READING THE BOARD, PLAYING ANOTHER ONE: TURN-LEVEL EVALUATION OF OPEN LLM AGENTS ON TEXTUAL SOKOBAN

Anonymous authors
Paper under double-blind review

ABSTRACT

Aggregate outcomes and no-op rates obscure *where* an interactive agent fails. We report a turn-level evaluation of five open 3–8B LLM agents on textual Sokoban: 180 episodes and 1,561 steps across six conditions, where no level is solved and no agent docks a box. Two further experiments bound instrument failure and level difficulty. Replaying push-optimal solutions through the harness, repaired to log the solving step, solves all six corpus levels. On a fixed ladder of one to six optimal pushes, five models from three families solve 0 of the 60 episodes at or above four pushes; on the three lower rungs, success is slightly below a pre-registered uniform-random baseline at one push and above it at two and three. With one board per rung, this bounds the fixed ladder rather than establishing a general complexity threshold. We code every corpus turn against a ten-family behavioural taxonomy induced from three blind reading passes, with a contrast set recording locally correct elements of a turn. In 14 episodes of one model across the three reading passes and both temperatures, an accurate board table appears *in the model's own output* beside prose that reasons from a different board; the clearest instance recurs in all three executions. The move a thought describes and the action it emits also dissociate: a plain move described, a push emitted, or the reverse. On ladder rungs 1–3, a pre-registered blind control of 291 shuffled single turns finds no separation between solved and unsolved episodes on the pre-specified whole-turn contrast item. We therefore present the taxonomy as a description of observed behaviour in this setting, not as a validated discriminator of competent and incompetent turns. A further observation is carried as a metric argument, not a frequency claim: the most irreversibly lost levels are lost to *legal, successful* pushes, so a count of ineffective actions ranks the most damaging episodes as the best. Both annotators are language models, so we report κ beside PABAK and prevalence and separately assess sampled label defensibility with 316 human verdicts across three instruments, never pooled. A third language-model annotator over the whole corpus has a substantially lower median in-band κ with either of the first two (0.319 and 0.381) than they have with each other (0.576), each median over its own item band. Re-running an identical temperature-0 configuration three times, only 14 of 30 cells take the same path: fixed-seed greedy decoding did not reproduce trajectories in this serving configuration.

1 INTRODUCTION

Large language models are increasingly deployed as agents: they read a state, choose an action, observe a result, and repeat. Games are a popular testbed for that loop, and a growing literature reports how well such agents do; it rarely diagnoses why an agent's decision on a given turn was wrong. Outcome metrics collapse many distinct failure processes into one number, and can conflate model ability with artefacts of the surrounding scaffold (Chen et al., 2026a).

We take the turn as the unit of analysis. For every step we hold what the agent was shown, what it said it was thinking, the action string it emitted, and the board that resulted, so a mistaken belief is legible against the state that contradicts it, at the step where it was formed. Our testbed is Sokoban: a box-pushing puzzle, PSPACE-complete in general (Culberson, 1999; Dor & Zwick, 1999), that is

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.

unforgiving (one bad push can make a level permanently unwinnable) and that exposes a complete state every turn, which we hand the agent as text rather than pixels so that visual perception leaves the loop entirely. We study fixed, publicly available open checkpoints (3–8B) that fail often enough to be legible. Since they get nowhere in all six conditions, how far they get is a short question; we ask instead where in the turn the error enters, and what an evaluation has to record to see it.

Contributions.

1. A turn-level evaluation instrument, and the narrowed finding it yields. A behavioural failure taxonomy in ten families, induced bottom-up from three blind reading passes over the traces, with a contrast set of things the agent gets right coded as carefully as the failures, because a step carrying no contrast label is not the same as a step nobody looked at. Applied turn by turn, it shows that aggregate outcomes and no-op rates hide distinct failure processes: across temperatures and independent executions, some turns of one model re-emit an accurate board in its own output beside prose reasoning from another, some describe one move and emit another, and legally successful actions can be more damaging than ineffective ones (an argument about metrics, §6). It rests on a conjunction over eight named items, so it can be tested by removing them one at a time (§5.3).
2. Agreement between two model annotators is not validity, so validity is assessed separately. Both annotators are language models, so we report κ with PABAK and prevalence, gate only the items whose prevalence admits it, and separately assess sampled label defensibility with 316 human label-verdicts across three instruments that are never pooled, the operator dependency disclosed on both registered sheets. Blindness to the study's direction was tested, and failed the first test.
3. Fixed-seed temperature-0 decoding did not yield trajectory reproducibility in this vLLM/GamingAgent configuration. Re-running an identical temperature-0 configuration on the same GPU and serving build, only 14 of 30 cells take the same path across three executions (1 of 30 at temperature 1), so studies of this kind should report behavioural statistics over episodes rather than individual trajectories.

2 Related work

The nearest neighbour on the game is SokoBench (Monti et al., 2026): a capability benchmark on large reasoning models, open and closed, with puzzles simplified to isolate planning from state persistence. It is the same game asked a different question; we describe what individual turns do wrong rather than how far an agent plans. The nearest neighbour on the method is *Why Reasoning Fails to Plan* (Wang et al., 2026b), which formalises chain-of-thought as a step-wise greedy policy and identifies long-horizon collapse, early myopia and irreversible errors, on graph traversal and tool use rather than a spatial game.

Failure analysis is an active area, but not in games. Process-level failure work is coding, SWE, CLI, tool-use or multi-agent (Zhao et al., 2026; Albayaydh et al., 2026; Wang et al., 2026a; Zhu et al., 2025); none targets games or puzzles. The grid and spatial cluster, conversely, is benchmarks rather than failure analysis (Hu et al., 2025; Paglieri et al., 2024; Tang & Kejriwal, 2024); internal-mechanism work (Han et al., 2026; Tan et al., 2026) diagnoses through activations and patching (Appendix G). Our position: open small models $\times$ textual Sokoban $\times$ a behavioural turn-level failure taxonomy built bottom-up from blind reading passes $\times$ explicit harness-confound control.

3 Experimental setup

Game, levels and termination. Sokoban on the harness's six fixed predefined levels. An episode ends on success or when the board is unchanged for five consecutive steps. Level 6 starts with one box already docked, and no docking figure here counts it. The six derived initial boards are md5-identical across all six conditions.

Observation and harness. Text mode. The board is serialised as an `ID | Item Type | Position` table, one row per cell, and every coordinate in this paper is the agent's `(x, y)` resolved against that printed table. We run `lmgame-org/GamingAgent` (Hu et al., 2025) at

Table 1: The corpus and its outcomes, per condition. Each condition is 30 episodes (5 models $\times$ 6 levels); the corpus is 180 episodes and 1,561 steps. No agent solves any level in any condition and no agent ever docks a box. No-op rates are reported per condition and are never pooled: they range 77–86%. The *Unknown* column is the renderer check, zero in all 1,561 steps. Source: `gates_all.py`, all gates pass. Executions ship as `passN_tempT`.

condition	episodes	steps	solved	frozen	no-op rate	*Unknown*
exec. 1, T=0	30	241	0	30	85%	0
exec. 1, T=1	30	292	0	30	77%	0
exec. 2, T=0	30	238	0	30	86%	0
exec. 2, T=1	30	246	0	30	84%	0
exec. 3, T=0	30	257	0	30	80%	0
exec. 3, T=1	30	287	0	30	80%	0
total	**180**	**1,561**	**0**	**180**	77–86%	**0**

upstream commit `996d848` with `harness: true`, executing the full per-step loop (perception, memory and reflection, reasoning) as separate model calls. In text mode the perception module makes no API call, so its prompt never reaches the model, and the memory window is effectively five turns rather than ten. Harness state is recorded per condition and verified by a gate on the prompt hash; Appendix D lists which harness defects were fixed before collection and which were deliberately left in place.

Models and serving. Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, Llama-3.2-3B-Instruct, Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3, all fp16; precision is not a variable here. Self-hosted vLLM behind an OpenAI-compatible endpoint on a single NVIDIA L4, fixed seed, no paid API.

The corpus. 180 episodes = 5 models $\times$ 6 levels $\times$ 3 replicates $\times$ 2 temperatures, 1,561 steps (Table 1). The three replicates come from three separate executions, and each execution produces its temperature-0 and temperature-1 episodes in one session. That pairing is deliberate: the decoding nondeterminism of §5.5 lives across sessions, so pairing inside one execution holds session constant for the temperature comparison while the three executions supply the replicates. Decoding conditions are never pooled into a behavioural rate: a paired comparison between them is legitimate, a single frequency computed over both is not. Two displays do report across conditions, by design: the ladder counts outcomes rather than behaviour (Table 2), and Appendix H's scaffold-exposure and reflection-channel bounds measure the harness, not the agent under a decoding condition.

Episodes are short: median 6 steps, minimum 6, maximum 26, and 95 of 180 episodes (52.8%) are exactly 6 steps long. That floor is the five-step stuck detector, so in a six-step episode the board never changed after the first action. That bounds what a turn-level analysis can see: the findings are overwhelmingly about the opening of an episode (§7).

4 Turn-level evaluation method

For each step we hold the board the agent was shown, the reasoning it stated, the action string it emitted and the board that resulted, with the coded layers attaching to the middle two (Figure 2, Appendix A). A step-N thought is judged against the step-$(N-1)$ board, because the logged observation is the *post*-action one.

The taxonomy was induced blind, in three reading passes. Each read one execution's 60 episodes, no episode read twice, so the three cover all 180 exactly once. Every reading pass spans the same 60 model $\times$ level $\times$ temperature cells, so reading passes 2 and 3 are replicates rather than new material. Each was read by ten empty-context readers (one per model $\times$ temperature workbook of six episodes; thirty across the study), each given that one file and a fixed brief: no checklist, no rubric, no category codes, no other reading pass's findings. Readers describe what they see; matching a description to a taxonomy item is a separate written step afterwards, made in a language-model session under author direction. `NEW` is the default whenever a fit is arguable, because absorbing a novel form into an existing item silently manufactures saturation while over-splitting is recoverable by merging. Every step-level finding carries a verbatim span checked mechanically against the thought of its own

step; spans failing that check were discarded, and the episodes containing every span quoted in this paper ship with the supplementary material. The three reading passes produced 1,552 step lines and 2,194 quoted spans; Appendix C holds the audit trail, with a second saturation curve the first one hides.

The codebook. Ten families in seven layers, organised by where in the turn the claim goes wrong rather than by severity: the board as stated (A); progress that did not happen (B); the already-satisfied goal (C); the move as planned (D); the action as emitted (E); whether the next step registers what happened (F); target commitment (G); where the damage comes from (H); the contrast set of what is *not* wrong (I); and the generation layer (J). That is 97 codeable items of a 99-item inventory, an over-split upper bound by construction; its arithmetic is in Appendix A, the generation layer is counted separately (§5.4), and the full codebook is Appendix B.

Both annotators are language models, and this bounds the reliability claim. All 1,561 steps were independently double-coded by two LLM annotators applying the same codebook, fixed before coding, each blind to the other's labels and to this draft: annotator A returned 10,996 label rows over 180 of 180 episodes, annotator B 7,100 over the same 180. Annotator A is Claude Opus 5 and annotator B is GPT-5 via the Codex coding agent; they are from different families, as the codebook's protocol requires. The three blind induction passes that produced the codebook were read by Claude Opus 5, so annotator A shares a model with the readers and is not naive to the taxonomy's framing. No human independently coded a complete episode. We tested blindness to the study's direction, and it failed the first test: seven of eight fresh empty-context readers could state the direction from the second annotator's materials alone. The carrier prose was removed, the definitions left unchanged, and no re-tested reader could; one residue is irreducible, which is why we treat κ as a measure of codebook specificity and the human label assessment as the pipeline's only evidence of label defensibility. Appendix H gives versions, access dates, harnesses and context isolation for all three roles.

κ has a trap here, and it is the opposite of the usual one. There are no near-universal items: the most prevalent of the 97 reaches 52.2% of steps and 72 of 97 fall below 10%, so chance agreement sits near ceiling and κ (Cohen, 1960) collapses toward zero even at very high observed agreement, the prevalence paradox (Feinstein & Cicchetti, 1990). We therefore apply the $\kappa \geq 0.6$ gate only where prevalence, taken as the mean of the two annotators', lies between 10% and 90% in the item's own unit. That admits 25 items: per step ($n = 1{,}561$) for 17 of the 25, per episode ($n = 180$) for 7, per run ($n = 878$) for 1, a run being a maximal stretch of identical consecutive actions. For the other 72 we report n, agreement, both prevalences and PABAK (Byrt et al., 1993) beside κ. 9 of the 25 pass: median in-band κ 0.576 (range 0.133–0.918), median observed agreement across all 97 items 97.9% (Appendix E).

A third annotator read the whole corpus, and its κ with either original annotator is substantially lower than theirs with each other. Gemini 3.7 Flash labelled all 180 episodes under the same frozen codebook, differing from annotators A and B on four axes at once: model family, harness, labelling prompt and context isolation. Only annotator A ran the project's own fixed prompt, so an operator-written prompt is what annotator B and the third annotator share rather than a defect peculiar to the third. Median in-band κ is 0.319 against A and 0.381 against B beside 0.576 for the reported pair, each over its pairing's own band (20, 18 and 25 items) and never pooled. I2, one of the two items §5.3 rests on, reads κ 0.259 for annotators A–B and 0.477 for A–third; it lies outside the prevalence band in every pairing, so neither is read as evidence. Appendix E carries the pairings, the two silent-drop defects those sheets exposed (one of which moved a conclusion) and our single use of the blind rule review's target list: a sanity check, not independent confirmation.

Human label assessment is a separate instrument. Two language models can converge on the same wrong label, so people reading the turn assessed whether sampled labels are defensible under the frozen codebook; three instruments returned 316 label-verdicts and are never pooled into a rate, having different checkers, different draws and opposite independence properties: two registered 40-turn draws (143 and 147 verdicts) and one purposive probe drawn agreed-only (26). The stratum that matters is where a shared blind spot would live. Of the 225 labels both annotators independently applied, three were judged indefensible: 3 of the 91 agreed labels on the first registered sheet, 0 of 108 on the second, 0 of 26 in the probe. Where only one annotator applied the label, 17 of 91 were judged wrong. That 91 is a *different* set: it sums the two registered sheets' single-annotator strata

(52 and 39), and is a count, never a rate. Each checker had operated one of the two annotators, in opposite directions (§7).

5 RESULTS

Per-item episode counts come from the blind reading passes of §4 (60 episodes each, ten empty-context readers per pass, no checklist), not from the two annotations. A figure written "at reading pass 3" is n of 60; "across the three reading passes" is n of 180. These counts support presence, breadth and relative magnitude. They do not support a rate, a prevalence, or a comparison turning on one or two episodes. The reading passes generate the descriptive counts, the two full-corpus model annotations assess codebook reproducibility, and the human label assessment judges sampled label defensibility: three separate evidence streams.

5.1 OUTCOMES ARE IDENTICAL IN ALL SIX CONDITIONS

Zero levels solved, no box docked by any agent, 180 of 180 episodes frozen at the final step, every step returning the step penalty. These hold identically across all six conditions (Table 1), verified by a gate over the logged traces. The no-op rate, the fraction of actions leaving board and worker unchanged, runs 77–86% per condition and is not pooled.

Perseveration needs its criterion stated. Under the primary measure `m3` (three or more identical consecutive actions, every one of them a no-op), perseveration occurs in 67 of 90 episodes at temperature 0 and 50 of 90 at temperature 1 (90 episodes per temperature). Six plausible variants span 50 to 90 of 90 on identical data, so a bare "perseveration in n/N" is uninterpretable here and every figure we report names its measure. A criterion keyed to near-verbatim repetition of the agent's *reasoning text* collapses between the two temperatures where the action-based criteria do not: two near-verbatim consecutive thoughts occur in 73 of 90 episodes at temperature 0 and 17 of 90 at temperature 1, two identical consecutive actions in 89 and 81. A text-based criterion measures the decoder rather than the agent.

5.2 A FAILURE THRESHOLD ON A FIXED LADDER, AND THE HARNESS CHECK

Zero solved in every condition admits three readings, which the corpus cannot separate: the agents cannot play, these six levels are too hard for any signal to appear, or the harness could not record a solve. Two further arms address the last two. Neither enters the annotated corpus and neither is pooled with the 180.

First the harness. A push-optimal solution from a solver sharing no code with the harness, replayed through the *repaired* harness (the ladder's configuration, which differs from the corpus's only in logging the step that solves a level), solves all six levels over 229 replayed steps, the solving step present in the record. The corpus's configuration would have missed five of the six: it resets into the next level before logging that step. Without relying on that signal, the recorded boards show the agents never came near such a win: the greatest number of boxes docked anywhere in the 1,561 steps is 1, and that one ships already docked, so no multi-box level was ever one push from completion, and on the one-box level the box never came closer than five cells to its dock (Appendix D).

Then the difficulty. We enumerated a ladder exhaustively on a single 5×5 open room and took the first placement in canonical row-major order at each optimal push count from 1 to 6; the rule was fixed before any episode ran, names no model and no outcome, and uses no seed and no generator. The same five models, three replicates at temperature 1 and one at temperature 0, give 120 episodes (Table 2).

Rungs 1–3 are pre-registered calibration and are never reported as a bare solve rate. At rung 1 the agents solve 8 of 20, 40.0% against a uniform-random baseline of 41.5%, slightly below chance. Rungs 2 and 3 sit above theirs, 5 of 20 (25.0%) against 16.2% and 7 of 20 (35.0%) against 6.1%; pooled exact one-sided binomial calculations give rung 2, $p = 0.215$, and rung 3, $p = 0.000117$, descriptive only, as each pools five models and two decoding conditions; that calculation concerns one board, not three-push puzzles. All five models solve rung 3 at least once. The result is the table's other half: 0 of the 60 episodes at or above four optimal pushes, sixty consecutive failures across five models and three families, where the baseline is under 0.15%. These are twenty repeated

Table 2: The difficulty ladder. 120 episodes: five models × six rungs × three replicates at temperature 1 and one at temperature 0, so 20 per rung. Solves are read off the board, never off the termination flag, which the stuck detector also sets and which is present on all 120. The baseline is the pre-registered probability that a uniform-random agent over the same nine action strings solves that rung inside the same step budget (Appendix D).

rung (optimal pushes)	solved	of	uniform-random baseline	
1	8	20	41.5%	calibration
2	5	20	16.2%	calibration
3	7	20	6.1%	calibration
4	**0**	20	0.14%	
5	**0**	20	0.08%	
6	**0**	20	0.06%	

executions of each of three boards, not sixty independent puzzles: one board does not represent its push depth. On this fixed ladder, success disappears after the three-push rung; that is the one claim here that needs no power.

