

Plans Don't Persist: Why Context Management Is Load Bearing for LLM Agents

Aman Mehta
Snowflake AI Research
aman.mehta@snowflake.com

Anupam Datta
Snowflake AI Research
anupam.datta@snowflake.com

Abstract

Long-horizon agents depend on context management: systems compress, summarize, and evict old tokens so tasks can continue beyond finite windows. That is safe only when dropped information is no longer needed or has been internalized. Plans are the stress case: they are written early, used for many steps, and first to be evicted. We introduce *replay pairing*, a diagnostic that runs the same trajectory with and without the plan in history and measures hidden-state cosine distance. On Llama-3.1-70B, plan signal spikes to 0.453 one step after the plan, then falls $4.1\times$ in a single action-observation step; HotpotQA falls $12.4\times$. This is evidence that standard LLM agents do not carry plans forward as persistent state, and instead depend on the plan remaining in context. A layer- L_{32} probe detects this decay as a diagnostic, not as proof that it reads plan content itself. Reasoning models add a measurement confound: their `<think>` traces re-derive plan content, so standard stripping leaves plan evidence in the stripped condition. We name this the *reasoning-trace confound* and fix it with strict stripping, which removes prior `<think>` blocks from the stripped run only. It recovers +163% of the step+1 signal in-sample and +153% held out, while not meaningfully changing non-reasoning Llama (+4.8%). On DeepSeek-R1-Distill-Llama-70B, a Llama-trained probe transfers at AUROC 0.748 ($p=6\times 10^{-4}$), while R1-specific probes reach 1.000, suggesting R1 encodes plan signal in a different hidden-state direction. Finally, a compression stress test shows the practical cost: naive plan eviction cuts ALFWorld success by 34.7pp, while probe-gated re-surfacing does not recover it. The contribution is a measurement and stress-test framework showing that agent-critical information can be context-resident rather than persistent. Context management is load bearing, but plan protection alone is not enough.

1 Introduction

Context management is becoming load-bearing for LLM agents. To run long horizons within a finite window, deployments compress, summarize, and evict old context, betting that dropped tokens do not change behavior. Whether that bet is safe depends on where information lives: in *context-time* memory (text in the window, re-read each step) or in persistent internal state that survives once the text is gone. The agent's *plan* is the sharpest case. It is written once, early, so it is first to be evicted. If the plan is held in internal state, dropping its text is free; if it is only re-read each step, dropping it silently breaks an agent that passed every behavioral test while the text was still visible. This did not matter when contexts were short and nothing scrolled away; modern context management makes it matter. Which mechanism holds is the question this paper answers.

Planning frameworks like ReAct (Yao et al., 2022), Chain-of-Thought (Wei et al., 2022), Tree of Thoughts (Yao et al., 2023), and Reflexion (Shinn et al., 2023) all assume the same thing: the agent writes a plan, and the plan guides what it does next. Behavioral alignment metrics confirm that it does. But behavior alone cannot tell two mechanisms apart. The plan could be held in hidden state across the trajectory (internalized), or it could live only as text in

the window and be re-read each step (context-time). While the plan text stays visible, both mechanisms produce the same actions. They differ only once the text leaves the window, which is exactly where context compression, summarization, and KV-cache eviction operate.

To separate the two, we use replay pairing (Figure 1). For each trajectory we run two matched conditions: A keeps the plan in history; B replays the same trajectory with the plan removed. The plan-fidelity signal is the cosine distance between the two hidden states at each step. It measures how much the hidden state changes when the plan is taken out of history. This signal also mixes in position, history length, and discourse structure (§3); length- and content-matched controls are the main follow-up (§9).

In a Llama-3.1-70B ReAct agent on ALFWorld (Shridhar et al., 2020), the plan signal jumps to 0.453 at step+1, drops $4.1\times$ in one action-observation cycle, and settles near 0.027 by step+5. HotpotQA drops even faster ($12.4\times$). A Ridge probe at the peak layer (L_{32}) predicts the signal at $R^2 = 0.875$, separates active from decayed plans at AUROC 0.999, and transfers to HotpotQA with no retraining. One caveat: step index is also almost perfectly decodable here ($R^2=0.978$ at L_{32}), so part of the AUROC reflects trajectory phase, not plan content. We have not run the mixed-effects control that would separate the two (§5). The headline reading holds either way: the plan-presence signal is a context-window effect that fades within a step, not a state the agent keeps.

Reasoning-trace contamination. Reasoning models emit `<think>...</think>` traces before each action (DeepSeek-AI, 2025). Run unchanged on DeepSeek-R1-Distill-Llama-70B, replay pairing makes the step+1 signal look $4.5\times$ smaller than Llama’s (0.022 vs. 0.099), as if R1 barely encodes the plan. This is a measurement artifact. The model re-states plan content inside prior `<think>` blocks, and standard stripping leaves those blocks in place, so both conditions still carry the plan and the distance under-counts. Strict stripping removes prior `<think>` blocks from condition B only. It recovers +163% in-sample, +153% held-out, and changes Llama’s signal by just +4.8%. The Llama-trained L_{32} probe on R1 strict-stripped states ($n=19$) reaches AUROC 0.748 ($p=6\times 10^{-4}$); R1 self-probes reach 1.000 on two distilled families, along a direction 89.3° from Llama’s.

Scope. The claims are representational, not behavioral: strict stripping has small, non-significant task-success effects, and uniform L_{32} steering is a null. Content-specific controls and stronger causal tests remain future work.

Contributions.

1. Evidence that the plan is a context-time object: in standard LLMs it decays $4-12\times$ within one step out of hidden state, with cross-domain and within-family replication.
2. A compression stress test: naive plan eviction cuts ALFWorld success by 34.7pp, and probe-gated re-surfacing does *not* recover it (§8).
3. Replay pairing as a measurement method, plus the *reasoning-trace confound* and strict stripping fix, which recovers +163% of step+1 signal on R1 and is a verified no-op on non-reasoning Llama.
4. Probe-transfer and self-probe evidence that corrected hidden states carry plan-related signal, with R1 using a different hidden-state direction from Llama.

2 Related Work

Context-time vs. weight-time memory. A growing line of work asks where information a model uses actually resides: in *context-time* memory (text in the window, supplied at inference) or *weight-time* memory (absorbed into parameters). Retrieval-head analysis shows that long-context factuality is carried by a small set of attention heads that copy from the window rather than from parameters (Wu et al., 2024); multi-turn studies find that models “get lost” when relevant content is distributed across turns instead of re-stated (Laban et al., 2025). Context compression, summarization, and KV-cache eviction all bet that some tokens can be dropped without changing behavior. Our plan-fidelity measurement is a direct test of that bet for one important token type: it asks whether the plan is re-read

from the window each step or held in durable state, and finds the former in standard LLMs. The plan has no weight-time backup, which places it squarely in the context-time regime, where eviction is unsafe.

