)$ over the repeat indices u and C share. The two are equally weighted. Every reduction is a mean *within* the example before the batch reduction, so a checkpoint branched at four actions and one branched at two each contribute a single number, and a checkpoint with three repeats does not outvote one with two. A checkpoint whose continuations ran at several allowances yields one example per allowance, each weighted by the reciprocal of that count. Unbranched actions are masked, never imputed.

Deployment. The manager renders x, scores the four candidates in one forward pass, and takes the highest predicted value among the available actions. An exact tie that includes C keeps C; "exact" means bit-equal, since a tolerance would become an undeclared switching threshold. There is no switch bonus, penalty, margin, hysteresis or exploration parameter. Two charges are posted against the executor's own budget: the compression that produced x, and one manager call that scores a fixed action set without decoding. The decision precedes the candidate — a suggestion is generated only after R or N is chosen, a handover package built only after F or N — so the manager estimates the value of handing over to the *fixed* generator and handover format, not content-aware routing over candidates it has inspected.

F.2 The two remaining deviations

Beyond the frozen-adapter deviation reported in Section 4, two further gaps between the specified design and what ran, both verified against the run directory and the pre-test freeze record:

- The outcome-only ablation (paired term switched off at matched capacity) and the separately trained risk-only arm are implemented and tested, but neither was trained. Only the paired objective was fit.

The B3 RiskFresh row in the campaign table therefore reads its risk signal off the *same* paired-objective head's $v_\theta(x, b)$ output, with the Fresh threshold selected on validation. It is not the independently supervised control the design describes.

- The four-action manager has no campaign row on the test split at all. It was run as full campaigns on validation, where the checkpoint was selected, but the held-out table (Section 6) contains only C-only and B1–B5. The only trained head that reaches the test maps is the v_θ read-out inside B3, at 0.07118 against 0.07127 for doing nothing, a difference far inside the interval.

F.3 What the null ceiling does and does not imply

The argument for calling this uninformative is narrow and we state its limits. The cross-fitted ceiling estimates the value of an oracle that picks the best of the four actions per state, using one repeat to choose and the other to score. It is null in both executor regimes. A deployed policy cannot exceed that oracle. So an at-best-null manager is the expected outcome under the data we have, and observing it tells us nothing about whether a better-trained manager, a trained encoder, or a different action set would do better.

What the null ceiling does *not* establish is that the available gain is zero. It is an interval, not a point. In the weak regime the ceiling interval admits gains up to $+0.0808$ achievements; in the strong regime, up to $+0.1213$. The paired-difference variance decomposition (Section 6) puts the true between-cluster standard deviation at 0.450 achievements against a within-checkpoint repeat standard deviation of 1.566, so 76% of the apparent between-cluster variance is repeat noise, and the minimum detectable effect at 80% power is 0.129 achievements with 354 clusters and two repeats, 0.190 with 397 clusters. A per-state gain below those thresholds would be real and invisible to us. The honest summary is: if a per-state gain exists, it is smaller than this design can see, and the manager's failure to find it is therefore not diagnostic.

F.4 Where we are unsure, and where a reviewer should push

- **Is the frozen-adapter head a fair instantiation?** We do not know how much of the manager's flatness is the frozen encoder. A last-token pooled representation of a $4{,}096$-token record, fed to a small head, may simply not separate the states that matter. We have no ablation isolating this, because the LoRA arm never trained.
- **Is selection-on-validation-campaigns the right criterion?** The checkpoint was chosen by full-campaign validation utility with cost as tie-break, not by offline value error. With a null ceiling, that criterion is selecting among candidates whose true differences may all be zero, which makes the selected checkpoint close to arbitrary. We do not have a principled alternative to offer.
- **Does the four-action menu exhaust the useful interventions?** The ceiling is null *over these four actions with this fixed suggestion generator and this fixed handover format*. The manager chooses among them; it cannot compose or modify them. A reviewer who believes the gain lives in the content of the intervention rather than in the choice among fixed interventions is, on this evidence, not contradicted.
- **Is a per-state policy the right target at all?** The one contrast that resolves — advice without context repair harms in the weak regime — is a main effect, not a state-dependent one. A fixed rule exploiting a main effect needs no manager. We did not pre-register, and have not run, a comparison between the learned per-state policy and the best fixed rule.
- **Would more data change this?** Resolving an effect of 0.12 achievements would need roughly 500 clusters. We expect that collection to narrow the ceiling interval around zero rather than move it, since the point estimate is zero in both regimes on independent data. That is a prediction, not a result; we did not run it.

We report the manager because it was pre-specified and executed, and because suppressing it after the ceiling came back null would be selective reporting of exactly the kind this paper criticises elsewhere. We do not report it as evidence.

G Experimental Setup in Detail

This appendix holds the parts of the setup displaced from §5 for length: the budget protocol, the opportunity and checkpoint rule, the campaign-level arms and their hyper-parameter selection, the statistics machinery, the freeze record, and the adapters that were implemented but never collected on. Nothing here is a result; every number quoted is the one that was already recorded in the manifests.

G.1 BUDGET PROTOCOL

A *turn* is one legal interaction the executor sends to the environment, queries included. Invalid actions, parse repairs and permitted retries are billed identically. Manager, generator and summariser calls spend model budget but never advance a turn. Any single exhausted cap ends the run. Both experiments ran the same mid-tier caps: 128 actor turns, 1024 native environment operations, 500,000 input and 320,000 output tokens per campaign, verified identical in both run manifests.