Scale does not move it. Llama-3.1-8B and Qwen2.5-3B each solve 6 of 24; Qwen2.5-7B solves 3 of 24, half what the 3B of its own family manages; Llama-3.2-3B 3 of 24 and Mistral-7B 2 of 24. Per-model rows are the reportable unit here and the pooled column describes no model.

What the arm's outcomes cannot do is test the taxonomy: its 20 solves sit in the calibration band, where a random agent also succeeds, and the pre-registered blind control on them is null (§7).

5.3 The finding, and why it is a conjunction

Two observations recur across temperatures and independent executions: some turns of one model re-emit an accurate board table beside prose that reasons from a different board, and some describe one move and emit another. That is the paper's empirical claim. We do not establish that the two fall on the same turn (both annotators agree on only one such turn) or that the thought causes the action. The claim rests on a conjunction of findings, whose accounting and the effect of removing any one are set out in Appendix E.

The weight rests on `I2` and `E2`. `E2` is the one conjunction item to pass the κ gate (0.749 at 25.18% mean prevalence, 12 of 12 sampled labels confirmed), and the one the prompt itself demonstrates: its only worked example, sent on every step, pairs a plain-move thought with a push action (§7). `I2`, at 3.07%, sits below the band, so we read it on observed agreement (95.52%) and PABAK (0.910) and make no directional claim from its κ of 0.259, which the prevalence paradox predicts there. Both items are scored per step, so neither figure is comparable with an episode-level one, and both are reported as existence and breadth, never as frequency. The remaining items corroborate without carrying weight, and one leaves the accounting outright.

A further observation, that legally successful actions can be more damaging than ineffective ones, is carried as a metric argument (§6) and not as a frequency claim. Its only frequency evidence is `H1`, with `H9` strengthening the same observation rather than supplying an independent one, and `H1` is rare: 0.70% and 0.64% of the 1,561 steps for the two annotators respectively (mean 0.67%), κ 0.664, outside the gate band. Its episodes are the existence proof that the case arises in practice.

Every item in the conjunction appears in at least one human verdict, though coverage is highly uneven, and no sampled label on any of them was judged indefensible: 79 sampled verdicts, 70 confirmed, 9 unclear, 0 wrong, broken out per item in Table 4 (Appendix E), verdict counts and never a validity rate. 53 come from two registered draws, which reached `I1`, `E2`, `I5`, `E3` and `I2` (1 verdict, confirmed: an existence check that cannot estimate `I2`'s precision or recall); the other 26 from a purposive probe naming the seven items neither draw could reach, which says nothing about `H1`'s prevalence, only that the four labels a human read were defensible.

5.4 CORRECT STATE IS PRESENT, AND IS NOT USED CONSISTENTLY

Under I2, an accurate board table appears inside the thought while the prose beside it reasons from a different board: 14 episodes across the three reading passes (3 / 5 / 6), both temperatures, all of them Llama-3.2-3B. Four more episodes were read this way and do not count: in three the only board table is an echo of the harness log (J1, Appendix F), and in one there is no table at all. It carries the first leg because the correct board is in the model's own output, in the same message that contradicts it. Re-emission under headings mirroring the prompt's own structure is not evidence of comprehension, and the claim needs none: presence in the output rules out the narrow explanation that the board information was unavailable. In `t0 llama3.2-3b L1` the thought re-emits the board under a heading of its own making, listing `17 | Empty | (2, 2)` correctly, while the prose below states that "the box remains blocked by the wall at (2, 2)" and plans around that wall, at three steps in each of the three executions (per-execution detail: Appendix F). Because 19 of the 106 steps carrying an in-thought board table echo the harness's log, those steps are excluded from this item and counted as J1 (§5.4). The claim arrives from a second direction: in every episode where a box is called a wall (A6), the same message's board table lists that cell correctly as a box.

I1, I5 and I6 are three distinct forms of accurate reading followed by wrong inference, each present in both replicates: I1 in 13 episodes at reading pass 3 (60 episodes) and 7 at reading pass 2, I6 in 5 and 1, and I5 at a count we do not report (below). The passes are replicate conditions read by different readers, and §5.5 shows replicates take different paths, so we read these as independent recurrence, never as growth across reading passes, a comparison this design cannot support. I1 is the shape in its purest form, correct positions turned into an impossible move, and I6 is stronger still: a true and decisive fact is stated and the opposite conclusion drawn from it. Both worked cases are in Appendix F.

I5 and B5 carry a unit defect (§7), so we report neither's episode count; no I5 label was judged wrong (Appendix E). B5 is the I2 shape in one sentence, and Figure 1 is the whole of it; a caveat that it might be an artefact of the input representation is withdrawn on mechanical verification (Appendix F).

Detection is not the missing piece. F4 and F10 are one decision point with two answers: a shallow direction swap, and a specifically correct self-diagnosis whose corrective is equally impossible. Both sit outside the conjunction, kept as evidence (Appendix F).

The plan and the emitted action dissociate. E3 is the shape inverted: the thought names the right box, the right direction, sometimes the right standing cell, and the emitted action does something else. We record it separately because the content of the plan is not in error; it is a contrast item wearing a failure label, and its worked case is in Appendix F. E2, in which a plain move is described and a push emitted, or the reverse, occurs in 11 episodes at reading pass 3 (60 episodes) and is not cosmetic: push and plain move fail under opposite conditions here and neither falls back to the other, so E2 is the dissociation present at the step where the no-op occurs.

A successful action can be more damaging than an ineffective one. H1 is a legal, board-changing action that walks a box to a cell from which it can never be recovered: 12 episodes at reading pass 3 (60 episodes, four models) and 4 at reading pass 2 (recurrence in replicates, not growth, under the same caveat as the I-family counts above), both temperatures; the two passes together reach all five models. In `t1 qwen2.5-3b L6` the episode's only board-changing action is the one that ruins it; the remaining cases are in Appendix F. H9 codes the stronger version, an unrecoverable state created while the same thought asserts there is no deadlock, in 7 of 180 episodes, both of its checked labels confirmed. Six of the seven end on one cell, twice with the losing board read back correctly (Figure 3).

The generation layer, counted separately. Family J is a layer rather than a mechanism. It holds failures of staying inside the response format rather than of reasoning about the board, counted apart from everything above as six forms, itemised in Appendix F. Folding them into the reasoning counts would inflate the central claim with defects a stricter parser would remove; dropping them would hide something bearing directly on evaluation validity. A benchmark that scores a format failure as "the agent chose a bad move" is measuring the wrong thing, and an aggregate score cannot tell the two apart while a turn-by-turn method can.

What the agent was shown

```
18 | Worker on Dock | (3, 2)
19 | Box | (4, 2)
26 | Box | (4, 3)
32 | Dock | (3, 4)
```

with `boxes_on_target = 0`.

What the agent said, same turn

"The worker is currently on the dock at (3, 2) with the box at (4, 2) and (4, 3) already on docks."

The first clause is a true readback of row 18. The second is false: rows 19 and 26 read plain `Box`, and the level's only other dock, (3, 4), is empty.

Figure 1: One turn, `t0 qwen2.5-7b L4 s6`: the `I2` shape in a single sentence: an accurate readback of the model's own input beside a claim the adjacent rows contradict. The turn reproduces verbatim at the same model, level and step in two of the three temperature-0 executions, the false belief recurring to the end of the episode in both; at temperature 1 the same false-arrival belief appears, not verbatim, in one of three executions (Appendix F). A verbatim span, the strongest evidence class in the paper; `B5`'s episode count is not used (unit defect, §7).

5.5 Fixed-seed greedy decoding did not reproduce trajectories here

Three executions identical in seed, harness, serving stack and configuration, differing only by session: 14 of 30 cells (47%) take an identical path across all three executions at temperature 0, and 1 of 30 (3%) at temperature 1, where a cell is one model $\times$ level and the denominator is the 30 cells of a condition. Step count differs in 16 of 30 cells and final worker position in 15 of 30 at temperature 0; at temperature 1 those are 27 and 28 of 30. No model is exempt, and none reproduces all six of its cells: at temperature 0, identical paths run 4/6, 1/6, 4/6, 2/6 and 3/6 per model.

The phenomenon is documented, its mechanism here is not: temperature-0 serving is not run-to-run invariant (He & Thinking Machines Lab, 2025), ours was not even at the one-request-at-a-time concurrency we ran (Appendix D), and a single flipped near-tie token changes the action and every subsequent observation. The consequence for evaluation is not documented in the same way, and what this section adds is its size at the trajectory level. An agent study reporting one temperature-0 trajectory per configuration is reporting one draw from a distribution it has not sampled, and roughly half of our cells would have supplied a different one. The comparison itself is one triplet of executions: we read 14 of 30 as an existence result about this stack, not as an estimate of a divergence rate. Whether the labels differ where the trajectories do is not evaluated here; it needs all three executions under one adjudicated codebook, which this paper does not have.

6 Evaluation implications

"The agent pushed into empty floor" is an outcome statistic. "The thought asserted a box at the cell it pushed into" is a failure process. Every result above is an instance of that difference, and it is why we report the claim that produced a no-op rather than the no-op itself. An evaluation of this corpus reporting only outcomes would report six identical zeroes and a no-op rate, and could not distinguish an episode in which the agent read the board correctly and built an impossible move from one in which it never read it at all. The generation layer makes the same point from the other end (§5.4).

Ineffective actions are the wrong thing to count. Where a level is most irreversibly lost the losing move is legal and successful (in two episodes the only successful move there is), so a metric counting ineffective actions ranks those episodes as better. This needs no claim about `H1`'s frequency, only that the metric is insensitive to the direction of the damage.

The scaffold is part of the object of study, and we can put a number on it. §7 reports two defects in the prompt this harness sends a Sokoban agent in its default configuration, both unmodified upstream at commit `996d848` (our own harness patch touched neither; Appendix D), and neither a general property of the tool: of its eleven games, Sokoban alone gets a harness-mode prompt substantially shorter than its base-mode one, with the fuller rules text unreachable from harness mode. A benchmark result here is a joint measurement of a model and an under-specified scaffold, and no win-rate number separates them. Our own pre-collection audit missed it: it asked whether the prompt described the board correctly, never whether it stated the preconditions for the actions it offered. The remedy is to ask whether, given only this prompt, a competent player would know the rules and produce a legal turn.

7 LIMITATIONS

Early stopping bounds the horizon. The five-step stuck detector ends most episodes at six steps, so failure processes needing a longer horizon (plan revision, recovery from a mid-game error) are out of reach by construction.

The pre-registered positive control ran, and the instrument did not discriminate. On 291 ladder turns labelled blind, the registered primary `I8`, predicted more common on solved episodes, returned 2 of 18 against 4 of 40 (one-sided Fisher $p = 0.61$, where the registered detectable effect needed 6 of 18), and the secondary moved opposite to registration; we narrow the contribution claim accordingly. With $n = 3$ boards and 12.76 of the 20 solves chance-expected, this is a null against a weak success regime, and its tests rest on annotator A alone (Appendix H).

One scaffold, and it is under-specified in a way that touches a conjunction item. The reasoning prompt closes with a worked example whose thought describes a plain move and whose action is a push in another direction, which is the pattern `E2` codes. It is upstream's, unmodified at commit `996d848`, and is sent on all 1,561 steps, so the example may account for a share of `E2`, and no arm here bounds that share: the example is present in every condition we ran. The prompt also omits the harness's own rules text, and a separate arm tests that omission: 30 episodes with the representation fixes alone and 30 with the same fixes plus the rules text. Neither arm solved a level (0 of 30 in the control and 0 of 30 in the treatment): restoring the rules text did not produce a solve. Content is not separated from length: the treatment prompt is 734 characters longer with no length-matched control (Appendix H). `E2` is narrowed by this disclosure and we do not withdraw it.

Both annotators are language models, so κ measures codebook specificity and not validity (§4). Roughly fifty of the 1,561 turns are named by codebook worked examples, so the second annotator is not blind on those turns; we keep them, because they separate the codebook's most confusable items, and disclose them.

Human coverage is partial and uneven, and neither checker is independent. The 316 verdicts are unevenly spread, with 111 of 316 (35%) on one model, so no per-model claim about label defensibility is available. Both checkers showed the same directional asymmetry, but the smaller cell contains only five verdicts; no inferential claim is made, as pre-committed.

Two items carry a unit defect that makes their counts unsound. `I5` and `B5` describe an episode-level property with a step-level coding condition. The form is real and no label was judged wrong, but the counts are not measurements; the rubric is deliberately unrepaired, having been frozen before both annotations.

Scope. Six fixed hand-authored levels, one harness, five open models of 3–8B, $n = 3$ per cell: presence, breadth and relative magnitude, not per-turn percentages. The taxonomy was induced on the same 180 episodes it is applied to, with no held-out set, and the `I2` shape occurs in one model, so we cannot tell a habit of that checkpoint from a more general dissociation. The two annotations are not adjudicated; §5 reports reading-pass figures, and what adjudication could change is bounded (Appendix H).

8 CONCLUSION

Across all six evaluated conditions, none of the tested agents solved a level; on a fixed ladder of one to six optimal pushes, none solved a board needing four or more (0 of the 60 attempts at or above that point), although all five solved the three-push board at least once, above a uniform-random agent's rate. Turn-level analysis nevertheless distinguishes failure processes that identical aggregate outcomes conceal: some turns of one model re-emit an accurate board beside prose reasoning from another, some describe one move and emit another, and some irreversible losses arise from legal pushes. In the examined cases, detection alone was insufficient: agents that correctly diagnosed a failed action chose a corrective that was equally impossible.

STATEMENT ON THE USE OF LARGE LANGUAGE MODELS

Large language models appear in this work in three distinct roles, separated here because they carry different risks.

As the object of study. The evaluated systems are Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, Llama-3.2-3B-Instruct, Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3, served locally at fp16. They are the subject of the experiment, not a tool used to produce it.

As the measuring instrument. This is the disclosure that bounds our claims. Both annotators are language models: annotator A is Claude Opus 5 and annotator B is GPT-5 via the Codex coding agent, applying a codebook fixed before coding, each blind to the other's labels and to this draft. A third language model, Gemini 3.7 Flash, labelled the whole corpus under the same codebook (§4, Appendix E), and the blind positive control of Appendix H was labelled by annotator A's model, with GPT-6 Astra, a newer model than annotator B's, through the same coding agent, double-coding a pre-committed sample of it for agreement only. The three blind induction passes that produced the codebook were also read by Claude Opus 5, so annotator A is not naive to the taxonomy's framing. The step that matched each pass's descriptions to taxonomy items, and the validation pass over those matches, were likewise carried out in a language-model session under author direction rather than by a human working unaided; they are judgements about the codebook rather than labels on turns, and we count them as instrument construction. No human independently coded a complete episode. This is stated in §4 and again in §7, and it is why we report κ as a measure of codebook specificity rather than of validity, always beside PABAK and prevalence, and why the human label assessment is reported separately and never pooled with the model annotators' agreement. Two model annotators agreeing is not evidence that either is right.

As an authoring aid. The authors used LLM assistance while preparing this work: for drafting and redrafting sections of this paper and editing its prose; for implementing and debugging the analysis and experiment-running scripts; for adversarial review of the argument; and for searching out candidate related work. The mechanically derived figures in the paper are produced by the released analysis tools from the released data rather than written by hand, and an audit script checks the central ones against the manuscript. Every reference was retrieved and then verified against its primary source, because a language model asked for a citation will supply a plausible one: each entry in the bibliography was checked against the arXiv API or the publisher's own record, and candidates that could not be verified were discarded. The authors read, verified and take responsibility for all content, including any errors.

REFERENCES

Wael Albayaydh, Rui Zhao, and Ivan Flechais. Beyond the leaderboard: A synthesis of tool-use, planning, and reasoning failures in large language model agents, 2026. arXiv:2607.05775 (v1).

Shraddha Barke, Arnav Goyal, Alind Khare, Avaljot Singh, Suman Nath, and Chetan Bansal. Agentrx: Diagnosing ai agent failures from execution trajectories, 2026. arXiv:2602.02475 (v1).

Ted Byrt, Janet Bishop, and John B. Carlin. Bias, prevalence and kappa. *Journal of Clinical Epidemiology*, 46(5):423–429, 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?, 2025. arXiv:2503.13657 (v3).

Mengzhuo Chen, Junjie Wang, Zhe Liu, Yawen Wang, Haiming Zheng, and Qing Wang. From failed trajectories to reliable llm agents: Diagnosing and repairing harness flaws, 2026a. arXiv:2606.06324 (v2).

Mengzhuo Chen, Junjie Wang, Fangwen Mu, Yawen Wang, Zhe Liu, Huanxiang Feng, and Qing Wang. Seeing the whole elephant: A benchmark for failure attribution in llm-based multi-agent systems, 2026b. arXiv:2604.22708 (v1).

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

Joseph C. Culberson. Sokoban is PSPACE-complete. In *Proceedings in Informatics 4: Fun With Algorithms*, pp. 65–76. Carleton Scientific, 1999. Also available as Technical Report TR 97-02, University of Alberta, 1997.

Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. Trail: Trace reasoning and agentic issue localization, 2025. arXiv:2505.08638 (v3).

Dorit Dor and Uri Zwick. SOKOBAN and other motion planning problems. *Computational Geometry*, 13(4):215–228, 1999. doi:10.1016/S0925-7721(99)00017-6.

Alvan R. Feinstein and Domenic V. Cicchetti. High agreement but low kappa: I. The problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549, 1990.

Huan-ang Gao, Zikang Zhang, Tianwei Luo, Kaisen Yang, Xinzhe Juan, Jiahao Qiu, Tianxing Chen, Bingxiang He, Hao Zhao, Hao Zhou, Shilong Liu, and Mengdi Wang. Cubebench: Diagnosing interactive, long-horizon spatial reasoning under partial observations, 2025. arXiv:2512.23328 (v3).

Ishir Garg, Neel Kolhe, Dawn Song, and Xuandong Zhao. Memfail: Stress-testing failure modes of llm memory systems, 2026. arXiv:2605.26667 (v1).

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023. doi:10.1073/pnas.2305016120.

Shou-Tzu Han, Rodrigue Rizk, and KC Santosh. Fragile reasoning: A mechanistic analysis of llm sensitivity to meaning-preserving perturbations, 2026. arXiv:2604.01639 (v1).

Horace He and Thinking Machines Lab. Defeating nondeterminism in LLM inference. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250910. `https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/`.

Lanxiang Hu, Mingjia Huo, Yuxuan Zhang, Haoyang Yu, Eric P. Xing, Ion Stoica, Tajana Rosing, Haojian Jin, and Hao Zhang. lmgame-bench: How good are llms at playing games?, 2025. arXiv:2505.15146 (v2).

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet, 2023. arXiv:2310.01798 (v2); ICLR 2024.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. LLMs can't plan, but can help planning in LLM-Modulo frameworks, 2024. arXiv:2402.01817 (v3); ICML 2024, PMLR 235.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention, 2023. arXiv:2309.06180; SOSP 2023.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. arXiv:2307.13702 (v1).