Planning and faithfulness in LLM agents. The standard paradigm interleaves reasoning and acting (Yao et al., 2022; Wei et al., 2022; Yao et al., 2023; Shinn et al., 2023; Huang et al., 2022). These frameworks assume explicit plans improve behavior, which behavioral metrics confirm, but they do not test the mechanism by which the plan influences subsequent steps. Repeated-run studies show that multi-step agents can diverge early under identical inputs, and that consistency can amplify both correct and incorrect interpretations (Mehta, 2026a;b). Our work asks a complementary question: whether the plan representation itself persists across those steps. We extend plan-faithfulness analysis to the representational level and find that high behavioral alignment can coexist with $12\times$ representational decay.

Hidden-state representations. LLM hidden states linearly encode many semantic properties (Burns et al., 2023; Marks & Tegmark, 2024; Zou et al., 2023). Circuit tracing (Ameisen et al., 2025) and the Tuned Lens (Belrose et al., 2023; nostalgebraist, 2020) look at the layer dimension within one forward pass. Replay pairing operates on a different axis: the temporal dimension of multi-step agent behavior. The question is how plan information persists across forward passes.

Interventions and reasoning-model probes. Activation steering (Turner et al., 2023; Zou et al., 2023), inference-time intervention (Li et al., 2023), and self-refinement (Madaan et al., 2023) target hidden states or context. For reasoning models, several works probe R1 hidden states in single-turn settings (Liu et al., 2025; Wang et al., 2025a; Chen et al., 2025); Wang et al. (2025b) report poor instruction-following inside traces and Wu et al. (2025) propose Thinking Intervention; Stolfo et al. (2025) apply activation differencing single-turn. The piece we add is identifying the contamination that hits paired-trajectory methods on reasoning models in multi-turn agent loops, and a fix.

3 Method

3.1 Agent Setup

We run two classes of model as ReAct agents (Yao et al., 2022). The non-reasoning model is Llama-3.1-70B-Instruct (Grattafiori et al., 2024) (primary), with Llama-3.1-8B-Instruct for within-family replication. The reasoning model is DeepSeek-R1-Distill-Llama-70B (DeepSeek-AI, 2025), a Llama-3.1-70B base distilled from DeepSeek-R1 reasoning traces. It produces explicit `<think>...</think>` blocks before each action. We inject a `<think>\n` prefix at generation time so the reasoning trace always opens with the right tag (the Llama chat template does not emit the opening `<think>` on its own).

The primary environment is ALFWorld (Shridhar et al., 2020), a text-based household environment where the agent completes tasks by navigating rooms and manipulating objects through a natural-language action space (`go_to`, `pick_up`, `put`, `heat`, `cool`, `clean`, `examine`). It spans six task types ranging from 3-4 to 6-8 plan steps. We use the text-only (non-visual) variant, following standard ReAct-on-ALFWorld baselines (Yao et al., 2022).

The secondary environment is HotpotQA (Yang et al., 2018). We collect 30 trajectory pairs using the same replay-pairing protocol (§4) and the same model and prompt configuration, for a direct cross-domain comparison.

Models run on $8\times$ H200 GPUs with pipeline parallelism. Full hidden states (\mathbb{R}^{8192}) are captured at the last token position at every forward pass.

3.2 Plan Elicitation

To study plan faithfulness we first need an explicit plan, so we inject a guard message asking the agent for a multi-step plan:

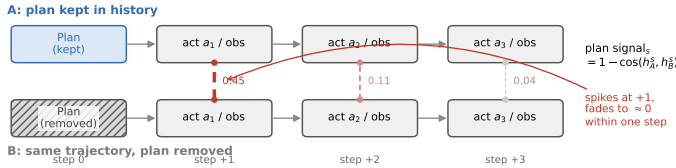


Figure 1: Replay pairing. The same trajectory is run twice: A keeps the plan in history; B replays the identical actions and observations with the plan removed. The plan-fidelity signal is the per-step cosine distance between the two hidden states. It spikes one step after the plan and fades to near zero within a step: a plan re-read from context, not stored in hidden state.

“Before continuing, please state your complete plan for finishing this task. Describe each step in order: where you will go, what you will pick up, what actions you will perform, and where you will place the object.”

The agent replies with a sequential plan, which then stays in history for the rest of the task. We inject at step 2 in ALFWorld (step 4 in HotpotQA): late enough that the agent has seen the task and initial context, early enough to leave a long post-plan trajectory for measurement.

3.3 Replay Pairing

For each agent run we create two matched conditions.

Condition A (plan-present). The agent runs normally with the guard at step 2; the plan enters history. We capture hidden states $\mathbf{h}_A^{(s,\ell)}$ at every step s and layer ℓ .

Condition B (plan-stripped replay). We replay A’s exact trajectory (feeding A’s observations and action outputs) but strip the guard exchange, and call the LLM on this plan-free history to produce $\mathbf{h}_B^{(s,\ell)}$. B’s output is discarded; A’s trajectory drives the conversation.

Both conditions see the same action-observation sequence; they differ only in whether the plan exchange is in history. Plan-fidelity signal at step s and layer ℓ is:

$$\text{PlanSignal}^{(s,\ell)} = 1 - \cos(\mathbf{h}_A^{(s,\ell)}, \mathbf{h}_B^{(s,\ell)}) \tag{1}$$

where step indices are relative to the guard: step 0 is the guard step, step +1 is the next agent turn, and so on. Pre-plan steps ($s < 0$) act as a validation baseline. A and B are identical before the guard fires, so the signal at steps -2 and -1 should be zero. In practice it is below 10^{-3} (Appendix A).

What plan-fidelity signal does and does not measure. A and B differ in plan content, but also in history length, the relative positions of subsequent tokens, and discourse structure. The pre-plan baseline rules out replay-mechanics confounds but does not separate plan content from these. The strongest content-isolation controls (length-matched filler, shuffled-plan placebos, paraphrase and ablation) are the priority next step (§9); “plan-fidelity signal” is the cosine distance attributable to the plan exchange’s presence, a composite that upper-bounds the content-specific component.

3.4 Strict Stripping for Reasoning Models

Standard replay pairing assumes that any hidden-state difference between A and B reflects only the conditioning input, i.e. the plan. For reasoning models this fails. As the model works through each subsequent action it re-states parts of the plan, both inside its `<think>...</think>` block and, to a lesser degree, in the retained Thought:/Action: text. When those turns are replayed verbatim into condition B, that re-stated plan content reap-

pears even though the original plan exchange was stripped, so both conditions still encode the plan and the cosine distance under-counts the true plan-following signal.

Strict stripping is a small change to the protocol. In addition to removing the plan exchange from B’s history, also remove every complete `<think>...</think>` block from each prior assistant turn. Content after `</think>` (Thought: and Action: lines) is kept so that B still sees the same external action-observation sequence as A. The protocol has two sanity checks, both tested in Section 7: on models that emit no `<think>` content it should be a no-op, and on reasoning models the strict-stripped plan signal should exceed the standard plan signal.