A handover never refreshes the allowance. Each branch ledger is forked from the consumed prefix, so the four arms of one checkpoint begin from the same spent budget. An artifact generated once and used by two arms is charged in full to each ledger that uses it, and arms that do not use it pay nothing: C pays for neither z nor H_0, F pays no generator fee. The remaining interaction allowance is drawn once per repeat, uniformly from $\{32, 64, 128\}$ turns, with token and environment budgets scaled proportionally, and is stated to the executor rather than left implicit. Budget exhaustion is a legitimate terminal state and enters the analysis as an outcome.

G.2 OPPORTUNITIES, CHECKPOINTS, AND WHAT WAS FROZEN WHEN

Management opportunities are a pure function of the executor's past turns: the first at the end of turn 16, then every 16 turns; three consecutive same-type tool errors may bring one forward; two are never closer than four turns; none is emitted on a turn that reached a terminal state. The rule's signature has no parameter for an outcome, a snapshot or a sibling branch, so the same set is legal at deployment time.

At most two checkpoints are taken per campaign, without replacement: one drawn uniformly over legal opportunities, one from the subset flagged as observably stalled, falling back to a second uniform draw when no stalled candidate remains. Each pick records its draw probability and marginal inclusion probability, so the natural distribution and the stalled subset can be reweighted separately. The stall flag is crude and fully observable: a repeated action batch, repeated tool errors, or no gain on an observable progress scalar over the last eight turns.

The campaign-level table. Separately from the four-cell diagnostic, six full-campaign arms were run on the 160 held-out test maps with one model seed: `C_only` 0.07127, B3 0.07118, B2 0.07099, B4 0.07090, B1 0.07015, B5 0.06856 in utility units. The spread between best and worst is smaller than any of the intervals around it. B2's $K = 16$, B3's threshold 0.08, B5's fraction 0.5 and the manager checkpoint were all selected on the validation split before any test map entered the table. The learned manager is scored at the quadruple level rather than as a campaign arm in this table; that asymmetry is a gap in the evidence, not a choice we can defend.

One defect in the freeze record. Before the final freeze, two 2-map smoke tests of the campaign harness ran against test seeds. They printed utility numbers to a log, they informed no tuned value, and they were discarded. An earlier freeze record asserted that no test map had been read, which was false; it is superseded rather than amended, and the correction is recorded in `manifests/freeze_before_test_v3.json`. We report this because a freeze record that states something untrue is worse than one that admits an exception.

G.3 THE CAMPAIGN-LEVEL ARMS

Table 3 defines the arms that appear in the campaign-level comparison. Each is a whole policy run end to end over a campaign: it decides for itself when to intervene, from the same management-opportunity set (§G.2), so that a difference between arms is never a difference in when they were allowed to speak. Every model call an arm makes is billed to that arm's own ledger, including $B1$'s self-reflection calls and $B4$'s manager calls.

$B2$'s $K = 16$ was chosen on validation from $\{16, 32, 64\}$, $B3$'s threshold 0.08 and $B5$'s rebuild fraction 0.5 likewise, each with its validation scores recorded in `manifests/freeze_before_test_v3.json` before any test map entered the table. Two properties of this set are worth a reviewer's attention before reading Table 1. First, the arms are not equally strong instantiations of the ideas they stand for: $B1$, $B2$, $B3$ and $B4$ are faithful to what the design asked for, and $B5$ is not. Second, the comparison that would have mattered most — the trained four-action manager as its own campaign arm — is absent, so the table cannot be read as a comparison between learned and heuristic management.

G.4 STATISTICS

Repeats are averaged within a checkpoint before any contrast. The resampling unit is the map cluster, never the branch: several checkpoints from one campaign are not independent tasks. Effects carry 95% paired cluster bootstrap intervals from 10,000 resamples with a recorded seed, effect size leading. Holm's correction is applied over a pre-specified family of three (Holm, 1979); the implementation refuses any label outside it. Every other contrast is labelled exploratory.

Table 3: The campaign-level arms, and the explanation each was built to remove. $B5$ is marked as a weak substitute in the baseline registry and in the freeze manifest: no trained context editor exists in this checkout, so what ran is an untrained, model-free summarisation rule, and no result against it excludes learned context management. The trained four-action manager has *no row of its own* on the test split (§4); it enters Table 1 only through the value trunk $B3$ reads.

Arm	Policy	Rival explanation it is meant to remove
C_only	never intervene	— (the reference arm)
$B1$	StrongSingle	that the benefit is only the extra self-correction the method hands itself: one executor, same logs and evidence, may request its own context rebuild
$B2$	PeriodicFresh	that restarting on a schedule is enough and no state-dependent judgement is needed: emit F every K executor turns
$B3$	RiskFresh	that it is enough to recognise states about to fail: emit F when a predicted failure risk crosses a threshold
$B4$	PromptManager	that learning added nothing over the backbone's own management common sense: the same frozen model reads the same rendered record and is prompted to pick one of C/R/F/N
$B5$	ContextManager **(weak substitute)**	that the benefit is entirely context tidying: rewrite, delete and merge the working context, same evidence and suggestion access, executor frozen

The ceiling estimator, and what it replaced. The withdrawn $+1.02$ achievements ($+22\%$) weak-regime ceiling was max-of-k winner's curse. The estimator now used is cross-fitted: the best action is chosen on one repeat of a checkpoint and scored on the *other* repeat, and the result is printed beside a permutation null that shuffles action labels *within* a quadruple only, preserving the shared environment RNG. Under that estimator the ceiling is $+0.0073$ achievements with permutation $p = 0.429$ in experiment 1 and -0.0000 with $p = 0.509$ in experiment 2. The withdrawn manager score, 4.1% of the ceiling, was in-sample on its own training quadruples and is $+0.00006$ held out.