A. Lawsen. Comment on the illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity, 2025. arXiv:2506.09250 (v2).

Oorja Majgaonkar, Zhiwei Fei, Xiang Li, Federica Sarro, and He Ye. Understanding code agent behaviour: An empirical study of success and failure trajectories, 2025. arXiv:2511.00197 (v1).

Mayug Maniparambil, Nils Hoehing, Janak Kapuriya, Arjun Karuvally, Ellen Rushe, Anthony Ventresque, Noel O'Connor, and Fergal Reid. Topobench: Benchmarking llms on hard topological reasoning, 2026. arXiv:2603.12133 (v1).

Nicolas Martorell. From text to space: Mapping abstract spatial models in llms during a grid-world navigation task, 2025. arXiv:2502.16690 (v1).

Tural Mehtiyev and Wesley Assunção. Beyond resolution rates: Behavioral drivers of coding agent success and failure, 2026. arXiv:2604.02547 (v1).

Sebastiano Monti, Carlo Nicolini, Gianni Pellegrini, Jacopo Staiano, and Bruno Lepri. Sokobench: Evaluating long-horizon planning and reasoning in large language models, 2026. arXiv:2601.20856 (v1).

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. BALROG: Benchmarking agentic LLM and VLM reasoning on games, 2024. arXiv:2411.13543 (v2); ICLR 2025.

Nicholas Pangakis, Samuel Wolken, and Neil Fasching. Automated annotation with generative AI requires validation, 2023. arXiv:2306.00176 (v1).

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity, 2025. arXiv:2506.06941 (v3); NeurIPS 2025.

Rongyuan Tan, Jue Zhang, Zhuozhao Li, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. Contrastive attribution in the wild: An interpretability analysis of llm failures on realistic benchmarks, 2026. arXiv:2604.17761 (v1).

Zhisheng Tang and Mayank Kejriwal. Grasp: A grid-based benchmark for evaluating commonsense spatial reasoning, 2024. arXiv:2407.01892 (v2).

Oguzhan Topsakal, Colby Jacob Edell, and Jackson Bailey Harper. Evaluating large language models with grid-based game competitions: An extensible llm benchmark and leaderboard, 2024. arXiv:2407.07796 (v2).

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting, 2023. arXiv:2305.04388 (v2); NeurIPS 2023.

Nemika Tyagi, Mihir Parmar, Mohith Kulkarni, Aswin RRV, Nisarg Patel, Mutsumi Nakamura, Arindam Mitra, and Chitta Baral. Step-by-step reasoning to solve grid puzzles: Where do llms falter?, 2024. arXiv:2407.14790 (v2).

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change, 2023a. arXiv:2206.10498 (v4); NeurIPS 2023 Datasets and Benchmarks Track.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models: A critical investigation, 2023b. arXiv:2305.15771 (v2); NeurIPS 2023 (spotlight).

Karthik Valmeekam, Kaya Stechly, and Subbarao Kambhampati. LLMs still can't plan; can LRMs? a preliminary evaluation of OpenAI's o1 on PlanBench, 2024. arXiv:2409.13373 (v1).

Xinyu Jessica Wang, Haoyue Bai, Yiyou Sun, Haorui Wang, Shuibai Zhang, Wenjie Hu, Mya Schroder, Bilge Mutlu, Dawn Song, and Robert D Nowak. The long-horizon task mirage? diagnosing where and why agentic systems break, 2026a. arXiv:2604.11978 (v1).

Zehong Wang, Fang Wu, Hongru Wang, Xiangru Tang, Bolian Li, Zhenfei Yin, Yijun Ma, Yiyang Li, Weixiang Sun, Xiusi Chen, and Yanfang Ye. Why reasoning fails to plan: A planning-centric analysis of long-horizon decision making in llm agents, 2026b. arXiv:2601.22311 (v1).

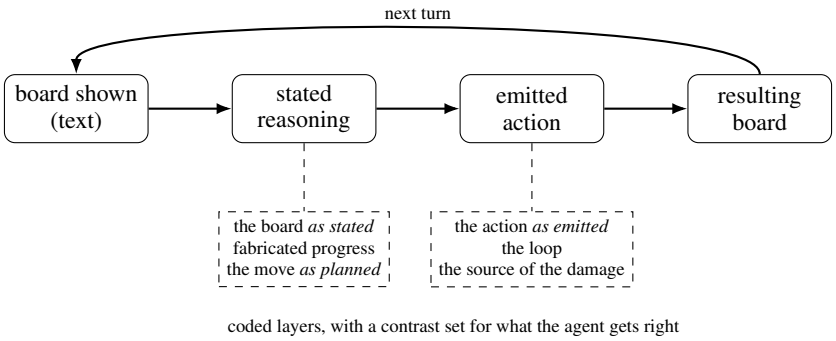


Figure 2: The unit of analysis is the turn, not the episode. For each of 1,561 steps we hold what the agent was shown, what it said it was thinking, the action string it emitted, and the board that resulted, so a mistaken belief is legible against the state that contradicts it, at the step where it was formed. The dashed boxes name the coded layers; the contrast set makes a step carrying no failure label distinguishable from a step nobody looked at.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. Agentracer: Who is inducing failure in the llm agentic systems?, 2025a. arXiv:2509.03312 (v2).

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems, 2025b. arXiv:2505.00212 (v3).

Yinger Zhang, Shutong Jiang, Renhao Li, Jianhong Tu, Yang Su, Lianghao Deng, Xudong Guo, Chenxu Lv, and Junyang Lin. Deepplanning: Benchmarking long-horizon agentic planning with verifiable constraints, 2026. arXiv:2601.18137 (v1).

Xiangxin Zhao, Han Li, Shuaiting Li, Tianyi Zhao, Earl T. Barr, Federica Sarro, and He Ye. Failure as a process: An anatomy of cli coding agent trajectories, 2026. arXiv:2607.09510 (v1).

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, 2023. arXiv:2306.05685 (v4); NeurIPS 2023 Datasets and Benchmarks Track.

Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, Xiaoteng Ma, Xiaodong Yu, Gowtham Ramesh, Jialian Wu, Zicheng Liu, Pan Lu, James Zou, and Jiaxuan You. Where llm agents fail and how they can learn from failures, 2025. arXiv:2509.25370 (v1).

Taiyu Zhu, Yifan Wu, Weilin Jin, Ying Li, and Gang Huang. Stepfinder: A temporal semantic framework for failure attribution in multi-agent systems, 2026. arXiv:2606.03467 (v1).

A THE FULL ITEM INVENTORY

Three numbers do three different jobs, and conflating them would misrepresent this paper as claiming ninety-odd distinct causes. Ten families (`A`–`J`) are the structure of the taxonomy. Seven findings across eight items are what the conjunction of §5.3 actually rests on. And 101 pre-merge / 99 top-level / 97 codeable items are a deliberately over-split inventory: an upper bound on the repertoire, not a count of causes.

We never write "99 failure modes". The inventory is over-split on purpose: during induction, `NEW` was the default whenever a fit was arguable, because absorbing a novel form into an existing item silently manufactures saturation while over-splitting is recoverable by merging later. Its size is evidence for the method argument; it says nothing about the number of distinct mechanisms.

The file holds 86 recurring items and 13 single observations. Three merges were made at codebook construction, one item (I8) was added at v5 before any labelling began, and nothing was renumbered or deleted, which is why the numbering has gaps: A15, B12 and F17 are the merged items' retired numbers, and B6 and F11 were never assigned. Two of the 99 top-level items, I4 and I7, are *input* findings; no coder applies them to a step. Hence the codeable count of 97, and hence I4 sits outside the reliability comparison of Appendix E by construction.

A single observation is quoted, never counted: the thirteen ONE items appear in this paper as verbatim examples with no frequency attached to them.

B THE CODEBOOK

The codebook is organised by where in the turn the claim goes wrong rather than by severity, and is written to be self-contained: a second coder needs it and nothing else, and if another document is required to decide a label, that is a defect in the codebook. Table 3 lists all 97 codeable items with their one-line definitions and recurrence tags. The full codebook (per-item coding conditions, exclusions, discriminating questions, boundary tests against confusable neighbours, and a worked example per item) is in the supplementary material; it is too long to reproduce here. It is the document both annotators were given, fixed before either annotation began, and Table 3 is generated from it.

Four coding rules change numbers and are stated wherever those numbers appear. (i) A step-N thought is judged against the step-$(N-1)$ board, because the logged observation is the post-action one. (ii) Perseveration is defined at the *action* level, with the clause that every action in the run must be a no-op. (iii) Recognition of a failure without adaptation still counts as failure to close the loop; our claim is that the agent almost never registers the non-effect and, when it does, does not act on it, never that it never registers it. (iv) Every label requires a verbatim span from the thought of its own step; a label with no span is not a label.

Table 3: The 97 codeable items of codebook v5, with recurrence tags. MANY marks a recurring form and ONE a single observation, which this paper quotes and never counts. [EP] marks the 11 episode-level items and [RUN] the one run-level item (F1); the other 85 are scored per step. The tags are generated from the same unit column the reliability analysis uses, so this table and Table 5 cannot disagree about an item's unit. I4 and I7 of the 99-item inventory are input findings and are not listed. Generated from the codebook.

item	definition	recurrence
A1	the worker's own cell is misstated	MANY
A2	the stated worker cell is a wall	MANY
A3	the stated worker cell holds a box	MANY
A4	a box is named at a cell that holds no box	MANY
A5	empty floor is called a wall or otherwise named as an obstruction	MANY
A6	a box is called a wall	MANY
A7	a wall is called a dock, or a dock is called a wall	MANY
A8	a dock is named at a cell that is ordinary floor	MANY
A9	a real dock is given the wrong role	MANY
A10	worker and box reported on the same cell	MANY
A11	one thought contradicts itself about the same cell	MANY
A12	a non-existent box is given a persistent identity and planned around	MANY
A13	a coordinate outside the board, including a negative index	MANY
A14	an object is named as its own obstruction	MANY
A16	two different identifiers for one object inside a single clause	ONE
A17	a real box or dock is dropped from the reasoning and never mentioned again	MANY [EP]
B1	a push is asserted as already completed when nothing has moved	MANY
B2	the fabricated progress is present in the step-1 thought	MANY
B3	the described board is the board that would exist if the intended push had succeeded	MANY
B4	an undocked box is declared docked or "in place"	MANY
B5	the worker standing on an empty dock is read as a box having arrived	MANY
B7	the transcribed board is itself fabricated	ONE
B8	the transcribed board is stale: table, prose and board give three different positions	MANY
B9	a multi-step, internally consistent fiction inherited forward as established fact	MANY
B10	the level is declared complete with every box off dock	MANY [EP]
B11	the thought is restated word-for-word on an unchanged board, carrying tokens of change	MANY
C1	an already-docked box is treated as still needing docking, or as an obstruction to clear	MANY
C2	the objective is inverted: the plan is to push a box off its dock	MANY
C3	the already-docked box and the free docks are never mentioned at all	MANY [EP]
C5	an occupied dock is treated as a free target for a different box	MANY
C6	a box one push from a dock is never mentioned in the episode	MANY [EP]

item	definition	recurrence
D1	a push is issued at a box the worker is not adjacent to	MANY
D2	the worker stands on the wrong side of the box for the push it describes	MANY
D3	the cell the push must be made from is a wall, so the push can never exist, and this is never noticed	MANY
D4	the worker is sent to stand on a cell a box occupies	MANY
D5	the worker is sent to a cell it already occupies, or told to satisfy a prerequisite it already satisfies	MANY
D6	the push destination is the box's own cell, or worker and box are sent to one cell	MANY
D7	the named direction does not match the geometry	MANY
D8	a displacement no single push produces	MANY
D9	a multi-move plan is emitted on a single-action turn	MANY
D10	a push direction outside the action space	MANY
D11	a mechanic the game does not have is invented and used as the plan	MANY
D12	a negation error inverts the stated intent	ONE
D13	a diagonally placed piece is asserted to be adjacent	MANY
D14	the push destination is a cell another box occupies	MANY
D15	the box is treated as something to pick up, receive, carry or unload	MANY
D16	a systematic axis relabelling held across the episode, which then corrupts its own blocking checks	MANY [EP]
E1	the emitted action is in a different direction than the one described in the same thought	MANY
E2	a plain move is described and a push is emitted, or the reverse	MANY
E3	the plan is correct and the action ignores it	MANY
F1	the same action is repeated after it changed nothing, and no step registers the no-op	MANY [RUN]
F2	the same thought is restated across steps although the board changed	MANY
F3	a stale position is carried forward: the box's pre-push cell keeps being named	MANY
F4	a previous action's failure is noticed, and the response is to swap direction rather than diagnose	MANY
F5	a reflection evaluates an action that never happened, or reports change where there was none	MANY
F6	a two-cell oscillation with the thought alternating in step, each swing presented as a fresh decision	MANY
F7	a thought reverses itself inside one paragraph	MANY
F8	the thought grows by accretion, appending each step's paragraph to the previous ones	ONE
F9	the prompt's own text is emitted in place of reasoning	MANY
F10	an accurate self-diagnosis is produced and the corrective chosen is equally impossible	MANY
F12	a move is declared invalid and re-proposed inside the same sentence	MANY
F13	the worker's own enclosure is never registered	MANY [EP]
F14	a stated prerequisite is achieved and the next step, without registering it, gives it up	MANY
F15	a fabricated object appears once and is silently dropped, with no step revising it	MANY
F16	the worker is declared stuck, cornered or trapped when it has open legal moves	MANY
F18	the level-winning move is identified and then rejected as a deadlock	ONE
G1	the goal is replaced by a housekeeping subgoal no rule of the game asks for	MANY
G2	the plan pushes away from the docks	MANY [EP]
G3	the level's real docks are never named	MANY
G4	the objective is replaced by a different game's objective: boxes are collected from docks	ONE
G5	the current puzzle is abandoned mid-episode for hypothetical future puzzles	ONE
G6	the target commitment switches between consecutive steps with no board change	MANY
G7	a level-scale sealed region makes every plan in the episode impossible, and is never registered	MANY [EP]
H1	the successful pushes are what lose the level	MANY
H2	a successful action moves a box the thought never named, or executes a push scheduled for later	MANY
H3	a cell ID resolving to the wrong cell, used stably beside a correct one	MANY
H4	an ID and an `(x, y)` pair are given for one object and resolve to different cells	ONE
H5	the reasoning names no cell at all	ONE [EP]
H6	notation switches mid-episode between cell IDs and `(x, y)` for the same cell	MANY
H7	a box count that disagrees with the board	MANY [EP]
H8	two separate objects of the same class described as one duplicated thing	MANY
H9	an unrecoverable state is created and the same thought asserts there is no deadlock	MANY
H10	a malformed hybrid reference: a cell number paired with a loose row digit, the halves disagreeing	MANY
H11	one identifier used for two different objects in the same thought	ONE
H12	a set of docks is collapsed to a single member; the others are never planned for	MANY [EP]
I1	the opening position statements are correct and the move built from them is impossible	MANY
I2	the echoed board table is correct while the prose in the same thought contradicts it	MANY
I3	the transcribed table omits the worker entirely	MANY
I5	one class of object is tracked accurately throughout while the other is fabricated	MANY
I6	a true and decisive fact about the board is stated, and the opposite conclusion is drawn from it	MANY
I8	the board is read correctly and the move built from it is legal and correct	MANY
J1	the harness's own logging scaffold is emitted inside the thought	MANY
J2	runaway verbatim template repetition, cut off mid-sentence	MANY
J3	no action is emitted at all; the turn ends at a bare prompt token	MANY
J4	a stray non-language fragment leaks into the thought	ONE
J5	the response alternates between two distinct formats on a fixed parity	ONE
J6	a fabricated future turn in the harness's own log format, asserting a completed impossible move	ONE

C THE BLIND-PASS AUDIT TRAIL AND BOTH SATURATION CURVES

THE PROTOCOL

Three blind reading passes, one per execution. Each read that execution's 60 episodes (all five models, all six levels, both temperatures), and no episode was read by more than one, so the three together cover all 180 episodes exactly once. Throughout this appendix "pass N" means the reading pass, never the execution; the two are numbered alike because reading pass N read execution N, and Table 1 names the executions. Because every reading pass spans the same 60 model $\times$ level $\times$ temperature cells, passes 2 and 3 are replicates of the same conditions rather than new material, which is what makes the saturation comparison below meaningful.

Every pass was read by empty-context reader agents, each given one six-episode trace workbook and a fixed brief, with no checklist, no rubric, no category codes, and no other pass's findings. Readers describe what they see; matching a description to a taxonomy item happens afterwards, as a separate written step, recorded one line per description in a per-pass mapping file (`pass2_mapping` carries 1,042 descriptions and `pass3_mapping` 1,314), so any single mapping call can be contested without re-reading the pass. The three passes produced 1,552 step lines and 2,194 quoted spans across 180 episodes (547 / 481 / 524 step lines; 751 / 693 / 750 spans).

Quote auditing is mechanical at every stage: one script audits the observation files against the trace workbooks, another audits the checklist and codebook for ellipsis splices, markdown inside quoted spans, and internal count consistency. Every step-level finding carries a verbatim span checked mechanically against the thought of its own step, and spans failing the check were discarded; the analyst repaired none. A validation pass then checked the finished checklist against all three passes; that pass is what corrected the saturation curve below.

THE TWO CURVES, AND THE GAP BETWEEN THEM

curve	what it counts	value
item-discovery	items each grouping step added to the file	**56 / +26 / +13**
saturation	forms absent from every earlier pass	**56 / +21 / +5**

Denominator: 3 passes $\times$ 60 episodes. We report both. The gap between them is almost entirely forms an earlier pass had already described and the grouping step had filed under a neighbour; it is accounted for step by step below.

The curves have to be joined to the inventory of Appendix A, because the two headline counts must meet. The discovery curve counts items a grouping step added as newly observed: $56{+}26{+}13 = 95$. Six further items were created at the grouping steps by splitting an existing item (pass 2 split `B10`, `C5` and `F12` out of `B4`, `C1` and `F7`; pass 3 split `A17`, `D16` and `G7`), and a split is not a discovery: the form was already in the file inside its parent, so it counts toward neither curve. $95 + 6 = 101$ pre-merge. Three merges at codebook construction take 101 to 98, one item (`I8`) added at v5 takes 98 to 99 top-level, and setting aside the two input findings `I4`/`I7` leaves the 97 codeable items.