What strict-stripping controls do not yet establish. The operator is asymmetric (applied to B only) and removes whole `<think>` blocks rather than only plan-relevant spans. The recovered signal could therefore reflect removal of plan-content contamination (our hypothesis), removal of generic reasoning-history content, or an asymmetry-induced shift in B’s local distribution. The Llama no-op (+4.8%; §7) rules out the generic-reasoning mechanism but not the asymmetry one. Three controls would settle it: strip both A and B, replace blocks with length-matched neutral text, or redact only plan spans. These are listed in §9.

3.5 Probe and Task Partitioning

To turn plan signal into a detector we train a Ridge regression probe on plan-present hidden states at the peak layer ℓ^* (Section 4), regressing the scalar plan-signal magnitude $y^{(s)} = \text{PlanSignal}^{(s, \ell^*)}$; a threshold τ converts predictions to a fire/no-fire decision. We use regression rather than binary classification because step index alone is trivially decodable from the residual stream (Appendix A). We collect plan-signal data from 80 ALFWorld tasks (50 train, 30 held out), 30 HotpotQA pairs, and 5+5 R1-Distill-70B ALFWorld tasks for the contamination analysis; all splits are disjoint (Appendix A).

4 Experiments: Plan Signal Decay

4.1 Overall Decay Curve

Figure 3 shows plan signal vs. step offset from the guard on ALFWorld (80 tasks). Pre-plan steps are a sanity check: A and B are identical before the guard, and we observe mean signal $< 10^{-8}$ at steps $-2, -1$ (floating-point noise), confirming the replay pairing is correct. At step $+1$ plan signal jumps to 0.453 ± 0.039 , drops to 0.110 ± 0.032 at step $+2$ ($4.1 \times$ in one cycle), and settles to ~ 0.027 by step $+5$. The initial magnitude nearly matches HotpotQA (0.453 vs. 0.445) though HotpotQA decays faster ($12.4 \times$; §6). In context-management terms: the plan’s footprint in hidden state is nearly gone within one step, so anything downstream that needs the plan must read it back from the window.

4.2 Layer and Task-Type Profile

Step $+1$ plan signal peaks sharply at layer 32 (signal 0.686 ± 0.024), with L_{28}/L_{36} forming a broad mid-network peak; early and late layers are lower. We use L_{32} as ℓ^* . Stratified by task type, all six ALFWorld types show closely matched spikes (0.445-0.474) and decay rates (4.0 - $4.4 \times$): complex plans (6-8 steps) decay as fully as simple ones (3-4), so the dynamics are architectural, not task-level (Figure 2).

5 Detection: Probe for Plan Decay

5.1 Probe Performance

We train a Ridge regression probe on layer- ℓ^* hidden states from 80 ALFWorld tasks (1360 post-plan steps) using 5-fold stratified cross-validation. The probe regresses plan-signal magnitude. A threshold τ converts predictions to a binary fire/no-fire decision. Table 1 reports results at $\ell^* = 32$ together with zero-shot transfer to HotpotQA.

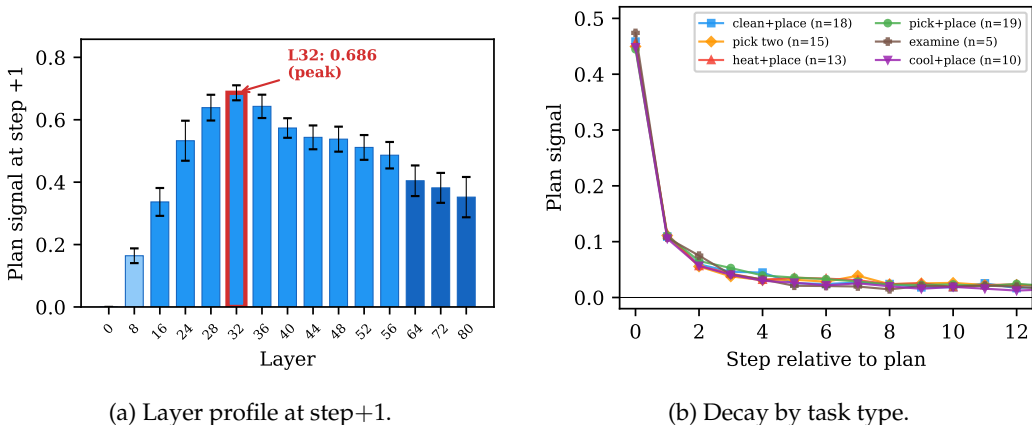


Figure 2: Plan signal is localized in depth but stable across tasks. The peak layer is L_{32} (signal 0.686), and all six ALFWorld task types converge to the same residual.

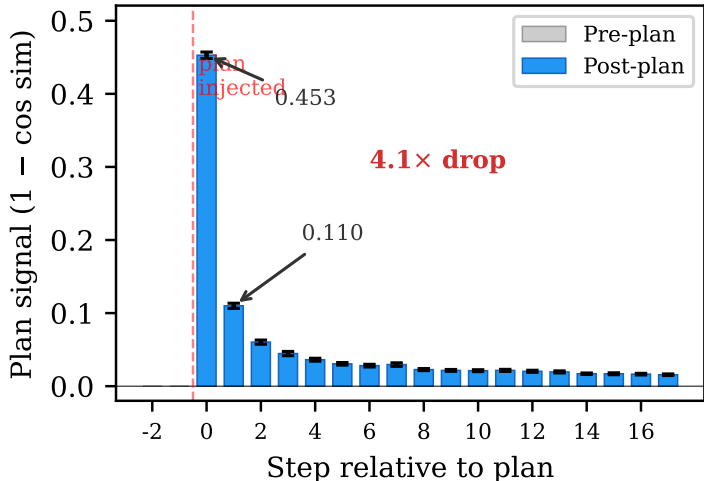


Figure 3: Plan signal (mean \pm SE, 80 tasks) as a function of step offset from guard injection. Pre-plan signal ($s < 0$) is near zero (validation). Signal jumps to 0.453 at step +1 then drops $4.1\times$ in one step.

On ALFWorld, the probe predicts plan-signal magnitude at $R^2 = 0.875 \pm 0.016$, well below the $R^2 = 1$ a trivially-leaked feature would produce. The induced binary decision at $\tau = 0.15$ reaches AUROC 0.999 with balanced precision (0.985) and recall (0.825). Applying the ALFWorld probe direction zero-shot to HotpotQA gives AUROC 1.000 at both L_{32} and L_{40} .

Caveat: probe AUROC near 1.0 is partly step-index leakage. Step index is near-perfectly decodable from the residual stream ($R^2=0.978$ at L_{32} ; Appendix A), and plan-signal magnitude is correlated with step index by construction, so a probe that learns “predict step index” would also score high on the active-vs-decayed contrast. We have not run the discriminating control: a mixed-effects analysis holding step index, task type, plan length, and observation-plan overlap fixed. Until then the AUROC should be read as “at least as informative as step index,” not as a content-specific representation.