The primary family changed between the two experiments. The pre-registration frozen on 2026-09-23 fixes the family as $\{N{-}C,\ N{-}F,\ R{-}C\}$. An earlier configuration, written before any data existed, named $\{N{-}R,\ N{-}F,\ F{-}C\}$. Experiment 1 was collected on 2026-09-18, between the two. So $R{-}C$ — one of the two contrasts that resolve — entered the primary family *after* it had been seen to resolve in experiment 1. For experiment 1 it must be read as exploratory. For experiment 2 the family was genuinely frozen before any branch was collected, and there $R{-}C$ does not replicate: the point estimate flips sign and the interval covers zero. This is the single place in the paper where we think a reviewer is most likely to find us guilty of something, and we would rather point at it than have it found.

A known inefficiency in the frozen estimator. The cross-fit uses the first two repeats of each checkpoint. In experiment 2 that discards four of six repeats, which costs power. A post-hoc variant using all ordered repeat pairs was computed and gives a different point estimate; it is not reported, because it was computed after the pre-registered one and choosing between them afterwards is exactly the failure mode the pre-registration exists to prevent. If reviewers judge the all-pairs version to be the better estimator on its merits, it should be re-frozen and re-run, not adopted retroactively: an all-pairs variant, which uses all $5{,}790$ ordered repeat pairs instead of the two the pre-registration fixed, gives $+0.0648$ $[-0.0452, +0.1793]$ — also unresolved, and reported here only because it was computed.

G.5 Caveats on the variance decomposition

The decomposition reported in §5.3 is estimated from the $N{-}C$ paired differences of experiment 1 and then assumed to carry to experiment 2 and to the other contrasts, and the power figures are a normal approximation rather than a bootstrap. Neither assumption is checked here, and both are cheap for a reviewer to attack.

Two further consequences of the decomposition. First, repeats buy more power per unit of compute than maps do, which is why experiment 2 was run at 6 repeats on 99 maps rather than at 2 repeats on proportionally more maps. Second, any analysis that treats one repeat per checkpoint as a per-map measurement will badly overstate its own precision. Resolving an effect near 0.12 achievements would require on the order of 500 clusters, which we did not run.

G.6 IMPLEMENTED BUT NOT COLLECTED

FLE Lab-Play and Open-Play, SWE-bench Pro and ARC-AGI-3 adapters are implemented, wired to the same opportunity rule, budget ledger and four-cell collector, and pass their tests. No data was collected on any of them. They appear in this paper only as evidence that the design is not Craftax-specific, and no claim in this paper rests on them. Craftax-Classic was used as an interface and recovery smoke test and enters no reported statistic.

G.7 WHERE WE MOST WANT TO BE PUSHED

Three places, in the order we rate them. (i) Whether the setup can distinguish a capability floor from a stall at all: in experiment 1 the executor never smelted iron in 400 runs, and if most sampled checkpoints are floor states rather than stalls, then the null is about our checkpoint distribution and not about management. We record a stall flag and the inclusion probabilities, so the analysis can be reweighted onto the stalled subset, but we have not shown that the stalled subset contains recoverable states. (ii) Whether G_0 and H_0, being fixed prompt artifacts drawn from the same model as the executor, are strong enough for the null to mean anything about advice in general. (iii) Whether the cross-fitted ceiling is the right ceiling. It measures what an oracle over the four actions would gain *given this action set at these opportunities*; it is silent about an intervention we did not implement.

G.8 DETAILS DISPLACED FROM THE SETUP SECTION

Interface. The executor reads a frozen text rendering of the 8268-dimensional observation and emits up to 8 of the 43 native actions per turn. The renderer takes the observation vector and never the environment state, so the full map, off-screen mobs and the achievement vector are unreachable from the agent-visible path. Normalized utility (*distinct achievements* / 67, the achievement count of this pinned version) is reported next to raw achievement counts and never as an official score.

What the weak executor never did. Across experiment 1's 400 campaigns, 51 of the 67 achievements were never unlocked once. The unlocked set stops at the stone tier: wood, table, wooden and stone tools, coal, a first iron ore. No campaign placed a furnace or smelted anything. The strong executor of experiment 2 smelts, makes iron tools and torches, reaches the deeper levels and casts spells.

Reasoning disabled in experiment 1. The disabled configuration is $17\times$ cheaper in output tokens, which is why it was chosen. We did not re-run experiment 1 with reasoning enabled, so we cannot say whether the floor, or the $R{-}C$ effect that sits on it, survives that change. Experiment 2 had no such setting applied.

Train-split collection. The four-cell contrasts involve no fitted model, so they are not in-sample in the usual sense; but the cross-fitted ceiling and the manager's own score are computed over overlapping checkpoints, and we treat the manager's number as held-out only when it is scored on quadruples it did not train on. A reviewer may reasonably read the two experiments as varying one factor (model) that we have described as two (capability regime); we cannot separate them with this data.

Infrastructure loss in experiment 1. Of the 162 prefix campaigns carrying `termination_reason =infra_error`, 34 ran zero turns and the median among them is 81.5 turns. The clean subset's minimum detectable effect is about 0.16 achievements, so it is underpowered for excluding zero. We have not established that the loss is independent of map difficulty, and a reviewer should not assume it is.

Freeze-record correction. The superseding record is `manifests/freeze_before_test_v3.json`. We report this because a freeze record that states something untrue is worse than one that admits an exception.

G.9 FURTHER DETAILS DISPLACED FROM THE SETUP SECTION

Site. We use `Craftax-Symbolic-v1` at `craftax==1.6.1` with `auto_reset` disabled (Matthews et al., 2024), because a stalled state resumes bit-for-bit there: that is what makes the four arms a comparison of interventions from one state. Craftax's published numbers come from RL agents trained on raw observations, so ours are not comparable with those leaderboards and we never place the two side by side.