Twelve items that entered the file at pass 2 or pass 3 had already been written out, with quotes, in an earlier pass's observations. They were read, described in the reader's own plain language, and then filed under a broader neighbour at the earlier grouping step, and separated only later. They are therefore splits and do not count toward saturation. The two sharpest cases are both high-frequency forms: `D14` (a box pushed into a cell another box occupies) and `B11` (a thought restated verbatim on an unchanged board while carrying change-tokens) were each read, described and filed under a neighbour by two independent blind 60-episode passes, under `D4`/`D6` and `F1`/`F2` respectively. It took the third reading pass to separate them. It is visible quantitatively in the mapping files: `D4` and `D6` fall by 26 descriptions between `pass2_mapping` and `pass3_mapping` while `D14` takes 23; `F2` collapses from 40 descriptions to 3 while `B11` takes 19. The same sitting applied the split tag correctly to three other items and not to these, even though the evidence pattern was the same, the tags came out opposite, and it happened in one session. The bottleneck is therefore the grouping step rather than the reading step.

How +26 / +13 became +21 / +5. The validation pass reclassified eleven items as splits, forms already present in an earlier pass's evidence and filed under a neighbour, and they do not all come off

the same pass: three come off pass 2, taking $+26 \rightarrow +23$, and eight come off pass 3, taking $+13 \rightarrow +5$. Two adjudications then move pass 2 alone: $+23 \rightarrow +22$ (a twelfth item, `I6`, adjudicated a first-seen split) and $+22 \rightarrow +21$ (the `A14`/`F17` merge, which takes one pass-2 addition off the tally). The `I6` step has a legitimate counter-reading: a reviewer who holds that a conjunction item requires the conjunction to be articulated as such returns `I6` to pass 2, which reverses that one step and no other, giving 56 / +22 / +5. We report both numbers.

Neither curve licenses completeness. $+5$ on a third 60-episode pass is a narrowing repertoire, not a closed one: it bounds what a single pass can build and does not show that the inventory is complete.

One alternative reading should be named. Because passes 2 and 3 replicate the same 60 cells (30 model $\times$ level cells at each temperature) and add no new material, and because §5.5 shows those replicates take different paths, an item first itemised at pass 3 could in principle be new behaviour in a different trajectory rather than a form the earlier grouping step missed. The twelve-item finding is immune to that, because those twelve were described in an earlier pass's own prose, so the forms existed in the earlier trajectories, whatever else differed. The alternative bites on any item claimed as newly absent before pass 3, and we claim none.

The evidence class. These figures are process-derived, not mechanical: they are contestable by a third party re-matching pass prose to item definitions. Every one of the twelve "already described in an earlier pass" calls was made by us, and we are the party the gap benefits. The `I6` counter-reading above demonstrates that such a call can go either way, which means the other eleven are contestable too and we have not contested them. The check that would settle it: give a third party the pass-1 and pass-2 observation prose and the pass-3 item definitions, with this claim withheld, and ask them to match descriptions to items. If ten or eleven of the twelve survive, the result stands on someone else's judgement; if six survive, the curve moves. The check is outstanding, and the result is not offered as a contribution of the paper.

Hand-written counts in this project, and why every figure is now derived

Hand-written counts in this project's own record-keeping were wrong nine times across seven audits, every one inside a section about counting, and not one was caught by reading; a script or a deliberate recomputation caught all nine, over roughly a dozen human passes that did not. The class kept firing inside the sentences written to prevent it. A taxonomy paper rests entirely on counts, and the only defence that has worked here is deriving them mechanically (Appendix D).

D Reproducibility configuration

Harness. `lmgame-org/GamingAgent` at upstream commit `996d848ae5e3bf68433d663f38ef4da5bdfe5332` (short `996d848`, authored 2025-09-11), with `harness: true`, running the full per-step loop (perception, memory and reflection, reasoning) with each stage a separate model call and `use_reflection: true`. The memory window is five turns, not the ten the setting names: `max_memory` counts entries and each turn appends two. In text mode the perception module makes no API call, so its prompt never reaches the model.

Harness state. The state of the reported corpus, recorded in each condition's `condition.json` and verified by a gate on the prompt hash: the renderer and vocabulary normalisation applied; the reflection board-description fix applied; the clean history encoding applied; the reflection-presupposes-prior-action defect deliberately not fixed (Appendix H); the prompt-gating dead code not fixed; memory window 5. In the bundled manifests and scripts these harness states carry the prefix `HH`, to keep them clear of the taxonomy's family-`H` items: `H1`, `H3`, `H5` in the results are always labels a coder applies to a step.

Serving. Self-hosted vLLM 0.26.0 (Kwon et al., 2023) behind an OpenAI-compatible endpoint on a single NVIDIA L4 (sm_89), with `VLLM_USE_FLASHINFER_SAMPLER=0`, `VLLM_ATTENTION_BACKEND=FLASH_ATTN`, `token_limit` 4096, seed 42, and `max_unchanged_steps_for_termination` 5. Concurrency is one: the driver runs `for level: for rep: run_episode(...)` against a single endpoint with no threading and no request batching, `--max-num-seqs` is left at the server default and never

exercised beyond one in-flight request, and the server is a fresh process per model. CUDA graphs are on (`--enforce-eager` is applied only below compute capability 8; every condition records 8.9). The consequence for §5.5 is that the trajectory divergence there cannot be concurrent-request batching, because there were no concurrent requests. What remains is run-to-run non-invariance across separate server processes; this study does not localise it further. No paid API was used at any point in serving the five models. Models: the HuggingFace repositories `Qwen/Qwen2.5-3B-Instruct`, `Qwen/Qwen2.5-7B-Instruct`, `meta-llama/Llama-3.2-3B-Instruct`, `meta-llama/Llama-3.1-8B-Instruct` and `mistralai/Mistral-7B-Instruct-v0.3`, all fp16; precision is not a variable in this study. Every model is the instruction-tuned checkpoint; the exact repository strings are recorded per condition in the bundled `manifests/*_condition.json`. Short forms elsewhere in the paper (Qwen2.5-3B, Llama-3.1-8B) always denote these checkpoints. A planned Gemma-2 arm was dropped because it does not serve on this architecture.

Determinism checks. All are mechanical. The six derived initial boards resolve uniquely and are md5-identical across all six conditions and across all five historic datasets this project holds. The historic sets are superseded and are not bundled, so the supplement re-runs the six-condition half of this check and reports the other half as not re-run. Temperature-0 and temperature-1 conditions of one execution share a run fingerprint and the three executions differ, which is the pairing the design depends on.

The unit of analysis. Per-step JSONL, one line per step, carrying `step`, `agent_action`, `thought`, `reward`, `perf_score`, `info` (the environment's own step counter, worker position and docked-box count), `agent_observation` (the rendered text board), `terminated`, `truncated` and `time_taken_s`, so each line is directly traceable board $\rightarrow$ thought $\rightarrow$ action $\rightarrow$ next board. Note that `agent_observation` is the post-action board, so a step-N thought is judged against the step-$(N-1)$ board; step-1 boards are derived from the deterministic level definitions.

Divergence, in full. §5.5 reports the headline. Across three executions identical in seed, harness, serving stack and configuration and differing only by session: 14 of 30 cells identical at temperature 0 and 1 of 30 at temperature 1; step count differs in 16 of 30 cells and final worker position in 15 of 30 at temperature 0, and 27 and 28 of 30 at temperature 1; pairwise the three executions diverge on [12, 13, 14] of 30 cells at temperature 0 and [25, 23, 26] of 30 at temperature 1. Per model at temperature 0, identical paths across all three executions run qwen2.5-3b 4/6, qwen2.5-7b 1/6, llama3.2-3b 4/6, llama3.1-8b 2/6, mistral-7b 3/6, where the denominator is that model's six levels and the five sum to the 14 above. At temperature 1 they run qwen2.5-3b 0/6, qwen2.5-7b 1/6, llama3.2-3b 0/6, llama3.1-8b 0/6, mistral-7b 0/6, summing to the 1 above.

The oracle replay. This is the first of the two arms of §5.2. The model is removed from the loop and a push-optimal solution is substituted for exactly one of the three calls the per-step loop makes (`env.render()`, `agent.get_action()`, `env.step()`), so parser, simulator, renderer, termination logic, logging and the episode record all stay on the real code path with their real arguments. The solutions come from a solver that imports nothing from the harness and re-implements the rules from the definition of the game, which is what makes its agreement with the harness's own simulator evidence rather than a tautology. Over 229 replayed steps on the six levels of the reported corpus: all six solve in the configuration the difficulty ladder ran in, with the termination flag set, every box on a dock and the solving step present in the episode file; one of six solves in the configuration that produced the reported corpus, where levels 1–5 report the flag unset, no box docked, and one row fewer than the steps actually taken. No cell in either configuration renders as `Unknown`. The cause is a logging defect, measured here: on the solving step `step()` progresses to the next level and returns *before* the adapter logs that step, so the step is absent from the file, the flag comes back unset, and the docked-box count reads zero because the board has already been replaced. Level 6 is the single exception in the corpus configuration, and only because there is no seventh level to progress into. That defect is fixed in the ladder's configuration and was left in place in the corpus's; it is the only difference between the two sets of rows, since the replay itself is identical in both.

The defect cannot have hidden a solve in the reported corpus. A push moves one box by one cell, so a solve needs the state just before it to be one push from done, and no logged state is, nor is any level's starting board. On no level does the number of docked boxes ever rise above its starting value (zero on levels 1–5, and one on level 6, which ships one of its five boxes already docked),

so no level with two or more boxes ever has all but one docked. Level 2 is the only one-box level, and across its 273 logged steps its box never comes closer than five cells (Manhattan distance) to its dock. The rows themselves also rule out an unlogged jump to the next level: step numbers run without a gap and equal the environment's own step counter on all 1,561 rows, no episode's wall layout changes, the six levels' layouts are all different, and the worker never moves more than one cell between consecutive rows. All of this is checked by script over the full corpus, and none of it uses the termination flag or the solved field.

The difficulty ladder. Same harness commit, same serving stack, same five checkpoints at fp16, same 4096-token limit as the reported corpus, run in one session per temperature. The level set is an exhaustive enumeration on a single 5×5 open room: for each optimal push count from 1 to 6, the first placement at that count in enumeration order. The rule names no model and no outcome, and there is no seed, no generator and no ordering choice inside it. The 120 episodes are five models $\times$ six rungs $\times$ one replicate at temperature 0 and three at temperature 1, that is, 30 and 90 episodes, whose denominators are reported as they stand and never reconciled into one. Temperature 0 takes one replicate, and the design's reason for that, greedy decoding making replicates bit-identical, is one §5.5 does not support: identical temperature-0 executions took the same path in only 14 of 30 cells there. The 30 temperature-0 episodes therefore carry no within-condition variance estimate, and none is claimed for them. The temperature-1 replicates are independent because the sampler is unseeded, not because of a per-replicate seed: the seed set on the command line is the environment's, it is inert for predefined levels and never reaches the sampler. That was measured: all 30 model $\times$ rung cells at temperature 1 produced three distinct action sequences. The one harness difference from the reported corpus is the level-progression logging defect above, which is fixed here.

The ladder's chance baseline. It was registered before any episode ran. The baseline column of Table 2 is the probability that a uniform-random agent solves a rung, drawing uniformly over the harness's nine action strings (the four moves, the four pushes and the no-op) and stopped exactly the way the harness stops an episode: on a solve, after five consecutive board-unchanged steps, or at the 200-step budget. It is estimated on the independent simulator above, at 20,000 episodes per rung under a fixed seed, so it costs no GPU time and no harness. Across the six rungs it runs 41.5%, 16.2%, 6.1%, 0.14%, 0.08% and 0.06%. Registering it in advance is what makes the rung-1 comparison reportable at all: 8 of 20 is 40.0%, slightly below the 41.5% baseline, so the solve rate alone would have read as partial competence. Rungs 1–3 are calibration on that basis and their rates are never reported without the baseline beside them.

Availability. The per-condition manifests, the prompt hash, the gate and analysis scripts, the per-item reliability table as a CSV, the per-step scaffold-echo flag table behind §5's `J1`/`I2` split, the full codebook, the trace episodes behind every verbatim span quoted in this paper (54 episode files: every cited model $\times$ level $\times$ temperature cell, all three executions), the *full* 180-episode trace corpus with each condition's run record and batch summary, the verbatim reader brief and annotation prompts (Appendix H), the three model annotators' per-turn label and coverage sheets with the script behind the same-turn counts of Appendix E, the positive control's registration, key, both model annotators' labels, dispatch rosters, unmatched-observation ledgers and readout, together with the readout of its human `I8` check (no human verdict sheet is released), and, for the two arms of §5.2, neither of which enters the annotated corpus, the ladder's level set, its pre-registered chance baseline, all 120 ladder episodes with their condition records, and the oracle replay's per-level readout, are available as the anonymised supplementary material accompanying this submission, a single archive whose `README` maps each directory to the section of this paper that uses it; a permanent public archive replaces it on acceptance. Every figure in this paper is derived by a named script, and the two long tables in Appendices B and E are generated from those sources; the supplementary bundle contains the generators, so both tables can be rebuilt and checked against what is printed here. The ladder table is reproducible the same way: the bundled readout script, run on the bundled episodes, regenerates it. Both full-corpus gates ship with the corpus they read, so Table 1's source check and the no-unrecorded-win check of §5.2 can be re-run from the bundle over all 180 episodes and 1,561 steps.

Licences and data provenance. The harness and the six levels are `lmgame-org/GamingAgent` (Hu et al., 2025), MIT-licensed, run from a clean checkout at the commit above with the run-time harness state recorded above; the five models are the named HuggingFace checkpoints, each under its own model licence as stated on its repository page; the

Table 4: Claim, evidence and human-check coverage. κ is agreement between the two model annotators over 1,561 steps; three items fall in the 10–90% band where the gate applies and `E2` passes it; n/a marks the rest, outside the band, where agreement and PABAK are the readable statistics (`H1`'s κ clears 0.6 but its 0.67% prevalence is out of band, so no verdict applies). `I2` and `E2` are the two the argument rests on; `F10` and `F4` sit outside the conjunction, their verdicts still in the total. The checked column counts human label-verdicts across three never-pooled instruments, as counts and never as a validity rate: 79 sampled verdicts, 70 confirmed, 9 unclear, 0 wrong; an unclear verdict is a finding about the codebook.

item	carries	κ	gate	checked	confirmed	unclear
correct state is present, and is not used consistently						
`I2`	board correct in the model's own output	0.259	n/a	1	1	0
`I1`	accurate reading, wrong inference	0.576	below	31	30	1
`I5`	accurate reading, wrong inference	0.591	below	6	2	4
`I6`	accurate reading, wrong inference	0.266	n/a	4	4	0
`A6`	corroboration	0.512	n/a	4	4	0
`B5`	accurate readback beside a false claim	0.476	n/a	4	2	2
detection is not the missing piece (kept as evidence, outside the conjunction)						
`F10`	detected, diagnosed, proceeded anyway	0.346	n/a	4	3	1
`F4`	the alternative it rules out	0.441	n/a	4	3	1
the plan and the emitted action dissociate						
`E3`	correct plan, wrong action	0.522	n/a	3	3	0
`E2`	correct plan, wrong action	**0.749**	**passes**	12	12	0
a successful action can be more damaging than an ineffective one (carried as a metric argument)						
`H1`	**sole frequency evidence**	0.664	n/a	4	4	0
`H9`	corroboration	0.443	n/a	2	2	0
total human label-verdicts				**79**	**70**	**9**

trace episodes in the supplementary material are our own runs' logged outputs, which embed board layouts from the harness's level definitions. The supplementary bundle carries this statement, with the per-checkpoint licence names, as a file.

E Reliability and validity in full

What is being compared

Each of the 85 step-level items is a cell at every step, each of the 11 `[EP]` items is a cell at every episode, and the 1 run-level item (`F1`, the unit its pre-committed definition names) is a cell at every run, $n = 878$; a label present in an annotator's sheet is a 1 and absence is a 0. Absence is a real judgement here, because both protocols required exhaustive application. `I4` is derived by script and is outside this comparison by construction. Annotator A produced 10,996 label rows over 180 episodes and 1,561 steps; annotator B produced 7,100 rows over the same 180. Every episode is labelled twice, so agreement is computed on 100% of the corpus.

The gate and its summary

The $\kappa \geq 0.6$ gate applies only to items with prevalence between 10% and 90%, measured in the item's own unit: per step over $n = 1{,}561$ for the 85 step-level items, per episode over $n = 180$ for the 11 `[EP]` items, per run over $n = 878$ for `F1`. These are the same units the comparison itself is computed on, and the units this paper elsewhere insists are not interchangeable. Prevalence for the band is the mean of the two annotators' prevalences (the CSV's `prevalence_mean_pct` column). The choice matters on this data, because the two prevalences diverge systematically: an either-annotator criterion would put 30 items in band and a both-annotators criterion 19, against the mean's 25. Outside that band chance agreement is already near ceiling, so Cohen's κ (Cohen, 1960) collapses toward zero or is undefined even at very high observed agreement (the κ paradox (Feinstein & Cicchetti, 1990)), which is why PABAK (Byrt et al., 1993) and both prevalences are reported

beside it. Of the 97 codeable items, 25 fall inside the band (17 step-level, 7 episode-level: A17, H7, H12, C6, G2, G7, F13; and 1 run-level, F1); the other 72 are reported with n, observed agreement, both prevalences and PABAK, and marked paradox, one or degenerate rather than given a κ verdict. All 25 are usable: 9 pass, with median in-band κ 0.576 (range 0.133–0.918) and median observed agreement across all 97 items 97.9%. These are the figures the supplementary per-item CSV reproduces directly; there is no second derivation.

F1's unit. An earlier version of this appendix reported it differently. F1 is compared per run, not per step, because the evidence for it describes runs. For part of this study the comparison script scored it per step, and the summary above was reported with F1 held out (24 usable, 8 passing, median 0.564) beside the CSV's 25 / 9 / 0.576. The script now scores F1 at the registered unit, κ 0.734 over $n = 878$ runs, mean prevalence 37.5%, which passes, so the two derivations have become one and nothing is held out.

Items reaching the gate: A1 (0.918), C6 (0.782), E2 (0.749), D4 (0.744), F1 (0.734), F13 (0.717), A4 (0.688), E1 (0.625), H12 (0.606). Items furthest below it: D3 (0.133), G2 (0.204), H7 (0.322), G7 (0.330), G6 (0.354).

Six recurring items have no variance in annotator B at all, so κ is printed degenerate in place of 0.000. We name them, because one is reported as a finding: H3, H6, D16, D10, F6 and F14 were assigned by annotator A and never once by annotator B across all 180 episodes. (Three single-observation items, H5, J5 and J6, also carry zero annotator-B prevalence but are printed one, the verdict for items this paper quotes and never counts; J5 was applied by neither annotator.) D16 is reported in Appendix F as a 9-episode reading-pass finding and carries that disclosure there.