The cross-domain transfer is more informative than the in-domain AUROC. At the ALFWorld cutoff $\tau = 0.15$, L_{40} transfer is sharp on HotpotQA (F1 0.968); L_{32} keeps full recall but fires on HotpotQA-specific variance (precision 0.513), which a short isotonic recalibra-

Table 1: Probe performance. ALFWorld: Ridge regression at $\ell^* = 32$, 5-fold CV on 1360 steps across 80 tasks. HotpotQA transfer: zero-shot application of the ALFWorld probe direction to 238 HotpotQA steps (60 positive, 178 negative) at two destination layers. R^2 is reported for regression; binary metrics are computed by thresholding regression output at $\tau = 0.15$.

Metric	ALFWorld	HotpotQA L_{32}	HotpotQA L_{40}
R^2	0.875 ± 0.016	n/a	n/a
AUROC	0.999	1.000	1.000
F1 ($\tau=0.15$)	0.898	0.678	0.968
Precision	0.985	0.513	0.938
Recall	0.825	1.000	1.000

tion lifts to F1 0.920. Some component of the plan-presence direction generalizes across domains beyond what step index would predict, but the operating point needs per-domain calibration.

5.2 Probe Validity, Gradedness, and Calibration

Three controls back the probe (full numbers in Appendix A): a shuffled-label control and step-index regressor confirm the residual fit ($R^2 = 0.875$) captures structure beyond the $R^2 = 0.978$ step-index baseline at L_{32} ; a 4-class step classifier attains macro $F_1 = 0.862$ with strictly monotonic confusion; and the main binary probe is near-perfectly calibrated (Brier $\approx 10^{-6}$).

5.3 Lag Analysis: Early Warning

The probe is useful only if it fires before behavioral failure. Defining $\text{lag} = s_{\text{deviate}} - s_{\text{probe_fire}}$ (deviation = first action with plan-alignment < 0.3 from a Claude Opus 4 judge), of the 80 tasks 31 deviated, and on those the probe fires on average 4.45 steps ahead (median 5), leading deviation in 74.2% of cases. A 4-5 step lead is well within ALFWorld’s 20-step cap, so a context manager could use the probe to decide *when* to re-surface the plan into the window.

6 Cross-Domain and Cross-Model Validation

We test the standard-LLM result along two axes: domain (HotpotQA) and scale (Llama-3.1-8B).

Cross-domain (HotpotQA). On 30 HotpotQA pairs with the same Llama-3.1-70B, step +1 plan signal is 0.445 and drops to 0.036 at step +2 ($12.4\times$, $p < 0.001$); the initial magnitude nearly matches ALFWorld (0.453). The faster drop fits HotpotQA’s observations being Wikipedia passages unrelated to the plan, vs. ALFWorld’s plan-relevant room states. The peak layer is L_{34} (vs. L_{32}); the encoding locus is domain-invariant. The ALFWorld-trained probe transfers zero-shot at AUROC 1.000, with isotonic recalibration lifting L_{32} transfer to $F_1 = 0.920$. The direction is domain-invariant; only the operating point needs recalibration.

Cross-scale (Llama-3.1-8B). On 20 ALFWorld tasks, plan signal peaks at 0.382 and decays $10.9\times$ by step +5, matching the 70B shape; the peak sits at $\sim 37.5\%$ relative depth (L_{12} of 32) vs. $\sim 40\%$ (L_{32} of 80) for 70B, as the attention-reconstruction account predicts: plan signal lives in a depth-relative band, not a fixed layer (full multi-model panel in Appendix A.7).

7 Plan Maintenance in Reasoning Models

In the standard-LLM regime, plans are not stored across steps; they are re-read from context. Reasoning models behave differently, and standard replay pairing does not see the difference.

The apparent deficit. On a 5-task ALFWorld subset, standard replay pairing on DeepSeek-R1-Distill-Llama-70B (DeepSeek-AI, 2025) gives step +1 plan signal of 0.022, versus 0.099 for Llama-3.1-70B on the same tasks, so R1 looks $4.5\times$ weaker. Taken at face value, a reasoning model that emits explicit <think> traces appears to encode the plan less than a standard instruct model.

Three hypotheses, one cause. Dilution (a uniform R1/Llama ratio across depth) is ruled out: the ratio varies $4.4\times$ across layers and the models peak at different layers (L_{72} vs. L_{32}). Position mismatch is small (Llama exceeds R1 at every matched position). The remaining cause is contamination via re-derivation: re-stated plan content inside prior <think> blocks reappears in B’s history, so both conditions encode plan content (Appendix A).

Strict-strip correction and verification. Re-running B with strict stripping raises step +1 plan signal from 0.022 to 0.058, a +163% change; every in-sample task moves the same way (Appendix A.5) and the R1-Llama gap shrinks from $4.5\times$ to $1.7\times$. Five held-out tasks of unseen types yield +153%. On non-reasoning Llama strict stripping is a +4.8% no-op, ruling out a generic reasoning-history mechanism. The recovery is large but partial, and the residual $1.7\times$ gap is consistent with plan content also re-stated in the retained Thought:/Action: text, which strict stripping does not remove; redacting plan spans wherever they occur is the clean follow-up (§9).

Probe transfer. The Llama-trained L_{32} probe applied zero-shot to R1 strict-strip ($n=19$) reaches AUROC 0.748 ($p=6\times 10^{-4}$); R1 self-probes ($n=85$) reach 1.000, in a direction 89.3° from Llama’s L_{32} (Table 5). R1-Distill-Qwen-32B is the same (self-probe 1.000, $n=85$). On a third family, Qwen3-32B-native (enable_thinking=True; $n=15, 270$ step pairs), the L_{26} self-probe fits per-step plan-fidelity at $R^2=0.997$ but its binary spike-vs-decay AUROC is only 0.616: Qwen3-native shows persistent plan-conditional drift (cos-distance rises from 0.030 at step +2 to ~ 0.13 by step +19) instead of spike-then-decay. So linear encoding of plan presence holds across all three reasoning families; the spike-decay timing is specific to distilled R1.

Behavioral check (cross-family, $n=25$). Across R1-Distill-Llama-70B, Qwen3-32B-thinking (Qwen Team, 2025), and their non-reasoning baselines, behavioral effects are small and not significant: both reasoning models score 0/25 raw, and strict stripping moves Qwen3-thinking to 2/25 (matching its no-thinking twin) while R1 stays 0/25 (Fisher $p=0.49$). Our claim is representational, not behavioral; it rests on the paired-trajectory and probe-transfer numbers above (full Phase-1 detail in Appendix B, Table 8).

8 When Context Management Fails: A Compression Stress Test

The decay finding makes a falsifiable operational prediction: if the plan is a context-time object with no internal backup, a compression policy that evicts plan tokens should sharply hurt task success, and the early-warning probe (§5.3) should be able to re-surface the plan in time to prevent the loss. We test both on 30 ALFWorld tasks ($\times 5$ runs = 150 runs/policy, Llama-3.1-70B, 20-step cap) under four context policies at a matched budget (keep_recent=4 messages plus the system prompt): none (no eviction); naive (system prompt + last 4 messages); plan_protected (naive with the plan span always pinned); and probe_gated (naive, re-injecting the plan only on steps where the probe fires). Significance is paired permutation (10k) with bootstrap 95% CIs (Table 9 in Appendix C).