Nested map seeds. Experiment 2 used the first 100 map seeds of experiment 1's 400: the maps are nested, not independent draws. Both collections were taken on the *train* split, so the cross-fitted ceiling and the manager's score share checkpoints with its training data.

Exploratory status of $R{-}C$. $R{-}C$ entered the frozen primary family $\{N{-}C,\ N{-}F,\ R{-}C\}$ *after* it had been seen to resolve in experiment 1, so in that experiment it is exploratory; for experiment 2 the family was frozen before collection.

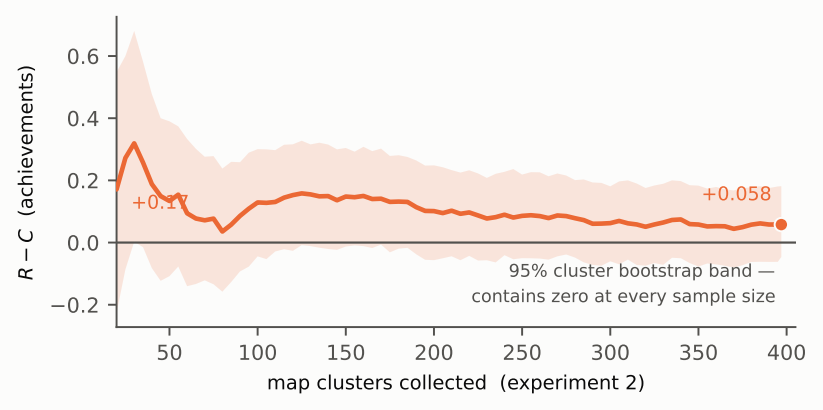


Figure 5: **The advice contrast in the strong regime shrank as evidence accumulated.** $R - C$ recomputed over the first k map clusters in collection order, with a 95% cluster bootstrap band. The point estimate decays from roughly $+0.2$ early to $+0.058$ at the full 397 clusters, and the band contains zero at every sample size. This is the signature of sampling noise being resolved, not of an effect being measured — which is why we decline to read the sign difference between regimes as capability moderation.

Contrasts below the detection floor. In experiment 1, $R-C$ at -0.1562 and $N-R$ at $+0.1395$ both clear the 0.129 minimum detectable effect and both exclude zero, with $N-R$ exploratory under Holm; $F-C$, $N-C$ and $N-F$ fall well below it, with intervals consistent with threshold-size effects in either direction. The power calculation uses a normal approximation throughout.

Clean-subset power. The infrastructure-loss clean subset (910 quadruples, 232 clusters) has a minimum detectable effect of about 0.16 achievements, so its failure to exclude zero is itself underpowered.

H Additional Results

This appendix holds the material displaced from §6 by the page limit. Nothing here is new: every number is the one printed in the main text or produced by the same frozen analysis script `analysis/preregistered_fourcell.py`, hashed in `manifests/prereg_fourcell.json` before any branch of experiment 2 existed. That file fixes the estimator, the primary family ($N - C$, $N - F$, $R - C$), the resample count and the seed. Contrasts are reported in achievements, i.e. utility $\times$ 67; intervals are 95% paired cluster bootstraps over 10,000 resamples, clustering on the originating map. We report effects and intervals rather than p-values for the contrasts; the only p-values are the permutation nulls on the ceilings.

H.1 Experiment 1: the regime, and what $R - C$ means in it

The executor is `qwen3-32b` with its reasoning field disabled. 400 prefix campaigns produced 699 checkpoints; branching four ways with two repeats gave 5,504 branches, of which 1,376 form complete quadruples, over 354 map clusters.

The regime is a capability floor. Agents unlock a mean of 4.0 of the 67 Craftax achievements. Only 16 distinct achievements are *ever* unlocked across all 400 runs; 51 are never unlocked once. Runs end on a mixture of budget exhaustion and in-game death by starvation or thirst. Whatever we measure here is measured on an agent operating near the bottom of the task's range, and that matters for reading $R - C$.

R and C differ in exactly one respect. Both keep the same executor session and the same accumulated context; R additionally receives one suggestion message from the frozen generator G_0, byte-identical to the one N receives (§3.2). Delivering that message into a context the agent has already filled with its own failed attempts is worse than delivering nothing. We report this as a measured effect of one manipulation under one executor, not as a theory of advice.

Two intervals exclude zero, marked in Table 4: $R-C = -0.1562$ $[-0.2467, -0.0648]$ and $N-R = +0.1395$ $[+0.0624, +0.2150]$. These are not two independent findings. By construction $N - R = (N - C) - (R - C)$, and since $N - C$ is itself unresolved at -0.0167, both resolved intervals are carried by the same fact: the R arm

Table 4: **Paired contrasts in both capability regimes**, in achievements (utility $\times$ 67), 95% paired cluster bootstrap intervals over 10,000 resamples clustered on the map. † marks an interval excluding zero; both are carried by the same low R arm, since $N - R = (N - C) - (R - C)$. "< MDE" means the magnitude falls below what this design detects at 80% power (§6.3): *unresolved*, not zero.