What the aggregate does not show

Only one of the nine items that pass the gate is an item the conjunction rests on. Of the eight items in §5.3, three fall inside the band (E2 at 0.749, which passes; I1 at 0.576 and I5 at 0.591, both below), and the remaining five are too rare per step for κ to be interpretable. Two deserve naming outright. I2, which §5.3 rests on, is labelled on 5.25% of steps by annotator A and 0.90% by annotator B, a ratio of 5.8 and the largest in the conjunction, at κ 0.259 and PABAK 0.910; that is the pair read there, its κ being a prevalence artefact and not a verdict. F10, which §5.3 demotes out of the conjunction on these grounds, runs 2.63% against 0.64% at κ 0.346. On both, the annotators agree overwhelmingly about where the item does not apply and disagree about most turns where one of them says it does. The qualitative form of each is carried by verbatim spans checked mechanically against the thought of their own step, which is what does not move; the rate is what adjudication and the validity check are for.

The dominant signal is a difference in labelling density rather than a dispute about individual turns. Annotator A applied a label more often than annotator B on 82 of the 97 items, against 9 the other way and 6 equal; on some the gap is large (D3 24.1% against 3.9%, G6 19.0% against 5.6%, G2 18.9% against 7.8%). Three things plausibly drive it and they are not separable with the data in hand. (i) Context isolation differed: annotator A saw one episode per context with no inheritance, annotator B worked in multi-episode sessions with context retained inside a session. (ii) Annotator B declined to force labels and logged the residue, an unmatched list of 794 entries, where annotator A stretched to the nearest item. (iii) The codebook is under-specified on the items that fail the gate; annotator B independently recorded four explicit ambiguities, including whether "to make progress" and whether "next" count as the change token B11 requires.

The 16 items below the gate should therefore be read as a measurement of the codebook. That is what κ measures on this design, and the low half of the distribution is a list of the rules that need tightening, with the ambiguity reports naming several of them independently. None of this is evidence that the labels are correct. Observed agreement of 97.9% across 97 items is what a shared blind spot would look like, and agreement cannot see that, by construction.

Agreement between two model annotators is a figure the literature reports as reliability, and the case for reading it as more than that rests on a different comparison. The LLM-as-a-judge result a reviewer will reach for first (Zheng et al., 2023) reports high agreement, but that figure is model-versus-human on pairwise preferences, and it says nothing about model-versus-model agreement on categorical codes, which is what κ measures here. The crowd-worker comparison (Gilardi et al.,

2023) reports agreement between model runs as a reliability metric beside accuracy, which is the practice this design adopts and then declines to read as validity. *Automated Annotation with Generative AI Requires Validation* (Pangakis et al., 2023) is the direct authority for the conclusion this appendix acts on; it measures model-versus-human agreement, so it supports the conclusion without drawing the distinction between specificity and validity that the design above rests on. Two models can agree on the same wrong label, and no agreement statistic can see that; so the human instruments below, and not κ, are the only correctness evidence in this study.

THE CONJUNCTION'S ACCOUNTING, AND THE EFFECT OF REMOVING ANY ONE FINDING

The body's empirical claim rests on a conjunction of seven findings spanning eight items. The ids give eight and the independent findings seven, because `E3`/`E2` is one finding the taxonomy splits in two. The accounting is set out here so that the claim is read as existence and breadth and never as a frequency.

The two legs are separate observations, not one turn. Joined on the step, `I2` and `E2` fall on the same turn 13 times for annotator A, twice for annotator B, once for both, 19 times for either and never for the third annotator, all on one model; in 18 of those 19 the accurate table is the observation echoed back. The claim is therefore two observations, and their co-occurrence on a turn is not established. Adding `E3` changes no count. The label sheets and the script that produces these counts are in the supplementary material.

The weight rests on `I2` and `E2`, the two the claim needs rather than the two best measured, and is not spread evenly across the eight. The two descriptions do not pick the same pair: `I1` and `I5` sit inside the gate band and so have a readable κ (both apply and fail, at 0.576 and 0.591), while `I2`'s, at 3.07% prevalence, cannot be read at all. `I1`, `I5`, `I6` and `B5` corroborate, as does `A6` from outside the eight, and none of them carries the claim; `E3` is `E2`'s finding under a second id, and `H1` is carried as a metric argument, not a leg. `F10` left outright: 1.63% prevalence, κ 0.346, and not independent of `F4`, the pair being one decision point that the accounting counted as a single finding.

The conjunction is therefore not symmetric, and removing items does not degrade it evenly. Demoting `F10` removed a whole finding, because `F4` went with it. Removing any corroborating item leaves the claim standing on `I2` and `E2`, where its weight already sits; removing either of those two is what the claim does not survive.

THE DIRECTION-LEAKAGE EXPERIMENT

Blindness to the study's expected direction was tested rather than assumed. Eight fresh empty-context readers were given only the materials the second annotator receives and asked whether they could tell which way the study comes out. In the first round seven of eight could, and one wrote a substantially correct abstract from the codebook alone. The direction was carried not by the labelling rules but by the prose ranking them: a roster naming the items that "settle" the question, one item flagged as the most important in the codebook, section headings asserting their own conclusions. That layer was removed from the annotator's copy, with item definitions, boundary tests and discriminating questions unchanged, and on re-testing no reader could state the finding; what remained was an acknowledged lean, and no reader could predict the direction from the label definitions alone.

A residue is irreducible. The step-level mechanical evidence an annotator must be given (whether the board changed, how long each run of identical actions is) is by itself enough to infer the direction. Any second-annotator design over data of this kind has that property; it is not a feature of this codebook.

HUMAN LABEL ASSESSMENT: THREE INSTRUMENTS, NEVER POOLED

Two human checkers each worked a registered 40-turn draw (143 label-verdicts over 39 items and 147 over 48 items), and a purposive probe added 26 on the conjunction items neither draw could reach: 316 label-verdicts. Each sheet is the draw as generated, with every turn $\times$ item pair matching in order, no drift in any field the verdict does not occupy, and no verdict pre-filled. Each label was judged not *would I have chosen this* but *is it defensible under the codebook's rule as written*.

Verdicts are `confirmed`, `wrong` or `unclear`, and `unclear` is never pooled with `wrong`: it is a finding about the rule, not about the label.

The three instruments are never pooled into one rate. They differ in checker, in draw design, and, decisively, in which annotator the checker operated.

instrument (annotator operated)	both applied	wrong	unclear	one only	wrong
first registered draw (annotator `B`)	91	**3**	2	52	11
second registered draw (annotator `A`)	**108**	**0**	9	39	6
purposive probe (not a sample)	26	**0**	4	n/a	n/a
totals (counts, never a rate)	**225**	**3**	15	91	17

The stratum has to be defined per label, not per turn; defined per turn, the shared blind spot would read as zero. The worksheet flags each *turn* as disputed or not; but a disputed turn is disputed over *some* item and still carries every other label on that turn, including labels both annotators applied. 34 of the 86 rows on flagged-disputed turns on the first sheet, and 23 of 62 on the second, are labels both annotators applied. Splitting by turn would count those as disputed and would put the shared-blind-spot figure at zero on both sheets, a property of the turn, not of the label being judged.

A shared blind spot therefore exists, it is small, and it is confined to one instrument. Of 225 labels that both annotators independently applied to the same turn, a human judged three indefensible. Two of the three are the same item, both `B11`, both with the same reason ("the sentence is repeated, but it does not claim that anything changed"), on an item whose change-token clause annotator B had independently flagged as ambiguous, and whose κ is 0.392, in band and below the gate. A model flagged the rule, the agreement statistic flagged the rule, and a human reading the turns rejected labels built on it. That is the instrument working: a validity check returning no errors anywhere would have been much weaker evidence that it can detect any.

The two checkers differ in one respect. One returned 14 `wrong` and 2 `unclear`; the other 6 and 18, nine times as many `unclear` verdicts. The plain reading is that the codebook's author knows where the rule is silent while an outside reader has to decide anyway. Both ways of judging are legitimate and answer different questions, which is a third reason the sheets are reported separately.

The probe is the weakest of the three instruments by design. `H1` occurs on 0.70% of steps by annotator A's labelling (0.64% by annotator B's, mean 0.67%), so a random sample large enough to reach it by chance is not affordable; the probe drew 26 labels by naming seven items. It licenses counts against named denominators and no rates, estimates no prevalence, and is never pooled. Every label in it is one *both* annotators applied, which also removes the operator asymmetry: where both applied the label, there is no other annotator's label for a checker to be harder on.

Coverage is uneven and no per-model validity claim exists. Of the 316 verdicts, 111 (35%) fall on Llama-3.1-8B; Mistral-7B has 31 and was covered almost entirely by the second registered draw, and Qwen2.5-3B has a single verdict on one of the two sheets. The manuscript's coverage figure is a count of verdicts, not a claim that every model is equally checked.

THE FULL PER-ITEM TABLE

Table 5: Reliability for all 97 codeable items, both annotations complete. Each item is scored in its own unit, which the n column states: 85 items over 1,561 steps, 11 over 180 episodes, and `F1` over 878 runs. agr is observed agreement; prev A and prev B are the two annotators' prevalences in that same unit, reported separately because their difference is the dominant signal. κ is bolded where the gate applies. *paradox* marks the low-prevalence band where agreement and PABAK are the readable statistics and κ is not; *degenerate* marks the six recurring items annotator B never assigned; *one* marks single-observation items this paper quotes and never counts. Generated from the comparison output, not retyped. `I8` is the exception and must be read differently: annotator A's `I8` column is *derived* from the reading passes' own "nothing wrong here" prose, written before the item existed, rather than labelled against the item's rule. It is the induction readers' positive assertion, made blind and transcribed into annotator A's column, and it is not blind labelling against the rule annotator B used. Its κ is therefore agreement between two different procedures, a post-hoc derivation and a blind rule application, and is not an agreement statistic in the sense the other 96 rows are; the row is printed so that the derivation is disclosed rather than hidden, and nothing in this paper rests on it. In particular the positive control of Appendix H does not: its `I8` counts come from a fresh blind labelling of its own material against the item's rule, both groups by the same procedure, and neither corpus column is its comparator. That control is the first time annotator A's model labelled `I8` against its rule; annotator B and the third annotator labelled it under v5 throughout. A κ marginally below zero in this band (`G3`) means the two annotators' handful of labels never fell on the same step, which at a prevalence under half a percent is what chance predicts; it is not evidence of systematic disagreement.

item	n	agr %	prev A %	prev B %	κ	verdict
A1	1561	97.63	18.58	16.59	**0.918**	pass
A2	1561	98.46	6.15	4.61	0.849	paradox
A3	1561	99.62	3.46	3.59	0.943	paradox
A10	1561	99.04	3.46	2.88	0.844	paradox
A4	1561	93.02	12.49	13.2	**0.688**	pass
A5	1561	96.48	7.62	5.0	0.703	paradox
A6	1561	99.04	1.47	0.51	0.512	paradox
A7	1561	99.04	3.14	2.56	0.827	paradox
A8	1561	96.03	6.66	2.82	0.564	paradox
A9	1561	95.2	6.6	2.69	0.462	paradox
A11	1561	94.49	5.77	1.67	0.239	paradox
A13	1561	99.62	0.58	0.32	0.57	paradox
D13	1561	96.73	3.97	0.7	0.293	paradox
A12	1561	93.53	10.19	3.84	0.512	paradox
A16	1561	99.94	0.19	0.13	0.8	one
A17	180	82.22	28.89	11.11	**0.471**	below 0.6
H7	180	87.78	11.11	8.89	**0.322**	below 0.6
I2	1561	95.52	5.25	0.9	0.259	paradox
B7	1561	100.0	0.13	0.13	1.0	one
B8	1561	99.68	0.51	0.19	0.544	paradox
I3	1561	99.3	0.83	0.13	0.265	paradox
H3	1561	99.94	0.06	0.0	n/a	degenerate
H4	1561	99.49	0.58	0.06	0.199	one
H5	180	99.44	0.56	0.0	n/a	one
H6	1561	99.74	0.26	0.0	n/a	degenerate
H8	1561	99.87	0.19	0.06	0.5	paradox
H10	1561	99.62	0.58	0.19	0.499	paradox
H11	1561	99.87	0.06	0.06	-0.001	one
H12	180	88.33	19.44	16.67	**0.606**	pass
F16	1561	98.46	2.75	1.35	0.618	paradox
A14	1561	97.95	2.31	0.26	0.196	paradox
B1	1561	85.91	22.36	16.34	**0.551**	below 0.6
B2	1561	98.85	2.82	1.67	0.737	paradox
B3	1561	91.29	11.08	3.4	0.365	paradox
B4	1561	98.72	4.93	4.29	0.854	paradox
B5	1561	99.17	0.58	1.02	0.476	paradox
B9	1561	92.44	10.25	8.84	0.562	paradox
B10	180	99.44	3.33	2.78	0.906	paradox
B11	1561	88.21	14.93	6.34	**0.392**	below 0.6
F3	1561	97.37	5.45	3.07	0.679	paradox
F5	1561	88.79	17.04	9.93	**0.525**	below 0.6
D1	1561	77.13	50.29	40.49	**0.543**	below 0.6
D2	1561	90.13	16.72	11.21	**0.592**	below 0.6
D3	1561	77.32	24.15	3.91	**0.133**	below 0.6
D4	1561	93.08	16.85	15.31	**0.744**	pass
D6	1561	95.32	8.52	5.51	0.643	paradox
D14	1561	92.44	8.97	3.72	0.371	paradox
D5	1561	90.01	13.65	4.42	0.407	paradox
D7	1561	81.74	27.03	13.9	**0.454**	below 0.6
D16	180	88.89	11.11	0.0	n/a	degenerate
D8	1561	91.35	13.13	10.38	**0.584**	below 0.6
D9	1561	75.91	53.49	50.93	**0.518**	below 0.6
D10	1561	99.81	0.19	0.0	n/a	degenerate
D11	1561	97.18	4.16	1.86	0.52	paradox
D15	1561	99.55	0.58	0.51	0.586	paradox
D12	1561	100.0	0.19	0.19	1.0	one
C1	1561	99.68	0.58	0.51	0.704	paradox

item	n	agr %	prev A %	prev B %	κ	verdict
C2	1561	99.94	0.32	0.26	0.889	paradox
C5	1561	99.49	1.15	0.77	0.731	paradox
C3	180	95.0	5.0	7.78	0.583	paradox
C6	180	94.44	16.11	13.89	**0.782**	pass
G1	1561	92.63	8.01	0.64	0.138	paradox
G2	180	81.11	18.89	7.78	**0.204**	below 0.6
G3	1561	99.17	0.45	0.38	-0.004	paradox
G4	1561	99.94	0.32	0.26	0.889	one
G5	1561	99.87	0.13	0.26	0.666	one
G6	1561	85.46	19.03	5.64	**0.354**	below 0.6
G7	180	85.56	14.44	10.0	**0.33**	below 0.6
F13	180	89.44	23.89	25.56	**0.717**	pass
F18	1561	99.74	0.32	0.06	0.333	one
E1	1561	86.68	26.52	19.09	**0.625**	pass
E2	1561	90.52	27.67	22.68	**0.749**	pass
E3	1561	97.76	2.31	2.5	0.522	paradox
H2	1561	99.23	1.54	1.79	0.765	paradox
F1	878	87.36	43.51	31.55	**0.734**	pass
F2	1561	94.55	5.96	0.51	0.15	paradox
F4	1561	99.04	1.28	0.45	0.441	paradox
F10	1561	97.89	2.63	0.64	0.346	paradox
F6	1561	98.65	1.35	0.0	n/a	degenerate
F7	1561	93.21	7.05	1.28	0.167	paradox
F12	1561	99.81	0.26	0.06	0.399	paradox
F14	1561	97.63	2.37	0.0	n/a	degenerate
F15	1561	97.89	2.43	0.32	0.228	paradox
H1	1561	99.55	0.7	0.64	0.664	paradox
H9	1561	99.68	0.45	0.13	0.443	paradox
I1	1561	78.54	53.04	40.04	**0.576**	below 0.6
I5	1561	89.37	16.98	13.65	**0.591**	below 0.6
I6	1561	96.67	3.52	1.09	0.266	paradox
I8	1561	98.72	1.67	1.67	0.609	paradox
J1	1561	96.93	5.7	6.09	0.723	paradox
J2	1561	99.81	0.58	0.38	0.799	paradox
J3	1561	99.94	0.13	0.06	0.666	paradox
J4	1561	100.0	0.06	0.06	1.0	one
J5	1561	100.0	0.0	0.0	n/a	one
J6	1561	99.94	0.06	0.0	n/a	one
F9	1561	98.14	2.63	0.77	0.446	paradox
F8	1561	99.17	0.77	0.45	0.312	one

THE THIRD ANNOTATOR, IN FULL

The confound comes first, because it bounds everything after it. A third annotator (Gemini 3.7 Flash) labelled all 180 episodes under the same frozen v5 codebook, returning 5,888 label rows with at least one label on 1,560 of the 1,561 steps, as many as annotator A (annotator B's rows cover 1,535), a complete third annotation. It differs from annotators A and B on four axes at once (model family, agent harness, labelling prompt, and context isolation), and no agreement figure below separates them. One correction to the obvious reading, and it cuts against it: only annotator A ran the project's own fixed 809-word labelling prompt. Annotator B's was written by its operator and varied from session to session, so no single fixed prompt exists on that side; the third annotator's was operator-written, fixed across its sessions, and 139 words. An operator-written prompt is therefore what annotator B and the third annotator have in common, not a defect peculiar to the third. The codebook is not part of the confound: all three labelled under v5.

Median in-band κ with the third annotator is substantially lower than between A and B. Each pairing is computed by the same tool, on the same gate band, in each item's own unit:

pairing	items in band	pass $\kappa \geq 0.6$	median in-band κ	median agreement
A vs B (the reported pair)	25	9	**0.576**	97.89%
A vs third	20	3	**0.319**	97.37%
B vs third	18	3	**0.381**	98.59%

The three bands differ in size (25, 20 and 18) because prevalence differs by annotator, and each is computed on its own pairing; no item is held out of any row, so the three are built the same way as each other and as the figure reported in §4, but they are not medians over the same items, so they are given side by side and never as a ratio such as "half". Median observed agreement stays between

97% and 99% in all three because most items are rare, so a comparison stated as agreement would hide the drop entirely; the drop is a statement about κ. The three pairings are never pooled into a single agreement number: three pairings, three different confound structures, the same rule the three human validity instruments carry and the rules-restored arm's two conditions carry.

The items the paper rests on. `I2`, the item §5.3 places its weight on, reads κ 0.477 for A–third (prevalence 5.25% and 1.79%, observed agreement 96.41%, PABAK 0.928) and 0.470 for B–third (0.90% and 1.79%, 98.59%, 0.972) against 0.259 for the reported A–B pairing (5.25% and 0.90%, 95.52%, 0.910). All three pairings sit outside the prevalence band, so none of the three κ values is read as evidence in either direction; they are reported only to document sensitivity to the annotation pipeline, and what the three share (observed agreement above 95% and PABAK at or above 0.910 at low prevalence) is descriptive. `E2`, the other leg, reads 0.473 for A–third and 0.469 for B–third against its 0.749. Most of the other conjunction items fall, some to near zero (`I5` -0.004 in both pairings, `E3` 0.053 and 0.049), which is the expected shape of a codebook that three readers apply at three different thresholds.