Eviction is catastrophic; protecting the plan does not fix it. Naive eviction nearly thirds task success ($56.7\% \rightarrow 22.0\%$, -34.7pp , $p < 0.001$): agents keep no usable plan once its tokens leave the window, exactly as the decay curve predicts. But neither plan-aware policy recovers ($p=0.89$ and 0.67 vs. naive), even though probe_gated re-surfaces the plan 6.1

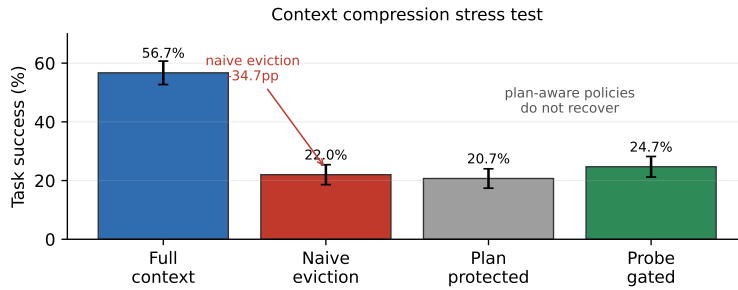


Figure 4: Context-compression stress test on 30 ALFWorld tasks and 5 runs per task. Naive eviction sharply reduces task success, while two plan-aware policies do not recover performance at the same context budget. Error bars show standard errors.

times per run. The likely reason: at `keep_recent=4` the evicted context is mostly recent observations and actions, the working state the agent needs, which re-surfacing a stale plan does not restore. So the lesson is sharper than “protect the plan”: plan tokens are unsafe to drop, but plan-protection alone is not a compression fix; plan-fidelity monitoring is a diagnostic, not a turnkey eviction policy.

9 Discussion and Limitations

Implications for context management. The plan is a case study in a general question: which tokens an agent can safely drop. The same replay-pairing test applies to other tokens that may live only in context, such as safety instructions, constraints, and tool schemas. Reasoning models add a second mechanism: their `<think>` traces re-derive plan content step by step, partially substituting for keeping the plan in the window (and why naive replay pairing under-counts on them). Since probe-gated re-surfacing did not recover the static-plan loss (§8), whether such a self-refreshing scratchpad survives compression better than a static block is an open question we hope this workshop takes up.

What we do not establish. The contribution is methodological and representational, not behavioral. Open controls remain: plan-content specificity vs. position/length/discourse; whether the orthogonal R1 direction is a distinct concept or a rotated encoding; strict stripping specificity; a `keep_recent` budget sweep; probe causality beyond step index at scale ($n \geq 100$); and replication beyond our models and domains.

References

- Emmanuel Ameisen, Tom Conerly, et al. Circuit tracing: Revealing computational graphs in language models. *Anthropic Research Blog*, 2025.
- Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.
- Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2023.
- Chen et al. Tracing the traces: Latent temporal signals for efficient and accurate reasoning. *arXiv preprint arXiv:2510.10494*, 2025.
- DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Nature*, 2025.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aieleen Letman, Akhil Mathur, Alan Schelten, Amy Yang, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

- Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.
- Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*, 2025.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 2023.
- Liu et al. Reasoning models know when they’re right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*, 2025.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 2023.
- Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2024.
- Aman Mehta. When agents disagree with themselves: Measuring behavioral consistency in LLM-based agents. *arXiv preprint arXiv:2602.11619*, 2026a.
- Aman Mehta. Consistency amplifies: How behavioral variance shapes agent accuracy. *arXiv preprint arXiv:2603.25764*, 2026b.
- nostalgebraist. interpreting GPT: the logit lens. LessWrong, 2020.
- Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 2023.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.
- Alessandro Stolfo et al. Improving instruction-following in language models through activation steering. *arXiv preprint arXiv:2410.12877*, 2025. ICLR 2025.
- Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Leela Castriato. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.
- Wang et al. From reasoning to answer: Empirical, attention-based and mechanistic insights into distilled deepseek r1 models. *arXiv preprint arXiv:2509.23676*, 2025a.
- Wang et al. Reasonif: Large reasoning models fail to follow instructions during reasoning. *arXiv preprint arXiv:2510.15211*, 2025b.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 2022.
- Wu et al. Effectively controlling reasoning models through thinking intervention. *arXiv preprint arXiv:2503.24370*, 2025.
- Wenhao Wu et al. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*, 2024. ICLR 2025 Oral.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 2023.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Implementation Details

Models, decoding, infrastructure. Llama-3.1-70B-Instruct (Grattafiori et al., 2024), Llama-3.1-8B-Instruct, and DeepSeek-R1-Distill-Llama-70B (DeepSeek-AI, 2025) served on $8 \times H200$ GPUs with pipeline parallelism. Decoding settings differ by experiment. The paired-trajectory experiments (Sections 4 to 7, $n=80$ ALFWorld and $n=30$ HotpotQA) and the early-warning analysis use temperature 0.7, top- p 0.9, max-new-tokens 512, single run per task. The cross-model behavioral sweep (§7) uses $n_{\text{runs}}=3$ at $T=0.7$ for reasoning models and $n_{\text{runs}}=1$ at $T=0$ for non-reasoning baselines. Deterministic decoding on non-reasoning baselines keeps variance out of the headline non-reasoning number; reasoning models are non-deterministic by default so we average over runs instead. The two settings are deliberate, and the non-reasoning Phase 1 condition is the only one with $T=0$. We use the ReAct (Yao et al., 2022) two-shot prompt template (pick_and_place_simple, pick_heat_then_place_in_recep). Hidden states are extracted at the last-token position; layer sampling stride 4 for layers 24-56 and stride 8 elsewhere (15 sampled layers for 70B, 9 for 8B).

Compute budget. Wall-clock per experiment on $8 \times H200$ (70B-class) unless noted. Paired-trajectory ALFWorld collection ($n=80$ tasks $\times 2$ conditions, single-pass replay with hidden-state caching) takes about 36 H200-hours. HotpotQA cross-domain replay ($n=30$ pairs) takes about 8 H200-hours. R1-Distill-Llama strict-strip recovery ($n=10$ pairs $\times 4$ modes: raw, strict, in-sample, held-out) takes about 22 H200-hours. The Phase 1 cross-model behavioral sweep (6 conditions $\times n=25$ tasks, reasoning $n_{\text{runs}}=3$) takes about 58 H200-hours total. R1-Distill-Qwen-32B and Qwen3-32B-thinking probe-transfer extensions ($2 \times H200$ each, $n=15-85$ paired) take about 14 H200-hours combined. The steering sweep ($n=150$ paired observations, $\alpha=0.5$, $n_{\text{runs}}=5$ pooled) takes about 18 H200-hours. Probe training and analysis (Ridge or logistic regression on cached hidden states, no GPU) takes about 2 CPU-days. Total reported compute: about 156 H200-hours. Including preliminary and exploratory runs not in the paper (intervention sweeps that did not enter, extra collection on archival models, method-development sweeps over layer sampling and calibration), the full project consumed roughly $3 \times$ the reported total, about 450-500 H200-hours on a shared internal cluster.

