WHAT SURVIVES IN A MULTI-TURN TRACE? DISSECTING STATE MAINTENANCE IN VLMS

Anonymous authors
Paper under double-blind review

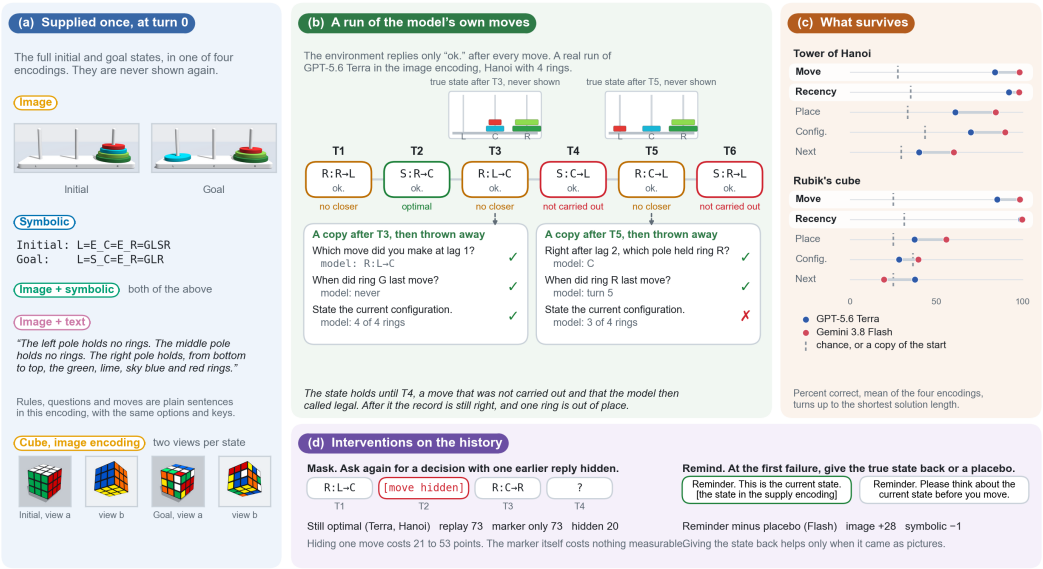


Figure 1: **Overview of SSD.** (a) The initial and goal states are supplied once, in one of four encodings. (b) A real run. The model makes its own moves and the environment only replies "ok.". Questions about the state run on copies of the conversation that are scored against the simulator and then thrown away. (c) Percent correct per probe for the two full grid models. The probes in bold can be answered from the model's own list of moves and stay high. The probes that need the state score lower, most clearly on the cube. (d) Mask and Remind change the history and measure the effect.

ABSTRACT

Agents built on vision-language models (VLMs) as vision-language-action (VLA) models often act for many turns. To perform well, they need to keep track of how their own actions change the scene even when it is no longer in view. When such an agent fails, current evaluations rarely tell whether it misread the scene, lost the state or chose a bad action. New observations let the model read the scene again, a question about the state can act as a reminder, and a model's account of its own behavior is not reliable evidence. We present Spatial State Dissection (SSD), a testbed that removes these three problems. The initial and goal states of a Tower of Hanoi or Rubik's cube problem are supplied exactly once at the beginning in one of four encodings that use images, symbols or text. Every probe runs on a discarded copy of the conversation and is scored against a simulator, and two interventions on the history test for cause. Across 19 model settings and more than 470K graded answers, SSD separates four steps that fail in different ways. Symbols are read almost without error while images of the cube are often misread. Models with little reasoning retain a record of their own moves but reconstruct the cube state poorly. With thinking enabled, the largest open-weight models retain more of the state. On Hanoi a model can work out the current state from its own earlier moves. When it tries a move that breaks the rules, such as putting a larger

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.

Table 1: **Comparison with the closest evaluations.** ✓ means yes and ✗ means no. ∼ means partly and n/a means that the question does not apply. *Run left unchanged* means that a measurement never enters the model's context. *Causal intervention* means that an input inside the run is changed to test whether it caused a failure. CubeBench adds solver tools, and a Cube Bench run has at most five steps. MET-Bench varies how the actions are encoded but varies the initial state only in a task with one step. Pereira & Zuidema (2026) is concurrent work on activations inside one reply.