`H1`. Its fall is undetermined, not measured. It reads κ 0.306 against 0.664, on a third-annotator prevalence of 0.13%, two cells against annotator A's eleven. But the third annotator wrote eleven `H1` rows, the same raw count as A, and filed nine of them against the episode where the codebook defines the item per step, so the comparison dropped them. The phenomenon was found; the filing was wrong. Rolled up to the episode, a robustness derivation in a different unit and never a replacement for the step-level figure, the item reads κ 0.613. The fall and its cause are reported together, and nothing rests on either number.

One item gains evidence. At the registered run unit the third annotator and annotator B agree on `F1` at κ 0.907 over 878 runs, the highest agreement anywhere in this study, while annotator A sits apart from both (0.734 and 0.712). `F1`'s entry says two different things about single-step runs, and annotators A and B split cleanly along that one clause; a third annotator of a different family, under a different prompt, independently lands on annotator B's side of it. That is direct evidence the disagreement is a defect in the rule text rather than carelessness in either reader.

Two silent-drop defects. Both were found on these sheets in one sitting. The smaller: the field-count gate (`csvcheck_labels.py`) fails on the third annotator's sheets with 70 malformed rows, where A and B pass with zero; an unquoted comma inside a free-text note gives the row more fields than the header, and the standard CSV reader parks the surplus silently. Nothing that a result rests on is damaged: the six key fields sit ahead of the split on all 70 rows, so zero key fields and zero quoted spans are affected and κ, prevalence and coverage all stand. What is cut is the annotator's written rationale, which only adjudication reads; recovery is lossless and is deferred, adjudication being out of scope for this version. The larger: rows filed against the wrong unit (a step-level item recorded against the episode, or the reverse) match no cell and were dropped without a warning. All three annotators do it (A 64 rows, B 44, the third 97), and it is the one that moved a conclusion, `H1` above being the case. The comparison tool now warns on any row that matches no cell.

The rule-gap list. It is a sanity check and nothing more. An earlier reading of this pass proposed that a blind rule review and a third annotator had arrived, months apart, at the same defective items. The number it rests on is real: the 15 items the blind review named as rule gaps show a median absolute prevalence gap between A and the third annotator of 10.19 pp against 1.66 pp for the other 82, a ratio of 6.14. The claim built on it is not. An absolute percentage-point gap is mechanically bounded by prevalence (an item occurring on 3% of steps cannot show a 10 pp gap whatever the readers do), and the 15 are not a random sample on that axis, their median prevalence being 16.98% against 2.98%. Restricted to the items inside the gate band, where prevalence is roughly held, the ratio falls to 1.44; and the same test on annotators A and B alone, with no third annotator involved at any point, already returns 3.72. The list therefore predicts where any two annotators diverge. That is a defensible sanity check on the review and a defensible reason for choosing those items for repair, and it is not independent confirmation. The body says the former and not the latter (§4). Two caveats belong with it: the list is not the top of the ranking, 8 of the 15 largest gaps corpus-wide falling outside it; and one of its items, `A12`, has a gap resting on a third-annotator prevalence of zero that is itself the filing artifact described above.

F CASE STUDIES AND THE REMAINING FORMS

Every count in this appendix is a reading-pass figure on the basis stated in §5: a figure written "at reading pass 3" is n of 60 and one written "across the three reading passes" is n of 180. Each item is reported with its parent where it is a split, because a parent item's decline at reading pass 3 is an artefact of its own split and reporting the decline alone would read as a finding. Episode counts across two items are unions, never sums, because one episode carries many labels, while description counts are additive; the two are different units and must not be compared across.

THE WORKED CASES THE BODY NAMES

I1, accurate reading, impossible move. Its sharpest case is `t0 mistral-7b L3 s3-s7`: five consecutive steps in which nothing the thought says about the board is false, and the emitted action is a *push* where the sentence describes a *walk*.

I6, the true fact and the opposite conclusion. In `t0 llama3.2-3b L4` "the box at (4, 2) is still blocked by the wall at (5, 2)" is true, and decisive, since docking that box means pushing it left from the very wall cell just named; the agent emits `push up` on all six steps. The stated obstacle is treated as something to push against rather than as the thing that rules the plan out.

E3, the correct sub-goal, the wrong push. In `t0 llama3.2-3b L3` the level's correct sub-goal is stated at four steps with the wrong push emitted each time.

H1, the remaining cases. Besides `t1 qwen2.5-3b L6`, which the body quotes, `t1 llama3.2-3b L6` is a second episode whose only board-changing action is the one that ruins it; and in `t0 llama3.1-8b L6 s2` the push jams the box against a wall where the recovery cell is that wall.

The same dead cell, reached twice. Figure 3 prints the pair the codebook singles out under `H1`. Level 2's box has to leave the row-2 corridor at `(6, 2)`, because `(6, 3)` is the only opening in row 3, and `(7, 2)` is walled at `(8, 2)`, `(7, 1)` and `(7, 3)`, so a box pushed one cell past the exit can never be moved again by any sequence of actions. Two models walk a box into that cell in the same pass, and the conjunction is what makes the pair worth printing: board reading is at its most accurate in the episodes where the level is most thoroughly ruined.

The cell is also where `H9` lives. Six of the seven episodes carrying `H9` anywhere in this study end with the box standing on `(7, 2)`: four distinct models across four of the six conditions. The figure prints two of the six; the seventh is `t1 mistral-7b L1`, on a different level. The item is close to a description of one trap, which is a fact about the level set as much as about the models, and it is the reason `H9` is reported as corroboration for `H1` and never as an independent frequency.

I5'S VERDICT SPLIT, AND WHAT AN "UNCLEAR" VERDICT MEANS

On the first author's registered sheet all four `I5` verdicts came back unclear, none confirmed and none wrong, the only item in that sheet to do so, and the claim is scoped to that sheet. Pooled across the two registered sheets it is 2 confirmed and 4 unclear of 6; the other checker's sheet confirmed both. No label was judged wrong on either. The form is present wherever the step-level condition is met, and what is unsound is the unit, an episode-level description sitting over a step-level coding condition. An `I5` episode count is therefore a quantity the rule does not pin down, and the body leans on none. `B5` shares the flaw. `I1` and `I6` carry the family claim, both checked without a wrong verdict.

THE WITHDRAWN B5 INPUT-REPRESENTATION CAVEAT, AND WHAT REPLACED IT

An earlier version of this work carried the caveat that `B5` (an accurate readback beside a claim the same table contradicts) might be an artefact of how the board is rendered to the agent rather than anything the agent does. It is withdrawn on mechanical verification of the rendered input: zero `Unknown` cells across all 1,561 steps, and 65 steps carrying a `Worker on Dock` row across 12 episodes and five of the six conditions, every one of them agreeing with the logged worker position, with `Dock`, `Box`, `Box on Dock` and `Worker on Dock` emitted as four distinct strings. The

`t0 llama3.1-8b L2 s6`

Shown: `Worker (5, 2), Box (6, 2), Dock (5, 7)`

"To avoid deadlocks and efficiently push the box towards the target, I should move the worker to a position where it can push the box directly towards the target. The current position of the worker at (5, 2) and the box at (6, 2) suggests that moving the worker to (4, 2) and pushing the box to (5, 2) would be a more direct approach. This will create space for further movement and help the worker push the box towards the target without getting stuck in a deadlock."

Emitted `push right`; the box goes to `(7, 2)`. 5 turns follow, 5 of them repeating the avoidance clause.

`t0 qwen2.5-7b L2 s9`

Shown: `Worker (5, 2), Box (6, 2), Dock (5, 7)`

"The current board state shows the worker at (5, 2) with a box at (6, 2). The next step is to continue moving the box towards the dock at (5, 7). The worker should move right to (6, 2) to push the box to (7, 2), then continue moving right to (7, 2) to push the box to (8, 2), and so on. This strategy ensures the box progresses towards its target without causing any deadlocks."

Emitted `push right`; the box goes to `(7, 2)`. 15 turns follow, 12 of them repeating the avoidance clause.

Figure 3: `H9` twice on one cell, execution 1, temperature 0. Both agents are shown the same board state and read it back correctly (coded `I1`, the opening position statements are correct and the move built from them is impossible); both emit `push right`, both put the box on `(7, 2)`, and both assert in that same thought that a deadlock is being avoided; each turn carries `H1` and `H9`. Neither episode ever registers the loss. Across the 20 turns that follow the two losing pushes, 17 mention a deadlock, 22 times, and every one of those mentions is on the avoidance side of the negation. Where the two differ: `qwen2.5-7b` names `(7, 2)` as its own destination and plans to push on into the wall at `(8, 2)`, while `llama3.1-8b`'s thought describes a *leftward* push and emits a rightward one (`E1`, `H2`); what the two share is the readback, not the plan. The spans are verbatim, the strongest evidence class in the paper.

rendering is unambiguous at every step where the item fires, so the contradiction is in the agent's own text. The caveat is recorded here because it was live in a circulated version.

`F4` AND `F10`: DETECTION IS NOT THE MISSING PIECE

Both items sit outside the conjunction (§5.3) and are kept as evidence. `F4` and `F10` are the same decision point with two different answers: under `F4` a previous action's failure is noticed and the response is a shallow direction swap rather than a diagnosis; under `F10` the self-diagnosis is specifically correct (it names the wall, or its own thought/action mismatch, or its lack of progress) and the corrective chosen is equally impossible. Detection is present in both items, since `F4`'s own definition has the failure noticed, so it cannot be what separates them. We do not read the two items' episode counts against each other: at 0.86% mean prevalence for `F4` and 1.63% for `F10` (κ 0.441 and 0.346) a difference between them is not a quantity this instrument resolves. In `t1 llama3.1-8b L3 s3`, "moving left was not effective due to the obstruction by a wall" is correct, and the same thought picks a direction that is also a wall one cell away. These cases show detection without a successful transition from the detected mismatch to the next action.

THE SECOND `I2` CASE, AND `B5` IN FULL

The transcription forms separate by outcome and by frequency into four different items. Transcription correct is `I2` at 14 episodes across the three reading passes, after the `J1` exclusion; stale is `B8` at 4, incomplete is `I3` at 2, and fabricated is `B7` at 1, a single observation, which this paper quotes and never counts. The correct-transcription case dominates, and it carries the argument.

Beyond the case in the body, `t0 llama3.2-3b L1` lists `17 | Empty | (2, 2)` correctly in in-thought tables under a heading of the model's own making at `s2`, `s3` and `s5` in the first and third executions, and at `s2`, `s3` and `s6` in the second (whose `s5` table arrives inside an echoed log block and is therefore counted as `J1`, not `I2`), while the prose beside the tables calls that same cell a wall. The wall belief itself runs all six steps of the episode in the first and third executions and five of six in the second. In `t0 llama3.2-3b L6` the tables at `s1`, `s3` and `s6` list `40 | Box on`

Dock | (3, 6) and 21 | Box | (2, 3) correctly while the prose calls (2,3) a wall and never mentions the docked box at all.

The B5 board of the body's case panel is t0 qwen2.5-7b L4 s6, with info.boxes_on_target = 0. The turn occurs in two of the three temperature-0 executions, opening at s6 with the same sentence word for word; the first execution never places the worker on a dock in this episode. In one of the two (an 18-step episode) the belief persists verbatim through s7–s9 and recurs at s10, s14 and s16–s18 between stretches of correct readings, never explicitly withdrawn; in the other (10 steps) it holds from s6 to s9. At temperature 1 the same false-arrival belief appears at s5 of one execution, as "at (3, 2) on a dock with one box already pushed to its target at (4, 2)", not verbatim; the other two temperature-1 executions carry neither the sentence nor the belief. A companion observation on the same input: in t1 llama3.1-8b L1 the board from s2 onward carries 9 | Worker on Dock | (1, 1) and not one of the seven following thoughts mentions that dock; every one tracks the other dock, at (2,4). That companion form recurs in 6 episodes at reading pass 3, three models, both temperatures; the count is its own, not B5's, which the unit defect leaves unreported.

Robustness of the I1/I6 recurrence

The body reports I1 in 13 episodes at reading pass 3 beside 7 at reading pass 2, and I6 in 5 beside 1, and reads them as independent recurrence in replicate conditions, never as growth: the two reading passes are different executions (§5.5) read by different readers, so a cross-pass rise is confounded by both and no claim in this paper rests on one; the immunisation argument of Appendix C covers the twelve split items and explicitly not growth. Measured in descriptions rather than episodes, I1 runs 14 at reading pass 3 beside 9 at reading pass 2, and I6 6 beside 1, so the recurrence reading does not depend on which unit is used; the body reports episodes throughout. I5's episode count is not reported anywhere in the paper, because its unit is unsound (above).

Other recurring forms

- **D14.** The push destination is a cell another box occupies. 16 episodes, both temperatures, five models. With its parents: D4 and D6 fall by 26 descriptions (70 → 44) in the same sitting D14 takes 23; in distinct episodes the parents go 27 → 24, a fall of 3, while D14 takes 16.
- **B11.** The thought restated word for word on an unchanged board, carrying change-tokens. 16 episodes, both temperatures, all five models. With F2: 40 → 3 descriptions as B11 takes 19; in episodes, F2 24 → 3 and B11 16. This count carries a known over-application. The validity check judged 2 of 4 sampled B11 labels wrong, both because the thought was repeated without claiming anything had changed, and the change-token clause is the one annotator B independently reported as ambiguous. The rule is under-specified, so 16 episodes is an upper bound on this item and not a measurement. Appendix C's B11 result is unaffected: that is a documentary claim about a form an earlier pass described and filed under a neighbour, not a claim about how often the label applies.
- **D16.** A sustained axis relabelling that then corrupts the agent's own blocking checks, so it concludes a cell is free because it checked the wrong axis. 9 episodes. With D7: 73 → 48 descriptions as D16 takes 25; in episodes, D7 35 → 25 and D16 9. D16 is one of the six recurring items annotator B never assigned once in 180 episodes, so on the annotation its count could be anywhere from 0 to 20 episodes; it is the only finding this paper reports whose count has a floor of zero. The reading-pass observation stands on its quoted spans; the reader should know that the second annotation does not corroborate it at all.
- **D13.** A diagonally placed piece asserted to be adjacent. 17 episodes at reading pass 3, both temperatures, all five models. Distinct from D1, which records only the geometry.
- **A12.** A non-existent box given a persistent identity and planned around. t1 mistral-7b L2 reasons about "Box 2" for eight consecutive steps on a board with exactly one box. Its sub-form A12/offset is sharper: the phantom sits exactly one row or column from a real box/dock pair and coexists with it as a separate object, which points at the coordinate table being read one line out, a different process from free invention.

- **A17 against F15.** A *real* object dropped from the reasoning and never mentioned again versus a *fabricated* one dropped after a single step. 9 against 2 episodes at reading pass 3, the only basis the two share: A17 was split out at pass 3 and so has no pass-2 count, while F15 adds one further episode at pass 2. Both recur, in opposite directions: whether an object survives in the reasoning is not tied to whether it exists on the board.
- **C6.** A box one legal push from a dock is never mentioned in the episode. 4 episodes. It bears directly on the central claim: the agent is not failing because the puzzle is hard.
- **B9.** A multi-step, internally consistent fiction inherited forward as established fact. 8 episodes. t0 qwen2.5-7b L3 runs a five-step story about a board that does not exist, concludes there are no more boxes to push, and at s7 snaps back to reading the real board correctly with no transition. The board it was shown never changed.
- **F16 against F13.** Exact inverses. The worker declared stuck, cornered or trapped *with open legal moves* against the worker's *real* enclosure going unregistered: 6 against 1 episodes at reading pass 3, again the shared basis, since F16 is a pass-3 item, and F13 adds one further episode at pass 2. They co-occur in the corpus and point opposite ways: the agent's report of its own enclosure tracks the board in neither direction.
- **G6.** The target commitment switches between consecutive steps with nothing on the board having changed. 7 episodes. The opposite failure of the same feedback loop as B11/F2: sameness where there should be change, against change where there is no reason for change.
- **G7.** A level-scale sealed region makes every plan in the episode impossible and is never registered. 8 episodes. In t1 llama3.1-8b L5, column 4 is solid wall on rows 1–3, so no box in the starting pocket can move rightwards at all, and every plan in the episode is a rightward push.
- **A14.** An object named as its own obstruction, so the obstruction has no separate referent. 4 episodes, both temperatures, two model families.
- **D15.** The box treated as cargo: picked up, received, carried, unloaded, while the push often survives in the same sentence. 3 episodes.
- **B2.** Fabricated progress present in the step-1 thought, before any action exists, so it cannot be accumulated error. See Appendix H on why this is *elicited* rather than injected.
- **B10.** The level declared complete, with no more boxes to push, while every box is off dock. It terminates planning, so where in the episode it lands is part of the observation.

The non-recurrences are reported too, because a taxonomy that only ever grows is not being tested. C2 (a plan to push a box off its dock) went to zero at reading pass 3 after being promoted at reading pass 2, as did H3, F4 and F12. But six items silent at reading pass 2 reappeared at reading pass 3: C1, D10, G1, H6, H8, I3. A pass of zero recurrence is not evidence a form is gone, and that cuts both ways. F8, H4 and H5 are kept as quoted single observations.

THE GENERATION LAYER, ITEM BY ITEM

Family J is a layer, not a mechanism, and is counted apart from everything above. J1 is the harness's own logging scaffold emitted inside the thought: timestamped turn-hash blocks, action and thought headings, fused lines. 4 episodes at reading pass 2, 2 at reading pass 3. J2 (runaway verbatim template repetition cut off mid-sentence) and J3 (no action emitted at all, the turn ending at a bare prompt token) were both promoted from single observation to recurring at reading pass 3. J4 (a stray non-language fragment) and J5 (two response formats alternating on a fixed parity) are single observations with zero recurrence at reading pass 3.

J1 is the one that matters for reading §5: a thought that re-emits the harness's log (## Turn Hash, a timestamp, ### Obs, then the board) contains a correct board table it never reasoned out. 19 of the 106 steps whose thought carries a board table are of this kind; they are counted here and excluded from I2, and the supplementary material ships the per-step flags behind this split (reliability/scaffold_echo_flags.csv: board-table-present and scaffold-markers-present for every step), so the partition can be checked. t1 qwen2.5-3b L3 s9 is the clearest case, and it is instructive because its prose is also wrong about the board it just echoed:

it opens "The Worker is currently at (7, 5) with Box 48 at (7, 6)" where the echoed table gives row 22 as `Worker (5, 2)` and rows 48 and 56 as `Wall`. We report it as `J1` because the board in that thought is the log, not the model's own account of the board. `J6` is a fabricated future turn in the same log format, a single observation, `t1 llama3.2-3b L6 s7`: "###Action move down ###Thought Reviewing the current board state, I notice that the worker moved down to (1, 3) clearing the blockade." (1,3) is a wall, and the action for that step had not been emitted when the thought was produced. `F9` is the prompt's own text emitted in place of reasoning.