Probe training. Logistic regression (binary, boundary, 4-class) with lbfgs solver, $C = 1.0$, $\text{max_iter} = 5000$, 5-fold stratified cross-validation on L^2 -normalized features. Ridge regression with $\alpha = 1.0$. Operating threshold $\tau = 0.15$ chosen on validation folds to maximize F1.

Validity controls (full numbers). At L_8 : real AUROC 1.000, shuffled 0.297 (gap 0.703), within-step 0.588, step-index R^2 0.998. At L_{32} : step-index R^2 0.978. Top-10 feature dimensions at L_8 and L_{32} are disjoint (IoU = 0.0).

Graded probe per-class. 4-way classifier at L_{32} , 5-fold CV, 80/class. Precision/recall/F1: step+1 0.988/1.000/0.994; step+2 0.876/0.888/0.882; step+3 0.728/0.738/0.733; step+5 0.857/0.825/0.841. Errors are strictly monotonic on adjacent decay classes.

Cross-domain calibration. Isotonic regression fit on a 10-task HotpotQA calibration split (93 steps), evaluated on 20 held-out tasks (145 steps); splits fixed and disjoint from ALFWorld training.

Strict stripping (reasoning models). All complete `<think>...</think>` blocks are removed from each prior assistant turn in condition B. Content after `</think>` (Thought: and Action: lines) is kept so B sees the same external action-observation sequence as A. The protocol is a no-op (+4.8%, within noise) on Llama-3.1-70B, which emits no `<think>` content.

Intervention harness. The intervention harness uses ALFWorld in standard DQN evaluation mode and reports task success via `info["won"]`. The probe is queried at the last-token position of each step. A re-anchoring intervention fires when the predicted plan signal falls below $\tau = 0.10$. A worked example on `pick_heat_then_place`: the probe fires at step 5 ($0.041 < \tau$), the re-anchoring intervention is applied, and the trajectory ends with `won=True` at step 7.

Judge. Behavioral alignment scoring uses a single LLM judge (Claude Opus 4 family) at temperature 0; alignment threshold for “deviation” set at 0.3 on the $[0, 1]$ scale.

Cross-model scope (negative result). Archival hidden states from Phi-3-Medium and Qwen2.5-72B were collected in a single-run regime where step-level activations are bit-identical across re-runs of the same question. Plan signal is a paired contrast, so it is undefined on those data, and we make no claim about non-Llama architectures from existing corpora. The most informative next experiments are replay-paired collection on a mixture-of-experts model and on an RL-tuned reasoning model. Both need fresh data collection per architecture.

A.1 Layer Profile and Task-Type Decay

The layer profile and per-task-type decay curves are promoted to Figure 2 in the main text because they show the core robustness check: the peak is localized at L_{32} , and all six ALFWorld task types show nearly identical decay dynamics regardless of plan length.

A.2 Probe Validity Controls and Leakage Analysis

Because plan signal is by construction maximal at step +1 and minimal at later steps, a probe trained on this contrast risks learning “which step is this” rather than “how much plan is present.” Table 2 reports two controls: a shuffled-label control (breaks the real relationship while keeping the step structure) and a step-index regressor (predicts step index from the same hidden states). At L_8 , real binary AUROC is 1.000 but shuffled AUROC drops to 0.297^1 and the step-index regressor reaches $R^2 = 0.998$, so L_8 representations carry nearly all step-index information for a trivial classifier. At L_{32} the step-index R^2 is still high (0.978), but the regression target (plan-signal magnitude) has graded structure (0.453, 0.110, 0.060, 0.027 at steps +1, +2, +3, +5) that step index alone does not predict. The probe’s residual fit ($R^2 = 0.875$) is the non-trivial component.

A.3 Graded Probe Confusion Matrix

The 4-class probe ($\{+1, +2, +3, +5\}$) at L_{32} , trained with 5-fold CV on 80 examples per class, attains overall accuracy 0.863, macro $F_1 = 0.862$, macro AUROC 0.976 (Table 3). Confusion is monotonic: errors concentrate on adjacent steps, never on distant ones, consistent with a one-dimensional plan-presence quantity that the probe orders correctly.

¹Below 0.5 reflects small-sample 5-fold CV variance under shuffling; the relevant comparison is the drop from 1.000.

Table 2: Probe validity controls. Real: binary AUROC of step+1 vs. step \geq +3. Shuffled: same probe with labels permuted. Step R^2 : linear regressor predicting step index from hidden state.

Layer	Real AUROC	Shuffled AUROC	Δ	Step R^2
L_8	1.000	0.297	0.703	0.998
L_{32}	1.000	n/a	n/a	0.978

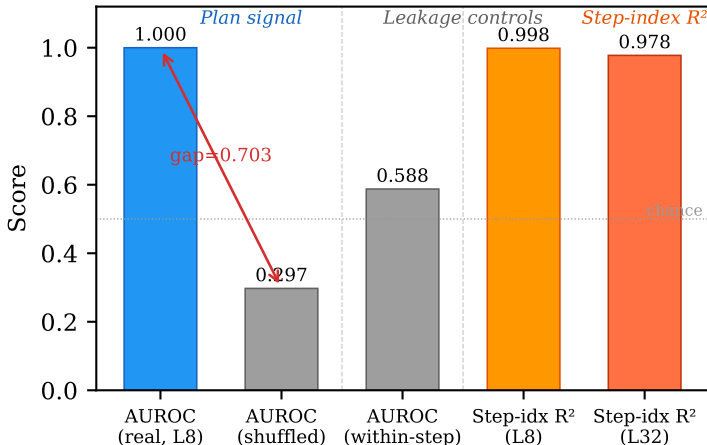


Figure 5: Step-index leakage at L_8 . Real binary AUROC (1.000) drops to chance (0.297) under label shuffling. The Ridge regression target at L_{32} (continuous plan-signal magnitude) sidesteps this confound.

A.4 Probe Calibration

Brier score and 10-bin Expected Calibration Error (ECE) at L_{32} are in Figure 6. The main binary probe (step +1 vs. \geq +3) is near-perfectly calibrated (Brier = 2.5×10^{-6} , ECE = 0.0002), which is unsurprising given the wide plan-signal gap between classes. The harder boundary probe (step +2 vs. +3, plan-signal gap 0.110 vs. 0.060) shows moderate miscalibration (ECE = 0.066), addressed via per-domain isotonic recalibration.