	E1: weak, 354 clusters, 2 rep.			E2: strong, 397 clusters, 2–6 rep.		
Contrast	effect	95% CI	vs MDE 0.129	effect	95% CI	vs MDE 0.190
$R - C$	-0.1562	$[-0.2467, -0.0648]$	†	$+0.0578$	$[-0.0622, +0.1757]$	< MDE
$F - C$	$+0.0015$	$[-0.0831, +0.0886]$	< MDE	-0.1125	$[-0.2302, +0.0035]$	< MDE
$N - C$	-0.0167	$[-0.1114, +0.0781]$	< MDE	-0.0439	$[-0.1687, +0.0793]$	< MDE
$N - F$	-0.0182	$[-0.1026, +0.0670]$	< MDE	$+0.0686$	$[-0.0331, +0.1690]$	< MDE
$N - R$	$+0.1395$	$[+0.0624, +0.2150]$	†	-0.1017	$[-0.2034, +0.0017]$	< MDE
cross-fitted ceiling	$+0.0073$	$[-0.0630, +0.0808]$	$p = 0.429$	-0.0000	$[-0.1188, +0.1213]$	$p = 0.509$
permutation null mean	-0.0003			$+0.0040$		

scores low. The honest statement is that one thing resolved, not two. $N - R$ is also outside the pre-registered primary family and is therefore exploratory; $R - C$ is inside it.

Infrastructure contamination, in full. 162 of the 400 prefix campaigns carry `termination_reason=infra_error`; 34 of those ran zero turns and the median campaign ran 81.5 turns. The clean subset is 910 of 1,376 quadruples in 232 of 354 clusters, and on it $R - C$ is -0.1033 $[-0.2108, +0.0055]$: the interval no longer excludes zero. That subset's minimum detectable effect is about 0.16 achievements, so it cannot resolve an effect of this size either way. We report both figures and claim only that the point estimate is stable under the removal, not that the contrast survives it.

H.2 The cross-fitted ceiling estimator uses two of the six repeats

The executor in experiment 2 is `claude-opus-4-6`: 100 prefix campaigns produced 193 checkpoints; branching four ways with six repeats gave 9,208 branches, 2,302 complete quadruples, over 397 map clusters. The same collector, the same opportunity rule, the same shared environment RNG per quadruple, the same frozen analysis script.

The frozen cross-fit picks the argmax on one repeat and scores it on the other, over the two lowest-indexed repeats per checkpoint, in both orders. With six repeats available, 6×5 ordered pairs exist and the frozen script uses two of them. A post hoc variant over all ordered pairs was computed after the fact and gives a different point estimate; we do not report it, because choosing an estimator after seeing both is the exact failure §6.6 is about. This is a place a reviewer should push: the pre-registered estimator is defensible but statistically wasteful, and we would welcome an argument for a better one that can still be fixed in advance.

The reported strong-regime ceiling is -0.0013 $[-0.1188, +0.1213]$ with a permutation null mean of $+0.0003$ and $p = 0.509$. A p of one half is not a near-miss; it is the estimator reporting that the observed ceiling is exactly what label-shuffling produces.

H.3 The advice contrast across regimes: the structural caution

Experiment 2 was collected in two stages: 100 maps first, then 300 more. §6.4 gives the governing argument — the $R - C$ point estimate moved monotonically toward zero as clusters accumulated, $+0.123$ at 99 clusters, $+0.074$ at 288, $+0.058$ at 397 — and the rest of the caution is structural.

The experiment 2 interval $[-0.0622, +0.1757]$ spans no effect, a large positive effect and a small negative one. The experiment's minimum detectable effect is 0.190 achievements, so $+0.058$ is below what it could resolve in any case. And the two experiments differ in more than executor capability — repeat count, cluster count, and the fraction of prefix campaigns carrying infrastructure errors all differ — so even a resolved difference would not isolate capability. We did not randomise executor within map, which is what a moderation claim would require.

We therefore report $R - C$ as resolved and negative in the weak regime, unresolved in the strong regime, and we make no capability-moderation claim. What would settle the question is a single pre-registered test of the regime-by-R-C interaction with executor randomised within map, at roughly 500 clusters per regime.

H.4 CAMPAIGN-LEVEL ARMS: INTERVALS, AND TWO QUALIFICATIONS

The campaign-level arms of Table 1 run over 160 held-out test maps, one model seed, frozen hyperparameters. $B2$'s $K = 16$, $B3$'s threshold of 0.08, $B5$'s fraction of 0.5 and the manager checkpoint were all selected on the validation split before any test map was read; the freeze is recorded in `manifests/freeze_before_test_v3.json`.

Against C the differences are $B1$ -0.075 $[-0.431, +0.300]$, $B2$ -0.019 $[-0.413, +0.363]$, $B3$ -0.006 $[-0.388, +0.356]$, $B4$ -0.025 $[-0.406, +0.344]$ and $B5$ -0.181 $[-0.538, +0.156]$ achievements — every interval spans zero and is about four times the width of the largest point estimate. We draw no conclusion from the ordering, in either direction: the correct reading of Table 1 is that at 160 maps and one seed this comparison has no resolving power, and that a reader who wants to know whether management helps at the campaign level should not use these rows to decide.

Two qualifications attach to specific arms. $B5$ is a weak substitute: no trained AdaCoM-style context editor exists in this harness, so what ran is an untrained, model-free summarisation rule (§G.3). $B5$ finishing last rules out nothing about learned context management. $B3$ reads the risk trunk of the same frozen manager checkpoint that was selected on validation; the trained manager enters this table only through that trunk, and no separately-trained policy arm appears as its own row.

H.5 COST AND PER-ARM TOKEN ACCOUNTING

Cost is measured rather than argued. In the experiment-1 branch collection, N spends 566M input tokens against C's 464M, about a fifth more, for the same turn allowance. These two figures come from the per-arm cost totals in `results/stageA/main_table.json`; they are the only numbers in this paper drawn from a cost ledger rather than from the frozen analysis script, and a reviewer checking our arithmetic should start there.

The mechanism is not subtle: the handover package and the suggestion are anchor messages, kept verbatim at the head of the session and re-sent on every subsequent turn (verified in `succession/worker/worker.py`). Since $N - C$ is -0.0167 $[-0.1114, +0.0781]$, the measured position is that handover costs a fifth more input tokens for a benefit this design cannot detect.