`J5` earned its place by failing. Its own stated test for reading pass 3, a format regularity on a fixed parity, did not recur, while the content form of the same shape (two thoughts alternating verbatim on a strict parity while the board never changes) did. A format regularity and a content regularity of the same shape recur independently and at different rates, so the generation items were not merged into one.

G Related-work enumeration

The failure-analysis cluster. *Failure as a Process* (Zhao et al., 2026), *Beyond the Leaderboard* (Albayaydh et al., 2026), *The Long-Horizon Task Mirage* (Wang et al., 2026a), *AgentRx* (Barke et al., 2026), *Where LLM Agents Fail and Learn From Failures* (Zhu et al., 2025), *Beyond Resolution Rates* (Mehtiyev & Assunção, 2026), *Understanding Code Agent Behaviour* (Majgaonkar et al., 2025) and *MemFail* (Garg et al., 2026) are coding, SWE, CLI, tool-use, multi-agent or memory systems. None targets games or puzzles.

The grid and spatial cluster. These are benchmarks rather than analyses of the failing turn: SokoBench (Monti et al., 2026), *lmgame-Bench* (Hu et al., 2025), grid-game competitions (Topsakal et al., 2024), *CubeBench* (Gao et al., 2025), *GRASP* (Tang & Kejriwal, 2024) and *DeepPlanning* (Zhang et al., 2026). *TopoBench* (Maniparambil et al., 2026) adds an error taxonomy over chain-of-thought traces, on puzzles rather than an agent's turns. The closest method ancestor is *Step-by-Step Reasoning to Solve Grid Puzzles* (Tyagi et al., 2024), a reasoning-error analysis on grid puzzles, but single-shot and prompt-level, pre-agent, with no open-model scale study.

The planning line. It establishes that language models plan badly; it does not record what a turn does instead. PlanBench (Valmeekam et al., 2023a) and its companion result paper (Valmeekam et al., 2023b) score plan generation on classical planning domains; the two share substantial text and are one line of evidence, not two. The o1 follow-up (Valmeekam et al., 2024) is why "cannot plan" needs qualifying: reasoning models improve sharply on standard Blocksworld and degrade as required plan length grows, so degradation with plan length is the durable part of the finding. The LLM-Modulo paper (Kambhampati et al., 2024) argues that models can neither plan nor self-verify, and positions them as helping planning from inside an LLM-Modulo framework. *The Illusion of Thinking* (Shojaee et al., 2025) reports frontier reasoning models collapsing past a complexity threshold, in single-shot generation with no environment in the loop, and is cited here beside its rebuttal (Lawsen, 2025), which attributes much of the Tower-of-Hanoi collapse to the output-token budget and finds some River Crossing instances unsolvable as posed. Our one point of contact is the ladder of §5.2: 0 of the 60 episodes at or above four optimal pushes, on one board per rung, a result that bounds that ladder and states no general threshold. What none of this line attempts, and this paper does, is to read what an agent that cannot plan does on each turn while failing, with the environment in the loop, on open 3–8B checkpoints.

The harness we build on. *lmgame-Bench* (Hu et al., 2025) reports frontier and proprietary headline results, frames Sokoban around *visual* perception, and explicitly advertises techniques to mitigate prompt variance, which is useful context for the text-path confound controlled in Appendix H.

Agentic game benchmarks. They are the frame this paper sits inside, not the question it answers. BALROG (Paglieri et al., 2024) scores agentic LLM and VLM reasoning by progression across a suite of games, with no Sokoban among them; *lmgame-Bench* (Hu et al., 2025), above, is the same genre. A progression score says that an agent failed and how far in; it does not say what the failure was, and on agents that get nowhere, as ours do in all six conditions, it would return six identical zeroes. That is the complement we supply. The benchmark ranks, and the turn-level record describes what the ranked agents did.

Internal-mechanism work. It is a different method. *Fragile Reasoning* (Han et al., 2026) and *Contrastive Attribution* (Tan et al., 2026) diagnose via activations and patching. Ours is behavioural trace analysis, which is why we are careful with the word *mechanism*. Representation sensitivity (Martorell, 2025) bears on the normalisation described in Appendix H.

Faithfulness and self-correction. These are the nearest lines on this paper's finding, and the finding extends them rather than replicating them. The unfaithful-explanations result (Turpin et al., 2023) shows a model's stated reasoning omitting the feature that actually drove its answer; the reasoning and the answer stay consistent with each other, and the unfaithfulness is visible only against a cause the design hides. *Measuring Faithfulness in Chain-of-Thought Reasoning* (Lanham et al., 2023), a preprint, asks instead whether the answer causally depends on the reasoning at all, by truncating and corrupting it. `E2` and `E3` record something neither design produces: an overt contradiction inside one turn, between the move the thought describes and the action string the same message emits (Appendix F), legible with no hidden cause and no intervention. Two caveats travel with that: the prompt's own worked example may account for a share of `E2` that nothing here bounds (§7), and no causal dependence of action on thought is measured here. On self-correction, *Large Language Models Cannot Self-Correct Reasoning Yet* (Huang et al., 2023) concerns *intrinsic* correction, without external feedback, on single-shot question answering, and the LLM-Modulo paper (Kambhampati et al., 2024) says the same of self-verification. Our agents do receive external feedback, since the next observation shows the board unchanged, and under `F4` they notice it and under `F10` diagnose it correctly, then choose a corrective that is equally impossible (Appendix F). That complicates the intrinsic result rather than replicating it.

Multi-agent failure attribution. This is adjacent but not ours. MAST (Cemri et al., 2025), TRAIL (Deshpande et al., 2025), Who&When (Zhang et al., 2025b), AgenTracer (Zhang et al., 2025a), StepFinder (Zhu et al., 2026) and *Seeing the Whole Elephant* (Chen et al., 2026b) attribute failure across agents; we are single-agent in a game.

The harness normalisation. HarnessFix (Chen et al., 2026a) shows harness flaws are systematic across all seven of its layers, and its taxonomy independently names our coordinate mismatch a *Context-layer* flaw. Different domain, goal and method; we use it as a one-line authority.

H LIMITATIONS IN FULL

A SUBSET OF TURNS IS NOT BLIND TO THE FIRST ANNOTATOR'S LABELS

The codebook teaches each item with worked examples drawn from the corpus, and those examples name a model, a level and usually a step. About forty such citations appear, thirty-one naming a specific turn and five covering a run of turns; only six give the execution number, so most match the same turn in each of the three executions. So while the second annotator is blind to the first annotator's label sheet, it is not blind to the labels attached to roughly fifty of the 1,561 turns, around 3% by the narrowest count, and more if a citation is resolved across executions. We kept the examples deliberately: they are the fastest way to separate the item pairs the codebook itself identifies as most confusable. The mitigation is disclosure. An annotator who recognises a turn is instructed to label it as they see it and mark the row, and marked rows can be excluded, but that depends on the annotator noticing the recognition, which we cannot verify. Agreement on that subset is therefore not fully independent evidence.

WHICH MODELS DID THE READING AND THE ANNOTATING

Every automated judgement in this study (the three induction reading passes, the three corpus annotations and the positive control's labelling) was made by a large language model, and each is named here or where it is reported. Naming them is not an anonymity cost: these are commercial models used by many groups. Language models were also used, under author direction, in drafting and revising this manuscript and in writing the analysis scripts; every number in the text is derived by a named script and checked by the audit tooling the supplementary material includes, and the authors take responsibility for the content. This statement is made under the ICLR 2027 AI policy for authors.

	induction readers (reading passes 1–3)	annotators
model	Claude Opus 5 (`claude-opus-5`)	**A**: Claude Opus 5 (`claude-opus-5`). **B**: GPT-5, via the Codex coding agent
version	access date 30 July 2026, after each execution's collection completed	**A**: access dates 17–18 August 2026. **B**: exact version string not exposed by the harness and not recoverable; same reported family throughout, per-session snapshot continuity unverifiable. Access dates 3–16 August 2026
harness	empty-context agents dispatched from a coding agent, one per batch, given the reader brief verbatim plus one trace path and one output path	**A**: a coding agent's task tool, model pinned explicitly on every call so the fleet could not drift with the session. **B**: the Codex desktop coding agent, working against local files, not a web chat interface
isolation	one batch per fresh context; no checklist, no rubric, no tracker, no other batch's output	**A**: one episode per fresh context. **B**: multi-episode sessions (see below)
sampling	not exposed by the harness	not exposed by either harness

The readers and annotator A share an underlying model, and this bounds what the blind passes establish. The codebook was induced from those passes, so an annotator on the same model applying that codebook inherits the readers' framing through the rubric itself. The fresh-context design removes *session* contamination (no reader or annotator saw the draft, the decision record, the mapping files or another annotator's labels), and it removes nothing else. This is a mitigation, not blindness, and no claim in this paper rests on treating annotator A as naive to the taxonomy's authorship. Annotator `B` is a different family, which is what the codebook's own protocol asks for. The one property we can verify mechanically is that the codebook contains zero episode *file* identifiers (no filename, replicate number or `episode_id` string), while its worked examples do name model $\times$ level $\times$ step turns, as disclosed above.

Prompts. The reader brief and annotator A's per-episode prompt were recorded verbatim at the time, annotator A's during the run itself, and both ship verbatim in the supplementary bundle: neither contains a tool or role name, so nothing in them needed redaction. Annotator B's prompts were written per session by the operator and varied; the one the operator's record preserves exactly, the final session's continuation prompt, ships with that caveat stated in the file. The surrounding methodology records, which do carry tool and role names, are released with the code and the trace corpus on de-anonymisation.

THE TWO ANNOTATORS WERE NOT MATCHED ON CONTEXT ISOLATION

	annotator `A` (Claude Opus 5)	annotator `B` (GPT-5, Codex)
harness	coding-agent subagents	a second coding agent, on local files
unit of context	**one episode per fresh context**	**multi-episode sessions** (10, later 3)
carry-over	none between episodes	retained across episodes within a session
sessions	180 dispatches	$\sim$63 sessions over two weeks

Annotator A's isolation was imposed deliberately, on the reasoning that an annotator who has read three episodes carries the first into the third. Annotator B did not have it, mitigated it by judging each episode turn by turn, and reduced batch size partway through. But the condition is not matched, and κ between the two annotations therefore reflects model, harness and context window together and cannot attribute the difference to any one of them. Every κ reported here is agreement between two annotation pipelines, and the prompt is one of the things that differs between them; only annotator A ran the project's own fixed prompt. So these figures bound the codebook's specificity as delivered by a particular pipeline, not the specificity of the codebook text on its own. That is a further limit beyond the one §4 already states, that κ measures specificity rather than validity, because here the specificity itself is measured through a pipeline.

This cuts both ways, and neither annotator is offered as the reference. Annotator A's context isolation is cleaner; annotator A's zero quote-failure rate across 10,996 rows is also partly a product of per-episode scripted self-checking that annotator B was neither asked for nor equipped with, so comparing that zero against annotator B's 20 defects in 7,100 rows is not a like-for-like quality comparison and is not offered as one. Neither is sampling reproducible: neither harness exposes temperature, `top_p`, seed or maximum output tokens, so neither annotation is bit-reproducible. One naming collision should be flagged: the `temp0` and `temp1` labels in this corpus are the decoding settings used when the Sokoban transcripts were generated; they have nothing to do with the sampling temperature of either annotation session, which is unknown for both.

Each human checker operated one of the two annotators

Neither human checker is independent of the labels they judged, and the two are non-independent in opposite directions: one checker operated annotator B and the other annotator A. In this study, independent human validators were not available, because the annotators are language models driven by people and the people available to check the output are the people who drove them. The size of the effect can be read off the sheets.

On a turn the annotators disagreed about, a checker sees labels exactly one annotator applied. If a checker is protecting the annotator they operated, they will mark the other annotator's labels wrong more often. The first checker: 10/43 = 23.3% on the other annotator's labels against 1/9 = 11.1% on their own. The second: 1/5 = 20.0% against 5/34 = 14.7%. Both lean the same way, and that is what an operator effect would look like. It is also not what the innocent explanation predicts: annotator A applies a label more often than `B` on 82 of 97 items, so if `A` simply over-applies, both checkers should find `A`'s labels worse. One did; the other found the reverse.

It settles nothing, because one of those cells holds five verdicts. A single verdict moves that 20.0% to 0% or to 40%. The direction is what a bias would produce; it is also what sampling noise at $n = 5$ produces. We deliberately compute no test statistic on them, because a p-value here would imply a discrimination the data does not support. The absolute counts do not change the reading. The second checker rejected five of their own annotator's labels and one of the other's, each with a written re-derivation; but five and one are the numerators of 5 of 34 and 1 of 5, so they cannot point a different way from the rates. What those five rejections establish is narrower: whatever protection operated on that sheet was not absolute. Both sheets are therefore reported as they stand, and the body makes no inferential claim from them.

The part of the validity result that carries the headline is structurally immune to this. Where *both* annotators applied the same label there is no "other annotator's" label for a checker to be harder on, so the 225-label figure inherits none of the asymmetry, which is also why the conjunction probe was drawn agreed-only. The two sheets remain worth more together than either alone: their non-independence runs opposite ways, so a label class cleared by both is the strongest validity evidence this design can produce.

The action semantics, and why they are not a harness defect

Read against the environment's own implementation: `push <dir>` requires a box adjacent and floor-or-target beyond it, otherwise nothing happens and there is no fallback to a plain move; `<dir>` requires floor-or-target adjacent, otherwise nothing happens and there is no fallback to a push. The source comments say so outright: "if a push action is chosen, it either pushes or does nothing" and "for now, a failed push is just a failed push".

The agent is told the preconditions in its own prompt, though not the failure semantics. The reasoning module's prompt states, in three bullets, that it can *move* onto empty floor or docks and *push* a box if the space beyond it is empty, and to avoid deadlocks. It states one precondition and not the other. The far side is required to be empty, in words. Adjacency is never stated as a precondition: the only mention of it in the live prompt is our own coordinate block, which presupposes adjacency and says nothing about a push with no box adjacent. Nor is the consequence of violating either precondition ever stated.

The claim therefore has to be split, and only one half survives. A push whose far side is blocked violates a rule the agent was given in words. A push at a box the worker is not adjacent to (item

D1, one of the highest-prevalence items in the corpus at 50.3% / 40.5% of steps across the two annotations) violates a rule the prompt only ever *implied*. We say the former and not the latter, and D1 carries this disclosure wherever it is reported. A related disclosure: the rules the agent receives never say that walls are immovable, so D11 (a mechanic the game does not have, invented and used as the plan) rests on a rule the prompt implies rather than states, and is reported with that attached. It touches none of the seven findings of the conjunction.

Three consequences follow. (i) No-op counts broken down by action type are reportable provided the semantics are stated wherever such a figure appears, or a reader will assume push degrades to a move. It does not. (ii) We do not code the no-op as the failure; we code the claim that produced it. (iii) We did not change the action semantics; the corpus was collected under them.

THE PROMPT'S ONLY WORKED EXAMPLE DEMONSTRATES THE E2 MISMATCH

Unlike the items above, this touches a finding in the conjunction. The reasoning prompt the agent receives on every turn closes with one worked example: a thought reading *"Moving right to get behind the box"* paired with the action *"push up"*. The thought describes a plain move; the emitted action is a push, in a different direction. That is the pattern E2 codes, so the agent's only demonstration of the required output format is a demonstration of the failure.

Provenance. It decides how this is read. The example is upstream's, not ours: the prompt file is unmodified at commit 996d848. It is live: the reasoning module formats this template with variable substitution only, so the example is sent verbatim on every one of the 1,561 steps, and it survives our representation normalisation, which rewrites only the board-format sentence.

The harness does contain a prompt that states the rules; harness mode simply cannot reach it. base_module, used when harness: false, states the adjacency requirement explicitly, states that a box cannot be pushed into a wall or another box, gives the full character legend, and carries a *Key Strategies* section. reasoning_module, used when harness: true, has none of those four and is less than half the length (1,370 characters against 2,853). The two are mutually exclusive by construction, no game in the repository ships a custom prompt, and harness: true is the shipped upstream default for Sokoban, as it is for 9 of the 11 games. So there is no configuration of the published harness under which a harness-mode Sokoban agent is told the adjacency rule, the wall rule, or the legend.

The obvious defence, that the pipeline compensates for the thinner prompt, does not hold, on two independent checks. First, it is not what the repository does elsewhere: seven of the eleven games carry a rules block in their *harness-mode* prompt, so harness mode is where this repository puts its rules; Sokoban's block is simply strictly weaker than its own base_module's. Second, in text mode the pipeline supplies nothing rule-bearing at all: the perception module makes no API call, and the reflection prompt asks about state changes without restating any rule. At step 1 all four pipeline channels are null in our traces, so on the opening turn a harness-mode agent holds strictly less than an un-harnessed one would.

Sokoban is an outlier in this repository, and that is measurable. Across the eleven games the ratio of reasoning-prompt length to base-prompt length runs from 0.48 to 2.11; nine of eleven give the harness-mode agent *more* text and a tenth is near parity at 0.91. Sokoban's ratio is 0.48, the only value below 0.9. And while eight of the eleven harness-mode prompts carry a worked example, so having one is ordinary, Sokoban's is the only example in the repository whose thought and action describe different operations.

Bounding the exposure, first measure. A lexical scan for the example's own phrasing returns 143 of 1,561 steps in 41 of 180 episodes, in all five models, and 75 of those 143 emit a push while the thought describes a move to get into position, which is the example's pattern reproduced. Against E2's prevalence of 27.7% / 22.7% across the two annotations, the echoing subset is on the order of a fifth of the item. The measure is lexical, not semantic, and is neither a lower nor an upper bound on priming.

Bounding the exposure, second measure. This one asks which illegal actions the missing rule can and cannot explain. Recomputed mechanically over all 180 episodes from the pre-action board: push with no box adjacent in the pushed direction, 786 of 926 pushes (84.9%), where the violated rule is not stated in the live prompt; push with a box adjacent but the far side blocked, 78 of 926

(8.4%), a stated rule; legal `push`, 62 of 926 (6.7%); plain move into a wall or a box, 256 of 445 moves (57.5%), a stated rule. These denominators partition the corpus exactly: 1,561 steps = 180 step-1 turns (excluded here, because the logged observation is post-action, so a step-1 action has no logged pre-action board to score against) + 926 pushes + 445 plain moves + 10 steps that emit no directional action at all (five `idle`, two `no-op`, two `none`, one runaway string; the non-emissions of family `J`). The large number is the one reached by the objection the agent cannot be faulted for a precondition the prompt never stated, and we do not defend it as a model failure. The rest is not reachable by that objection at all: more than half of all plain moves are attempts to walk into a wall or a box, violating a rule stated plainly in the same four-line block the agent receives every turn, and 105 of 180 episodes contain at least one violation of an explicitly stated rule. An agent that cannot follow the preconditions it was given is not exonerated by the preconditions it was not.