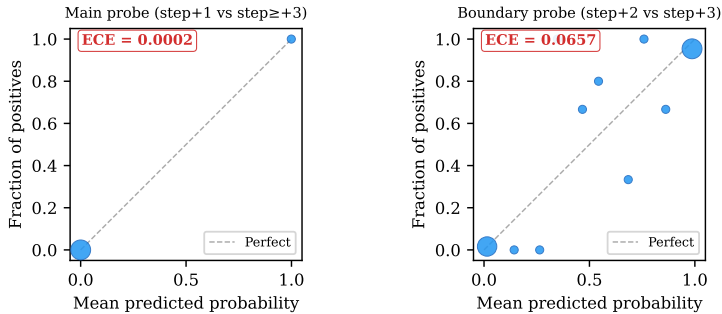


Figure 6: Reliability diagram for the L_{32} probe. Diagonal indicates perfect calibration.

A.5 Strict-Strip Per-Task Recovery

Per-task strict-strip recovery on R1-Distill-Llama-70B is in Table 4. Every task shows positive recovery; the cross-task mean is +0.036 (+163%). Five held-out tasks covering two task types not in the in-sample set yield +153%.

Table 3: 4-way graded probe at L_{32} (5-fold CV, 80 examples/class). Rows are true classes, columns predicted.

True ↓ Pred →	+1	+2	+3	+5	F_1
+1	80	0	0	0	0.994
+2	0	71	9	0	0.882
+3	0	10	59	11	0.733
+5	1	0	13	66	0.841

Table 4: Strict-strip recovery on R1-Distill-Llama-70B (5 in-sample ALFWorld tasks).

Task (type)	Standard	Strict-strip	Δ
task_000 (pick_heat)	0.029	0.067	+0.038
task_002 (pick_two_obj)	0.020	0.062	+0.042
task_003 (pick_clean)	0.020	0.053	+0.033
task_004 (pick_and_place)	0.018	0.054	+0.036
task_006 (pick_and_place)	0.023	0.055	+0.032
Mean	0.022	0.058	+0.036

A.6 Reasoning-Model Per-Layer Plan Signal (Dilution Test)

Table 5: Cross-regime probe transfer (referenced in §7). The Llama-trained L_{32} probe is at ceiling on Llama, drops on R1 raw, and partially recovers under strict stripping ($p=6 \times 10^{-4}$, $n=19$). Self-probes reach ceiling on both R1 distillations. Qwen3-32B-native (enable_thinking=True) shows linear encoding ($R^2=0.997$) without spike-decay locality (binary AUROC 0.616). The R1 self-probe is at 89.3° from Llama’s L_{32} (cosine 0.012); R1-Qwen = R1-Distill-Qwen-32B.

Probe	Eval target	AUROC	n
Llama L_{32}	Llama ALFWorld	0.999	80
Llama L_{32}	Llama HotpotQA	1.000	25
Llama L_{32}	R1 raw (transfer)	0.600	5
Llama L_{32}	R1 strict (transfer)	0.748 [0.62,0.87]	19
R1 L_{32} (self)	R1 strict	1.000	85
R1-Qwen L_{32} (self)	R1-Qwen strict	1.000	85
Qwen3 L_{26} (self)	Qwen3-think [†]	0.616*	15

[†]enable_thinking=True, 270 step pairs. *Regression $R^2=0.997$.

To test the dilution hypothesis (that a reasoning model spreads representation more thinly across layers, predicting a uniform R1/Llama signal ratio across depth), we report per-layer plan signal at step +1 on the 5-task ALFWorld subset in Table 6. The R1/Llama ratio varies $4.4\times$ across depth and the two models peak at different absolute layers (L_{72} for R1, L_{32} for Llama). Strong-form dilution is ruled out.

A.7 Cross-Scale Multi-Model Panel

Table 7 collects the cross-model decay results referenced in Section 6; Figure 7 summarizes them graphically.

A.8 Intervention Sweeps and Head-Level Analysis

Context-level re-anchoring. On 30 held-out ALFWorld tasks ($\text{max_steps} = 200$), the control, probe-triggered, and oracle always-replan conditions all land within ± 1 task of one another. This is a clean 30-task null, which is what the in-context retrieval account predicts.

Table 6: Per-layer step+1 plan signal (5 ALFWorld tasks). The R1/Llama ratio is far from uniform; peaks differ.

Layer	R1-Distill-70B	Llama-70B	Ratio
L_8	0.002	0.031	0.07
L_{32}	0.023	0.184	0.12
L_{40}	0.026	0.129	0.20
L_{48}	0.030	0.113	0.27
L_{56}	0.032	0.096	0.33
L_{72}	0.039	0.070	0.56
L_{80}	0.022	0.044	0.50

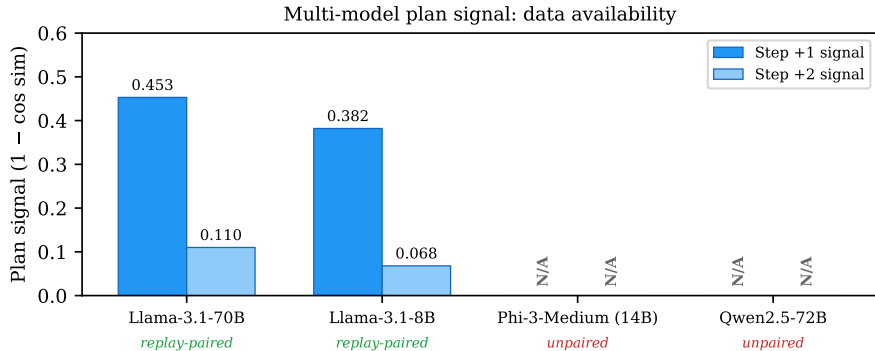


Figure 7: Multi-model panel. Llama-3.1-70B: full replay-paired decay curve, peak L_{32} , decay $4.1\times$. Llama-3.1-8B: same shape at smaller scale (peak 0.382, $10.9\times$ decay) at matched relative depth around 33%. Phi-3 and Qwen2.5: plan signal not computable from archival data (non-replay-paired).

Residual-stream steering ($n_{\text{runs}}=5$ paired, $n=150$ pooled). We sweep at L_{32} with $\alpha \in \{0.5, 1.0, 2.0\}$ on 30 held-out ALFWorld tasks \times 5 paired runs. At $\alpha=0.5$ the pooled paired contrast is a clean directional null: $\Delta = +1.33$ pp (SR 54.7% \rightarrow 56.0%; paired Wilcoxon $p=0.97$; McNemar 11/12/7 help/hurt/tie, $p=1.00$; 95% bootstrap CI $[-8.0, +11.3]$). Within-replicate Δ s straddle zero: $+8.9$ pp at $n_{\text{runs}}=3$, -10.0 pp at the new $n_{\text{runs}}=2$ replicate. Pooling resolves them. $\alpha=1.0$ and $\alpha=2.0$ both yield -3.3 pp drops below baseline. Uniform residual addition along the L_{32} plan direction is not a behavioral knob at this scale, which is the directional null the population-coded view predicts: a single-axis push along a measurement of the plan-presence contrast does not function as an intervention.