H.6 WHAT IS NOT HERE, IN FULL

Only Craftax was collected. The FLE and SWE-bench Pro adapters are implemented and tested but no data was gathered through them, so every number in this paper comes from one environment and one task family. The manager's LoRA adapters were frozen during training — an MPS out-of-memory failure at 62 GB — and only the value head trained, so "learned manager" here means a trained head on a frozen backbone. The weak executor ran with its reasoning field disabled for throughput, which costs capability (0.082 against 0.108 utility on identical maps) and buys a $17\times$ reduction in output tokens. Finally, the test split was touched by two two-map harness smoke tests before the final freeze; those runs informed no tuned value and were discarded, and the sequence is recorded in `manifests/freeze_before_test_v3.json`. We record it rather than hope no one asks.

H.7 MATERIAL MOVED OUT OF THE MAIN RESULTS SECTION

Reading the strong-regime ceiling. A p of one half is not a near-miss; it is the estimator reporting that the observed ceiling (-0.0013 $[-0.1188, +0.1213]$, permutation null mean $+0.0003$, $p = 0.509$) is what shuffling the action labels produces. The cross-fitted estimator takes the argmax on one repeat and scores it on the other, which is what removes the upward bias of the in-sample maximum.

Why the strong executor was the interesting test. If per-state management is worth anything, it should show on the executor that can act on advice; that is why experiment 2 was run at $3.6\times$ the task competence of experiment 1, and why its null is reported rather than set aside.

Scope of the resolved contrast. R and C differ in exactly one respect (§3.2), so $R - C = -0.1562$ $[-0.2467, -0.0648]$ is a measured effect of one manipulation under one executor in one environment, not a theory of advice.

Sample sizes that would settle the open questions. In experiment 1, $N - C$, $F - C$ and $N - F$ all sit well inside the minimum detectable effect of 0.129 achievements. In experiment 2 the largest point estimate, $R - C = +0.123$, falls below that experiment's own threshold of 0.190; resolving an effect of 0.12 achievements there would need roughly 500 clusters. Settling whether the sign of $R - C$ is capability-dependent needs a pre-registered test of the regime-by-R-C interaction with the executor randomised within map, at roughly 500 clusters per regime.

The methodological lesson of the withdrawn ceiling. Any per-state oracle computed over repeated noisy measurements should be printed beside its permutation null; we did not, and were wrong for it.

I THREATS TO VALIDITY IN FULL

§7 states each threat in compressed form. This appendix gives the full discussion of every one of them, unchanged; no threat appears here that is absent from the main text, and no number differs.

Infrastructure contamination in experiment 1. Of the 400 prefix campaigns that generate experiment 1's checkpoints, 162 terminate with `termination_reason=infra_error`. Every one of the 162 is the same failure: an HTTP 429 rate-limit rejection from the shared inference endpoint, raised inside the executor's call and not recoverable by the collector. 34 of them ran zero turns and produced no checkpoint. The median across the 162 is 81.5 turns. A truncated prefix is not a neutral loss. It censors the prefix length distribution, and it censors it in a way that is plausibly correlated with how long an agent survives: a run that lives longer is exposed to more requests and therefore to more chances of being rejected. The checkpoints drawn from those prefixes are therefore not a uniform sample of the stall states the design intends to study.

We report both the full and the clean analysis. $R - C$, the one contrast that resolves on the full data, is -0.1562 $[-0.2467, -0.0648]$ there. On the subset descending only from uncontaminated prefixes — 910 of 1,376 quadruples in 232 of 354 clusters — it is -0.1033 $[-0.2108, +0.0055]$, which does *not* exclude zero.

We do not claim the contrast survives contamination removal. Two readings are available and these data cannot separate them. At 232 clusters the minimum detectable effect is about 0.16 achievements, so a true effect of 0.10 would not be resolvable there even if it were real; on that reading the clean subset is simply underpowered and the point estimate is stable ($-0.156 \rightarrow -0.103$). Alternatively part of the full-data effect is contamination. We report both numbers, prefer neither, and note that the one contrast this paper resolves is the one most exposed to its worst data-quality problem. Experiment 2 does not inherit this problem in the same form, but it also does not resolve $R - C$ in either direction ($+0.0578$ $[-0.0622, +0.1757]$), so it cannot be used to arbitrate.

The weak executor ran with its reasoning field disabled. Experiment 1's executor is a reasoning model run with that field off. On identical maps the same model scores 0.082 utility with reasoning disabled against 0.108 with it enabled. We chose the disabled configuration because it is $17\times$ cheaper in output tokens and the design does not fit the available serving throughput otherwise. All four arms share the executor, so this does not subsidise an arm. It does move the regime. A weaker agent may be more responsive to advice and to a rebuilt context than a stronger one, which is one reading of why $R - C$ resolves in experiment 1 and flips sign without resolving in experiment 2 — and we cannot distinguish that reading from the alternative that experiment 2 is simply underpowered at 99 clusters. The reasoning-enabled replication at experiment 1's scale was never run.

The outcome range is narrow, in both directions. Across 400 experiment-1 runs only 16 distinct achievements are ever unlocked; 51 of the 67 never fire. The mean is 4.0 achievements. The utility denominator is 67, but the outcome effectively varies over a much smaller set, and no intervention can produce a difference on a dimension that never moves for any arm. Experiment 2's executor reaches 14.4 achievements and utility 0.2145 against 0.0597, which is why we ran it: it is the check on whether the null is a floor effect. The null holds there too. We regard that as the single strongest piece of evidence in the paper, and also as the one most limited by sample size, since experiment 2 has 99 clusters against experiment 1's 354.