The conjunction does not route through the illegal-action items at all. `I2`, `B5` and `A6` turn on a board transcribed correctly in the model's own output beside prose that contradicts it; `H1` and `H9` turn on legal, board-changing pushes that ruin the level. None is a claim about whether the agent knew a precondition. The objection narrows `D1` and a share of `E2`; it does not reach the paper's result, and the conjunction's seven findings stand with `E2` discounted outright, because `E3` carries that finding on its own. What does not stand intact is the spine of §5.3: `E2` is one of its two legs, so discounting it leaves `I2` carrying the empirical claim with `I1`, `I5`, `I6`, `B5` and `A6` corroborating.

THE RULES-RESTORED ARM, AND WHAT IT DOES AND DOES NOT ESTABLISH

We ran the arm that bears on this. The `base_module` rules were restored verbatim into `reasoning_module`, everything else held constant, as a pre-registered single-variable manipulation: control and treatment differ by exactly 734 characters of rules text and nothing else, asserted as a literal diff; both arms of each model were drawn from one served instance, across the same five models and six levels, 30 episodes per arm. The decision rule was fixed in writing before four of the five models had been run.

model	control	rules restored	change	plain moves, treatment	n steps
qwen2.5-3b	29.2%	73.5%	+44.3 pp	**0.0%**	34
llama3.2-3b	50.0%	58.8%	+8.8 pp	14.7%	34
mistral-7b	70.7%	57.6%	−13.1 pp	18.6%	59
qwen2.5-7b	68.1%	35.3%	−32.8 pp	51.5%	68
llama3.1-8b	59.5%	26.7%	−32.9 pp	62.9%	105

No-adjacent-box pushes per analysed step, per model. Per-model rows are mandatory here and a pooled column would describe no model in the study. The change column is computed from the underlying counts and then rounded, not from the rounded percentages, so llama3.1-8b reads -32.9 pp ($25/42 = 59.52\%$ against $28/105 = 26.67\%$, a difference of 32.86) where the displayed figures would subtract to 32.8. All five values come from `r6_compare.py -per-model`.

The last two columns should be read together, because they are why we do not offer this direction as a result. The two models where no-adjacent-box pushes *rise* are the two whose plain-move rate in the treatment arm is negligible; all three models with a healthy plain-move rate go the other way. The ordering is perfect across all five models: not one cell separates a change in what the agent emits from a change in how often it emits an illegal push. Because the measure's denominator is analysed steps rather than pushes, a model that stops emitting plain moves fills those steps with pushes, and the per-step rate of no-adjacent-box pushes rises with the push share. So the direction column is indistinguishable from a composition artefact on this matrix, and we report it as a disclosed non-result. Higher in two models, lower in three, spanning -33 to $+44$ pp, sign test $p = 1.00$.

One model changed qualitatively, and it does not support the reading it first suggested. Qwen2.5-3B stopped emitting plain moves entirely under the restored rules while its thoughts went on describing repositioning, which looks like a clean dissociation of plan from action. It is not safe to read it that way: scoring every typed coordinate claim against the board the agent actually saw (validated against a shuffled-board control, which separates by 65–72 pp), its board-reading accuracy in that arm is 52.2%, the lowest of any cell in the study, below both its own control arm (59.8%) and its corpus behaviour (70.3%). Its action channel and its reasoning channel degraded together.

Three limitations of this arm. (i) It is 6 episodes per model per arm, a single execution, and §5.5 shows that re-running an identical temperature-0 configuration sends roughly half of cells down a different path; the control arm reproduces the corpus's aggregate illegality (four of five models within ± 4 pp) but not its push/move composition. (ii) The treatment prompt is 734 characters longer than the control, and nothing in this design separates the rules' *content* from their *length* or position; a filler-matched third arm would, and is not run. (iii) This is not the un-primed arm. The worked example is present in both arms, so this experiment says nothing about whether the example primes `E2`, and nothing else in this study does.

What it establishes. It establishes neither that the rules are irrelevant, nor an effect size, nor, given the composition confound, a direction. What it does establish is the one thing needing no statistical power and no composition assumption: restoring the harness's own rules text verbatim solved nothing. 0 of 30 episodes in the control and 0 of 30 in the treatment, matching the corpus's 0 of 180. The failure documented in this paper is not repaired by telling the agent the rules, which is a weaker statement than "not explained by", and the one this arm can carry.

What survives. Stated conservatively: `E2` is not withdrawn. Roughly four fifths of it carries no trace of the example's phrasing; the two model annotators agree on it at 90.5% with $\kappa = 0.749$, the highest κ of any item in the conjunction, which bounds codebook specificity for the item, not its validity, and its 12 of 12 confirmed human verdicts are what bear on that; and its mechanism is corroborated by `E3`, the same dissociation with a *correct* plan, which the example does not model. But we no longer describe `E2` as evidence that these agents spontaneously dissociate plan from action. The claim is narrowed to this: the dissociation is present across the corpus, it is present at the step where a large share of the no-ops occur, and the agent was shown one instance of it.

THE POSITIVE CONTROL: PRE-REGISTERED, RUN, AND NULL

This was the principal design limitation of the study, and it has now been tested. No level is solved in any corpus condition and none in either rules-arm condition, so on coded corpus material the contrast item `I8` (a correct read followed by a correct legal move) is scored on 1.67% of steps and the instrument had never been shown to *pass* a turn a competent agent would pass. What that leaves unseparated is real: "the model does not use state it has itself transcribed" and "nothing in this scaffold enables any behaviour but noise" both predict this corpus, and the `I`-family's verbatim spans (accurate transcription beside contradicting prose, reproduced across executions) are evidence the first description is apt at the turns quoted, not a separation of the two hypotheses at the corpus level. The reflection-channel analysis later in this appendix narrows the second reading but does not close it.

We therefore ran the positive control. It is pre-registered: the protocol, its date and its commit fix the material, the exclusions, the admissible items, the unit, the denominators, the two tests with their directions, and how each outcome would be read, all of it written and committed before any label of the arm existed. The material is the ladder's three solvable rungs, the only place in this study where the system demonstrably works. Each staged unit is one turn, presented alone in a shuffled pool with the episode around it removed: no episode identity, no step index, no outcome, and no run-length cue. Each solved episode's winning step is dropped, because it is the one place `Box on Dock` appears; each unsolved episode's terminal five board-unchanged steps are dropped, because that is the stuck detector's signature and it is present in 100 of 100 unsolved episodes and none of the solved ones. Two rung-1 episodes are single-step wins, so dropping the winning step empties them; they are recorded as *not assessable*, since item-absent would be a claim about turns that do not exist, and the solved denominator is therefore 18 and not 20. After these exclusions the board-unchanged rate is 58.8% on both sides, so the contrast is not carried by the detector. 291 turns were labelled by annotator A, one fresh context each, under a prompt that admits only the seven items a single turn can support.

The registered primary. It is null. `I8`, predicted *more* common on solved episodes, since it is the item a turn with nothing wrong carries, was scored on 2 of 18 solved episodes against 4 of 40 unsolved, one-sided Fisher $p = 0.61$. This is not a near miss: the minimum detectable effect computed at registration needed 6 of 18, and at the turn level `I8` is 2 of 80 solved steps against 7 of 211 unsolved, indistinguishable, and numerically lower on the solved side. The registered secondary, `I1`, was predicted *less* common on solved episodes and was observed *more* common (15

of 18 against 29 of 40), the opposite direction, not significantly. The pre-specified sensitivity analysis on rungs 2–3, where the agents' rates exceed the chance baseline, reproduces both nulls. Both I8 counts fall below the pre-committed small-n threshold, so they are reported as counts against named denominators with no rate, which is itself one of the registered outcomes.

I2 in this arm. I2, the item the body's claim rests on, was not a registered test here, and its prevalence was recorded anyway because the earlier registration that motivated this ladder predicted I2 would collapse on solved episodes. It was scored on 1 of 18 solved episodes against 5 of 40 unsolved, numerically lower on the solved side, which is the predicted direction, on a solved-side count below the same pre-committed small-n threshold that governs I8. One labelled episode cannot distinguish a collapse from an absence, so this is reported as a count and not as support: the prediction is neither confirmed nor refuted, and we do not lean on it anywhere in the paper.

What we conclude. The reading was fixed before the data. The registration fixed how this outcome would be read, as "neither item moves: the items describe how these models play, not why they fail", and we report it in those words, with its scope stated. What was tested is the codebook's one whole-turn contrast item, I8, with I1 as the secondary, on single turns presented without their episode; what the null shows is that on this material neither item separates solved episodes from unsolved ones. The codebook is therefore *not shown* to separate competent turns from incompetent ones, and we present it as a description of observed behaviour rather than as a validated discriminator. The null does not show that the codebook cannot separate them: only seven of its items can be scored on a single turn, and the rest (items whose rules quantify over an episode or a run, items that need the turns around the one being scored, and the two unit-defective items) could not be scored by this design and are untested, not failed. Four things bound even the narrow statement. One board supplies each rung, so the independent unit is the puzzle and $n = 3$. Against the pre-registered chance baseline 12.76 of the 20 solves are expected by chance, so the "success" regime is weak and a null against it is correspondingly weak evidence. The registered tests rest on annotator A alone. And H1 was never applied on either side, so the arm says nothing about it in either direction. 246 of the 291 turns carried an observation that fitted no admissible item and was recorded instead of being forced into one; with seven admissible items this is partly by construction, but it is the clearest signal here of what the single-turn subset cannot express.

The annotator-artefact check. An annotator that missed I8 on clean turns more often on the solved side would produce this null with no failure of the codebook, so we tested for that asymmetry. One author judged drawn turns clean, not clean or unsure (board facts accurate, move legal, action matching; not move quality), blind to side and to annotator A's labels: all 16 solved-side turns A left unlabelled, 16 unsolved-side ones drawn by seed as a control, and all 9 I8 labels, shuffled. The check stopped at turn 30 of 41, fixed before unblinding, so the checked turns are a random subset of the draw, not all of it. A had missed a clean turn on 0 of 10 solved-side unlabelled turns against 2 of 12 unsolved, both close calls, and all 8 checked I8 labels were confirmed; none was unsure. The difference runs opposite to the direction that could manufacture the null. It is a direction, not a rate: the cells are small, 6 of the 16 solved-side turns went unchecked, and the checker wrote the codebook and operated annotator A, so the check rests on blinding, not independence. It concerns I8 only and is pooled neither with the model-annotator agreement below nor with the 316 verdicts of Appendix E.

Model-annotator agreement on this arm. A second model annotator (GPT-6 Astra, a newer model than the corpus's annotator B, run through the same coding agent) double-coded a pre-committed, seed-shuffled sample of 87 of the 291 turns, for agreement only; the registered tests are annotator A's, on all 291. I1, the registered secondary, reaches κ 0.839 and E2 0.973, both inside the prevalence band. I8, the registered primary, is too rare to read on κ (mean prevalence 2.87%), so we read its observed agreement, 98.85%, and PABAK, 0.977. As everywhere in this paper, agreement between two language models measures codebook specificity, not validity, and these figures are never compared with the corpus κ, which is a different arm with a different annotator B. Three things about the run are disclosed. It stopped one unit short of the sample on a clock-and-quota rule fixed before the stopping point was known, and the last unit was added by a declared amendment before any κ was computed; without it no figure moves by more than 0.01. The sample spans two builds of the annotator's client, with model, reasoning settings and system prompt measured identical before the second build labelled anything; the first unit was run by hand in the agent's desktop app and the rest by a script. And the second model's blindness to annotator A's labels is audited from its own session

logs, which record no read of them, rather than enforced by its sandbox: the audit shows it did not look, not that it could not have.

THE LABEL EXPOSURE IN THE TRACE WORKBOOKS

The trace workbooks used for the three blind reading passes carried lines exposing bare label strings to every reader. The exposure was constant across all three passes and consisted of names without definitions; a mechanical check verifies that no label code reached any observation file. It was not corrected mid-corpus, because correcting it partway would make the reading passes incomparable. The workbooks issued to the second annotator carry no such line.

CORRECTION TO AN EARLIER PILOT INTERPRETATION

An earlier version of this work carried a third input limitation: that when the worker stands on a dock the renderer emits no worker entry and labels the cell `Unknown`. That is real in the pre-patch pilot datasets (12 of 243 steps in one, 15 of 535 in another), datasets retained as provenance and not reported here. We patched it (the renderer and item-vocabulary normalisation Appendix D lists as applied) before the reported corpus was collected, and the limitation does not hold for this corpus. What sustained the mistaken belief was a defect in our own analysis tooling, not in the harness: our rendering summary computed the dock list by matching the exact string `Dock`, silently omitting any cell reading `Worker on Dock`, and that summary line was the header of every step the blind readers read. This history is reported because it moved `B5` from a limitation to *evidence*.

THE REPRESENTATION NORMALISATION, A VALIDITY STEP AND NOT A CONTRIBUTION

The harness's coordinate framing was normalised before collection: the shipped text path described the board in one convention while delivering it in another, and we normalised the prompt so the stated format matches the delivered one. This is a validity step in the sense of HarnessFix (Chen et al., 2026a) and representation sensitivity (Martorell, 2025), not a finding, and we do not headline the bug. In text mode the perception module makes no API call, so its coordinate prompt never reaches the model, and the confound surface is therefore smaller than earlier accounts of this project claimed.

THE REFLECTION CHANNEL: ELICITED, NOT INJECTED, AND LARGER THAN THE THOUGHTS

The reflection module asks how the state changed after the last action on every step, including the first, when the prior context is empty. It was deliberately left in place, because changing it mid-project would have made the corpus internally incomparable. It is the reason `B2` (fabricated progress in the step-1 thought) is described as elicited, never *injected*: reflection and reasoning are two calls to the same model, so nothing is inserted from outside. But the harness's role is the invitation and the carriage: the reflection's output is placed in the next call's context under a `Reflection on Past:` heading, so a fabricated narration is handed forward to the reasoning step as given. The content is still the model's.

The figures below are mechanical over the full corpus, 1,561 steps, 180 of 180 episodes aligned prompt-for-step. First, a disclosure about the record itself. `agent_observation.reflection` is null on 1,561 of 1,561 steps, as are `game_trajectory` and three other fields; only `textual_representation` is populated. The reflection text exists, recoverable from the run logs inside the reasoning call's own prompt, but it never reached the structured logs the annotation traces are built from. Both labellings therefore scored the reasoning output with no sight of the text that produced it. This is a property of the instrument, not an annotation error, and it is not repairable after the fact.

Second, the channel fabricates, at a step where fabrication is impossible. At step 1 no action has been taken, so any assertion of change is false by construction. 66 of 180 step-1 reflections (36.7%) assert a state change.

Third, the reasoning step inherits it, measured without reference to either channel's vocabulary: the longest run of consecutive words a thought shares with the reflection it was handed is ≥ 8 words on 50.9% of steps, ≥ 15 on 19.1% and ≥ 30 on 6.6%; at step 1 alone, 51.1% / 17.8% / 5.6%.

Roughly one turn in five is substantially restating its reflection, and the reuse is as strong at step 1 as anywhere.

Fourth, the two channels do not track each other, which is what lets the `B` family stand as a count of the reasoning channel, and as no more than that. If every fabrication were inherited, the two channels would track. They do not: the reflection fabricates far more than the thoughts do (36.7% against 3.3% at step 1), and on the 1,310 steps where a change claim is provably false there are 659 where the reflection asserts a change and the thought does not carry it forward. The reasoning step mostly declines the invitation. So the channel is real and it is upstream. What these figures do not establish is that the fabrications the thought does make are its own rather than inherited: on the same provably-false steps, most of the false change claims the thought makes were already present in the reflection it was handed; this is a direction and not a rate, because our change-claim patterns are calibrated to the reflection's vocabulary and undercount the thought side. Whether a given `B`-family label sits on such a step is the cross-tabulation left open below.

What follows for the reader. This paper's fabricated-progress counts describe the reasoning channel, which is the channel a turn-level reading can see. The agent as a whole fabricates more than we report, because a second channel does so and no annotation of ours ever looked at it. It is also a result about scaffold design: the reflection is solicited on *every* turn, and the board it is asked to narrate a change in was unchanged by the previous action on 76.7%–86.1% of comparable steps, per condition (the range spans the six conditions; we do not pool them). A narration of change requested unconditionally on a mostly-static board is a request with no true answer most of the time, and it gets answered anyway.

Left open, deliberately. Whether any individual `B`-family figure needs an inheritance caveat attached requires cross-tabulating the `B1`/`B2`/`B3`/`F5`/`B9` labels against the recovered reflection text. That cross-tabulation has not been run and is future work.

Statistical and annotation limits

- $n = 3$ **per cell.** This corpus supports claims about presence, breadth and relative magnitude. It does not support per-turn percentages, and we make none from the reading passes.
- **No pooled frequencies across decoding conditions.** Temperature 0 and temperature 1 are reported separately everywhere.
- **External validity.** Six fixed hand-authored levels and a single harness bound it, and small open models only (3–8B); at the smallest tier some output is closer to a floor effect than to a legible failure process.
- **Every annotator is a model.** κ measures codebook specificity, not validity. No human independently coded a complete episode.
- **Human-verdict coverage.** It is uneven across models, so no per-model claim about label defensibility is available.
- **No adjudication.** The two annotations are not adjudicated, and the results section does not depend on their being adjudicated; it reports reading-pass figures throughout. What adjudication could and could not change is bounded without doing it: any adjudicated count must lie between the labels *both* annotators applied and the labels *either* applied, and on 30 of the 31 items this paper cites, the "both temperatures" breadth claim already holds at that floor, so no adjudication outcome could falsify it. Two model-breadth claims would not survive a rebasing unadjudicated (`D13`'s *all five models*, `H1`'s *four*), and we do not rebase.
- **Two unit defects.** Two items carry a unit defect that makes their counts unsound, disclosed where they are used: `I5` and `B5`. Both describe an episode-level property with a step-level coding condition, and the two can disagree about the same turn. No `I5` or `B5` label was judged wrong. The codebook is deliberately not repaired: it was fixed before either annotation began, and editing it now would leave both annotations referring to a document that no longer exists.
- **Identifier misparse.** Four candidates in 180 episodes (two at temperature 0) are unadjudicated; nothing in this paper claims anything about that misparse.