Head-level structure. A deterministic head-level analysis shows the plan direction is distributed across the 64 heads at L_{32} with mixed-sign top-3: top-3 captures 23% of $\sum_h |c_h|$, top-10 captures 55%. A head-aware intervention that respects the signs of c_h ($n=90$ paired, $\beta=1.0$) gives $+3.3$ pp at effective magnitude 0.094 (directional but within noise). At $\beta=2.0$ (effective magnitude 0.188) it flips to -4.4 pp. Scaling head-aware to match $\alpha=0.5$'s effective magnitude ($\beta \approx 5.3$) destabilizes generation: the model stops emitting termination tokens within the LLM-call budget. Uniform addition at the same effective magnitude ran cleanly but landed at the $n=150$ null above. Both intervention shapes (uniform at matched magnitude, head-aware at low magnitude) fail to reliably move task success. The asymmetry is what isolates the population-coded contrast as a measurement of which heads carry productive vs. antagonistic plan signal, rather than as a knob that can be scaled arbitrarily.

A.9 Why Does ALFWorld Decay Slower than HotpotQA?

The $3\times$ gap between ALFWorld decay ($4.1\times$) and HotpotQA decay ($12.4\times$), given the near-identical initial signal (0.453 vs. 0.445), has a plausible structural explanation in the observations themselves. ALFWorld observations (“You arrive at countertop 1. On the

Table 7: Cross-model plan signal. Llama-3.1-70B: 80 ALFWorld tasks. Llama-3.1-8B: 20 tasks. Phi-3 / Qwen2.5: not replay-paired (archival-only data; plan signal undefined).

Model	Step+1	Residual	Decay	Peak layer
Llama-3.1-70B	0.453	~ 0.027	$4.1\times$	L_{32} (40%)
Llama-3.1-8B	0.382	~ 0.035	$10.9\times$	$L_{12/32}$ ($\sim 37\%$)
Phi-3-Medium		<i>N/A (not replay-paired)</i>		
Qwen2.5-72B		<i>N/A (not replay-paired)</i>		

countertop 1, you see a tomato 1.”) are directly relevant to the plan (“go to countertop, pick up tomato, heat it”), so plan-relevant features stay in the residual stream across steps. HotpotQA observations (Wikipedia passages about unrelated entities) are informationally orthogonal to the plan (“search for X , retrieve Y ”), which allows faster representational overwrite. If this account is right, plan-decay rate should correlate with the semantic overlap between plan content and observation content. This is a testable hypothesis for future work.

B Phase 1 Cross-Condition Plan-Signal Post-Processing

The Phase 1 cross-model sweep (Section 7) uses an independent-trajectory architecture. Each of the six conditions (r1_raw, r1_strict, qwen3_think_raw, qwen3_think_strict, llama_70b, qwen3_nothink) runs its own forward agent loop on each task ($n_{\text{runs}} = 3$ for reasoning conditions, $n_{\text{runs}} = 1$ for non-reasoning baselines). The behavioral task-success results are in Table 8. The 5+5 paired-trajectory replay-pair semantics from Section 7, in which condition B replays the actions and observations of condition A token-for-token, do not port directly to Phase 1, because no two Phase 1 conditions are paired by construction.

Table 8: Phase 1 behavioral task success ($n=25$ ALFWorld tasks). Strict stripping moves Qwen3-thinking from 0% to 8%, matching its no-thinking baseline, but does not move R1-Distill-Llama-70B.

Condition	Solved	Task SR
<i>Non-reasoning baselines</i>		
llama_70b	3/25	12.0%
qwen3_nothink	2/25	8.0%
<i>Qwen3-thinking family</i>		
qwen3_think_raw	0/25	0.0%
qwen3_think_strict	2/25	8.0%
<i>R1-Distill-Llama family</i>		
r1_raw	0/25	0.0%
r1_strict	0/25	0.0%

To compute strict-strip recovery on Phase 1 data we use a cross-condition post-processor. For each (task, r, t) with matching run index r and step t , we load the saved hidden-state arrays from the matched raw and strict directories (r1_raw/task_NNN/run_RR/hs_step_TT.npy vs. r1_strict/task_NNN/run_RR/hs_step_TT.npy) and compute per-layer cosine distance:

$$\text{PlanSignal}_1^{(\text{task}, r, t, \ell)} = 1 - \cos\left(\mathbf{h}_{(\text{task}, r, t, \ell)}^{\text{raw}}, \mathbf{h}_{(\text{task}, r, t, \ell)}^{\text{strict}}\right). \quad (2)$$

This measures the same conceptual quantity as the paired-trajectory plan signal (the representational change attributable to whether the contamination correction is applied), at the cost that the two trajectories are not action-replay-paired. We restrict comparisons to (task, r, t) cells where both the raw and strict trajectories executed the same step count without environment truncation, and report cross-task mean signal only over those cells. Within-model strict-vs-raw recovery is evaluated with paired permutation tests across the matched per-task means. The paired-trajectory +163% headline number is unaffected by this post-processor and remains the principal evidence for the methodological claim.

Behavioral results do not depend on this post-processor. The Phase 1 task-success table (Table 8) uses only $\max_t \text{score}(t) > 0$ counts per condition, which are computed directly from the saved trajectory results without any cross-condition pairing. The per-run success flag in our raw logs is decoupled from goal completion (it triggers on step-limit truncation), which is why we explicitly redefine real success as $\max\text{-score} > 0$ throughout Section 7.

C Context-Compression Stress Test Details

We evaluate four context policies on 30 ALFWorld tasks (task indices 50-79), 5 runs each (150 runs/policy), with Llama-3.1-70B as a ReAct agent under a 20-step cap. All policies share the budget `keep_recent=4` retained messages plus the system prompt; the agent’s full message list is preserved internally, but only the compressed view is sent to the model on each call. The probe used by `probe_gated` is the L_{32} Ridge probe of §5, firing at threshold $\tau=0.15$; on a fire it re-injects the pinned plan span for that step. Per-policy success is $\max_t \text{score}(t) > 0$. Comparisons use a paired permutation test (10,000 permutations) with 95% bootstrap CIs over the 30 tasks; all results integrity-checked for 30 tasks \times 4 policies \times 5 runs with no errored or zero-step runs (Table 9).

Table 9: Context-compression stress test (150 runs/policy). Δ and p are paired permutation tests vs. the noted baseline; CIs are bootstrap 95%. Naive eviction is catastrophic ($p < 0.001$); neither plan-aware policy recovers (both CIs cross zero).

Policy	Success	SE	Δ (ref)	p	Resurf.
none	56.7%	4.0%	n/a	n/a	0.0
naive	22.0%	3.4%	-34.7 (none) [-44.7, -24.7]	< 0.001	0.0
plan_protected	20.7%	3.3%	-1.3 (naive) [-10.7, +8.0]	0.89	0.0
probe_gated	24.7%	3.5%	+2.7 (naive) [-6.7, +12.0]	0.67	6.1