Power bounds the null, and the bound is not tight. The paired differences decompose into a within-checkpoint (repeat) standard deviation of 1.566 achievements against a true between-cluster standard deviation of 0.450: 76% of the apparent between-cluster variance is repeat noise. The minimum detectable effect at 80% power is 0.129 achievements in experiment 1 (354 clusters, 2 repeats) and 0.190 in experiment 2 (397 clusters, 6 repeats). Nothing observed in experiment 2 clears its bar. So the honest statement of experiment 2's result is not "there is no effect" but "there is no effect of size 0.190 achievements or larger". Resolving a 0.12-achievement effect would take roughly 500 clusters. A reviewer who believes the interesting effects live below 0.129 achievements should read this paper as uninformative rather than as negative, and we would not argue.

B5 is a weak substitute and forbids the conclusion it looks like it supports. The context-management baseline B5 is not a trained context editor. No trained AdaCoM-style editor exists in this codebase, so B5 is an untrained, model-free summarisation rule that deletes repeated action signatures, merges consecutive same-error turns, and truncates over-long payloads. It is labelled `is_weak_substitute` in the baseline registry and in the freeze manifest. B5 is last in the campaign table at 0.06856 against 0.07127 for doing nothing, and that comparison *does not* rule out learned context management. It rules out one fixed heuristic. Any reading of the campaign table as evidence against context management as a class is a reading we disclaim in advance.

The manager's adapters were frozen. The plan trains LoRA adapters and a value head on the manager backbone. Only the value head trained: the LoRA path ran out of memory at 62 GB on the available MPS device. The manager therefore learns a value function over frozen features. This is a limit on the manager arm specifically, not on the four-cell experiment, but it means the campaign table's manager entry is a lower bound on what a trained manager could do, and the withdrawn "4.1% of ceiling" claim (below) is not evidence about learned managers either.

One environment. Only Craftax was collected. The FLE and SWE-bench Pro adapters are implemented and tested end to end, and neither was ever run for data. Every claim in this paper is a claim about one text interface over one survival game. The mechanism we do observe in the weak regime — advice without context repair harms, and rebuilding the context cancels that harm ($R - C = -0.1562$, $N - R = +0.1395$ $[+0.0624, +0.2150]$) — is exactly the kind of result that could be specific to an environment where the context is dominated by repetitive observation frames. We have no cross-environment evidence and do not imply any.

The test split was read before the freeze, and the freeze record said otherwise. The campaign table uses 160 held-out test maps with hyper-parameters selected on validation before any test map ran: B2's $K = 16$, B3's threshold 0.08, B5's fraction 0.5, and the manager checkpoint. Version 2 of the freeze manifest asserted that no test map had been read before the freeze. That assertion was false. Two smoke tests of the main-table harness had been run with a 2-map argument, which resolves to the first two test seeds, and their utility numbers were printed to a log. Those runs informed no tuned value, and the directory was deleted before the real table was collected. We superseded the manifest rather than amending it, so both the false claim and its correction are in the record (`manifests/freeze_before_test_v3.json`, superseding `..._v2.json`). The defect is in the claim, not in the selection, and we would rather a reviewer judge that distinction than take our word for it. Note also that the campaign table is one model seed: it carries no seed variance, and its contrasts have intervals roughly twice the width of any difference in it.

J AUDITED DEFECTS

In an experiment whose output is a difference between four arms, the dangerous failure is not a crash. A crash is loud and attributable. The dangerous failure leaves every arm running, every record well formed, and every number plausible, and moves the contrast. Defects of that shape were found repeatedly across adversarial audit passes of this implementation, and not one of them raised an exception. The first pass is released in full as `docs/AUDIT_FINDINGS.md`; the repository's audit log catalogues twelve numbered findings across four adversarial passes, each with a regression test, and 54 test files cover them. Table 5 lists six; the rest are in `docs/AUDIT_FINDINGS.md` and the audit log, each locked by a regression test that fails against the previous behaviour.

The truncated-allowance defect, in numbers. This row is the clearest case in the paper of an arithmetic error producing a resolved contrast, so we give it in full. Under the pre-fix code, 0 of 1,376 F branches and 0 of 1,376 N branches reached the turn allowance they had been sampled, against 794 and 809 of 1,376 for C and R. After the fix the four counts are 768, 797, 722 and 749.

On the pre-fix data $N - C$ was -0.2478 $[-0.3423, -0.1548]$: an interval excluding zero, in the direction a reader would have taken as succession actively harming the agent. After the fix the same contrast on the same checkpoints is -0.0167 $[-0.1114, +0.0781]$, unresolved. Nothing about the environment, the executor or the estimator changed between those two numbers — only the token allowance that decided how many turns two of the four arms were permitted to take. Both collections are retained in the release.

What these share. In each case a quantity was *computed* where nothing had been *measured*, and nothing downstream could tell. Prose discipline is not a defence, because each defect produces exactly the artifact a careful reader expects to see. Two of the four late ones were caught by a check that looked past the record to the thing the record described — residency of z in the actual context rather than equality of a reference string; turns actually taken rather than turns nominally allowed. The other two were caught by arithmetic on the allowances and by a permutation null. One episode from the earlier passes fixes the general shape: while landing two fixes, two duck-typed decision classes fell out of sync, every executor call raised `TypeError`, each was absorbed by an upstream `except` as an infrastructure error, and the test suite stayed green because it imported the class that was fine. Every episode recorded zero turns and a termination reason. A dead executor does not look like a crash. It looks like a result.

Table 5: Six audited defects. Each ran to completion and emitted a well-formed record. The last four were found late.