	Image input	State given once	Changed by own moves	Many turns	Measured during the run	Run left unchanged	Encoding varied	Causal intervention
MET-Bench (Cohen & Mooney, 2026)	✓	✓	✗	✗	✗	n/a	∼	✗
OST-Bench (Lin et al., 2025)	✓	✗	✗	✓	✓	✗	✗	✗
CubeBench (Gao et al., 2026)	✓	✗	✓	✓	✗	n/a	✓	∼
Cube Bench (Anand & Shareghi, 2025)	✓	✗	✓	∼	✗	n/a	✓	✗
Hanoi world models (Pereira & Zuidema, 2026)	✗	✓	✓	✗	✓	✓	✗	✓
SSD (ours)	✓	✓	✓	✓	✓	✓	✓	✓

ring on a smaller one, the move is not carried out and the environment still replies only "ok.". The models often assume the move was carried out anyway, so they report that ring in the wrong place. In most of these failures, the model could report the exact state just before making the move but still chose an illegal action. States reached forward survive better than states worked out backward, and giving the state back helps only when the supply included images. SSD will be released upon acceptance.

1 Introduction

Long tasks often ask a model to keep track of a state that its own actions change. A robot moving blocks needs to keep track of where each block is after every move. An agent that edits files builds on what it has already changed. The environment does not always show the state again after each action. The model can still see what it has done, but it is left to work out the current state. We call this *state maintenance* and study how well vision-language models preserve the state over multiple turns. There is evidence that this is hard and that the difficulty is selective. Jung et al. (2026) find that models correctly judge over 79 percent of manipulation actions initially, but accuracy drops below 16 percent after eight turns even though perception of the current view remains high. Lin et al. (2025) find that accuracy falls the longer a model explores a scene.

These results show that performance degrades over turns, but they do not reveal where the failure occurs. A failure can arise from misreading the scene, updating the state incorrectly, failing to detect an unsuccessful action, or choosing the wrong action. Current evaluations rarely separate these steps for three reasons. First, new observations arrive at every step in most benchmarks. A correct answer may therefore come from the latest observation rather than from maintaining the state over time. Second, a question about the state can be a reminder. The question and the model's own answer stay in the conversation and can shape later turns. Third, a model's own explanation of its behavior is often unreliable (Turpin et al., 2023; Chen et al., 2024). Related work addresses some of these issues, but not all of them (Table 1). Studies of state tracking give the state once and ask only for the final state (Kim & Schuster, 2023; Cohen & Mooney, 2026). In CubeBench the agent can look at the cube again during the run (Gao et al., 2026), and Pereira & Zuidema (2026) probe activations inside one reply.

We present SSD, a testbed benchmark that removes all three of these problems. The initial and goal states of a Tower of Hanoi or Rubik's cube problem are supplied once in one of four encodings. The model then makes one move per turn, and the environment only ever replies "ok.". Every question about the current state is asked on a copy of the conversation, scored against the simulator, and then discarded, so the measurement never alters the original run. Each problem also runs backward. The model then starts at the goal and names at each turn the move that led to the current state, undoing moves until it reaches the initial state. Since nothing in the run reveals the state, SSD can ask what the model still knows at any turn and trace failures to the four steps.

We first measure what survives and then ask two questions:

- **Direction.** Is a state produced by acting kept better than one worked out backward?
- **Cause and recovery.** When a run fails, is the lost state the cause? We ask whether giving the state back helps more than a placebo and whether the decision depends on earlier moves.

A control varies the **encoding** of the supply to ask whether the loss is only a failure to read it.

We run two proprietary models on the full grid of 1,056 runs each, three more settings on a pilot grid and 14 open weight settings. The four steps fail in different ways, and the two tasks stress two of them. For **reading**, the control shows that symbolic inputs are interpreted almost perfectly, whereas cube images are often misread. For **updating**, models that reason little recall at least 78 percent of their recent moves, while on the cube they write out only 22 to 53 percent of the pieces correctly against 36 percent for a copy of the initial state. With thinking enabled, Gemini 3.8 Flash in a small pilot and the two largest open-weight models retain more of the state. For **noticing**, the models often read the Hanoi state off their own list of moves as if attempted moves had been carried out, so they lose it mostly at moves that were not. For **choosing**, the model could often report the exact state immediately before a failure but still selected an illegal move. A symbolic reminder of the correct state was also incorporated without improving the next action. States reached forward also survive better than states worked out backward.