What the code did	What it produced	What prevents it now
A `hasattr` guard installed the per-branch environment RNG only if the adapter exposed a particular method name. The adapter exposed a different name.	The derived key was dropped in silence, while every record named an RNG stream that had never been used.	Installation raises when no stream can be installed, and the record stores the stream actually installed.
A broad `except` around the environment step relabelled budget exhaustion as an infrastructure fault; infrastructure rows default to utility 0 and entered the analysis.	Legal terminations booked as faults, then fed in as true zeros, penalising whichever arm exhausted its budget more often.	Separate termination reasons, ledger re-checked after each charge, and infrastructure rows filtered before the four-cell computation with their count reported.
Branch budgets were rescaled to the sampled remaining turns, which divided the tier factor straight back out.	One allowance for all three budget tiers, each record naming the tier it had asked for. A budget-transfer experiment would have reported a null caused by arithmetic.	Tier-scaled draws, and the builder raises when a draw exceeds its own tier's cap.
The turn allowance was sampled in turns and the token allowance scaled from the tier table, which assumes every arm spends the same input per turn. F and N resend their handover package on every turn.	No F or N branch reached its sampled turn allowance under the pre-fix code. The token cap redefined the horizon for two of four arms, biasing $F-C$, $N-C$ and $N-F$ downward; on that data one of the three resolved, with an interval excluding zero, in the wrong direction. See the note below the table.	The anchor cost is added to the input allowance so the turn cap binds for every arm, with the one-time generation cost still charged once.
R's copy of the shared suggestion z was appended to a live session, making it the oldest non-anchor message: the first thing the compressor deletes.	R lost z within a few turns while N, holding it as an anchor, kept it for the whole branch — inflating $N-R$ and the interaction. The sharing audit, which compared reference strings, certified the pair as correctly shared.	z is inserted ahead of the non-anchor region with the anchor count raised, mirroring F and N; residency is measured per branch, and the audit also compares each arm's generator and summariser charges.
The oracle ceiling was computed as the max over four arms $\times$ repeats.	$+1.02$ achievements, reported internally as a $+22\%$ headroom. Max-of-k is biased upward even when all arms have the same expectation.	Cross-fitting, reported beside a permutation null: $+0.0073$ $[-0.0630, +0.0808]$, $p = 0.429$ in experiment 1 and -0.0013 $[-0.1188, +0.1213]$ in experiment 2.

K Where We Think a Reviewer Should Push

1. **Whether experiment 1 is usable at all given $162/400$ contaminated prefixes.** We think the clean-subset re-analysis is the right response. A reviewer may think the right response is to discard experiment 1's $R-C$ and report only experiment 2, which does not resolve it.
2. **Whether two capability regimes are enough to call the ceiling zero.** Both point estimates are at zero with permutation p near 0.5, which we read as the estimator saying per-state choice bought nothing. It is also consistent with an effect below 0.129 achievements that neither experiment could see.
3. **Whether $R-C$ in the weak regime is a mechanism or a regime artifact.** The sign flips in experiment 2 without resolving. We do not know which.
4. **Whether the audits found the defects of the class they went looking for.** Four adversarial passes are not independent replication. Two of the four defects in Table 5's lower half were found only after data had been collected under them, which is direct evidence that the passes were not exhaustive. We have no way to bound what remains, which is why the manifests, per-branch records, exclusion lists, sharing and charge audits, the frozen pre-registered estimator and the regression tests are released: every check named here can be re-run rather than taken on report.

K.1 WHERE A REVIEWER SHOULD PUSH, IN SHORT FORM

Moved here from the introduction.

- *Floor effect?* The weak executor unlocked 16 achievements across 400 runs and never unlocked 51; experiment 2 cannot settle it at an MDE of 0.190.
- *One environment?* Only Craftax ran; the Factorio and SWE-bench Pro adapters are wired but never executed.
- *A fair context-management opponent?* No: B5 is an untrained summarisation rule Yi et al. (2026), so its 0.06856 against 0.07127 for always continuing rules out nothing about learned context management.
- *An undertrained manager?* Its LoRA adapters froze at an MPS out-of-memory (62 GB); only the value head trained, but the ceiling bounds any per-state policy.
- *Clean weak-executor data?* 162 of 400 experiment-1 campaigns carry `termination_reason =infra_error`; clean-subset $N - C$ is $+0.0593$ $[-0.0582, +0.1722]$, $N - F$ $+0.0560$ $[-0.0465, +0.1579]$, $F - C$ $+0.0033$ $[-0.0976, +0.1083]$.
- *An untouched test split?* Two two-map smoke tests hit it before the freeze; they informed no tuned value; logged in `manifests/freeze_before_test_v3.json`.
- *Executors at their best?* No: the weak one had its reasoning field disabled for throughput, so distrust experiment 1 in isolation.

K.2 KNOWN DEFECTS, AS LISTED IN THE SETUP SECTION

Moved here from Section 5.

- **B5 is a weak substitute:** a fixed, model-free summarisation rule, so a result against it does not rule out learned context management.
- **The manager's LoRA adapters were frozen** at MPS out-of-memory at 62 GB: only the value head trained, so the manager is a value head on a frozen backbone, not the trained policy of §4.
- **Infrastructure loss in experiment 1.** 162 of the 400 prefix campaigns carry `termination_reason =infra_error`; on the clean subset (910 quadruples, 232 clusters) $R-C$ is -0.1033, $[-0.2108, +0.0055]$, no longer excluding zero. We have not established that the loss is independent of map difficulty.
- **One model seed** and **one site**: all of this is Craftax.
- **A defect in the freeze record.** Two 2-map smoke tests ran against test seeds before the final freeze; they informed no tuned value, but an earlier freeze record's claim that no test map had been read was false and is superseded.