Our contributions are as follows:

- **A testbed that locates the failure.** The state is shown once, the environment gives no feedback, and every probe runs on a discarded copy with a key from a simulator. A new model run on the same messages yields a measure for each step (Table 2).
- **Forks as a tool for analysis.** A fork copies the conversation at any turn, asks one question and is thrown away, so the state can be measured at every turn without changing the run. Forks also ask for the state just before and after a failure (Figure 3 in Appendix A.4).
- **Two causal tests on the history.** Built on forks, Remind gives the true state back at the first failure and compares it with a placebo. Mask hides one earlier move and compares it with a control that marks the gap but gives the move back.
- **Four steps that fail differently.** Images of the cube are often misread. With little reasoning the cube state is barely updated, and thinking helps the largest open weight models keep the state. On Hanoi the models often treat attempted moves as carried out and make illegal moves even when they can write the correct state on request.

2 Related Work

State tracking. One line of work gives a model a start state and a list of operations and asks about the resulting state, for boxes and objects (Kim & Schuster, 2023), for chess and Othello (Toshniwal et al., 2022; Li et al., 2023) or for permutations of objects (Li et al., 2025). In all of these the model watches operations that someone else chose within a single input. Generative models can pass world-model evaluations while maintaining incoherent implied states (Vafa et al., 2024), and language models often track state changes unreliably in text-based games (Wang et al., 2024). In concurrent work, Pereira & Zuidema (2026) report that reasoning models encode the Hanoi state well at the end of the prompt and lose it during the chain of thought. Their evidence comes from activation probes applied to open-weight text models within a single response. Ours is behavioral and covers many turns, proprietary and open weight models and several encodings.

Evidence for the three problems. New observations arrive during the run in many evaluations, as in VAGEN, OST-Bench and CubeBench (Wang et al., 2025a; Lin et al., 2025; Gao et al., 2026). Earlier turns can influence later decisions, as models may commit to their own early guesses when following an instruction across multiple turns (Laban et al., 2026). Explanations can omit a biasing feature that changed the answer (Turpin et al., 2023) and predict answers on changed inputs poorly (Chen et al., 2024).

Encoding, puzzles and causal tests. With the input in view, models do better with text than with images (Fu et al., 2024; Tang et al., 2025). MET-Bench, CubeBench, and Cube Bench vary the input encoding (Cohen & Mooney, 2026; Gao et al., 2026; Anand & Shareghi, 2025), but the model either does not act or can observe the state again during the run. Puzzle benchmarks score full move lists written in one reply (Valmeekam et al., 2023; Shojaee et al., 2025), a practice that Lawsen (2025)

questions, and we ask for one move per turn. An external instruction can restore attention to an image when self-prompted inspection fails (Shi et al., 2026). Our interventions carry this logic to a state no longer in view (Appendix E).

3 METHODOLOGY

At the start of a run the model receives the initial and goal states, and after that the state changes only through its own moves. The conversation then contains the initial state and the model's own moves which we call its *record*, but never provides the current state. To know where the pieces are now, the model has to work out what its moves did to the initial state. We say that the model *maintains* the state when it can report the current state correctly when asked and act on it. We evaluate this behavior rather than how it is achieved, so a model that maintains the state and one that reconstructs it at each probe are treated equivalently.[1] SSD is built on four design principles. **Supply once.** The full initial and goal states are given at turn zero and never again. After every move, the environment replies only "ok.", preventing the model from inferring the current state from a new observation. **Questions on discarded copies.** We ask every question on a *fork*, a copy of the conversation up to a given turn plus one question (Figure 3). The answer is scored and the fork is discarded, allowing the main run to continue as if no question had been asked. **Ground truth before asking.** Every scored question has a key that a simulator computes before it is sent, and a model's own account of what it did is never used as evidence. **One builder for every condition.** A single message-building library is used across all conditions, ensuring that they differ only in the factor being studied. Small changes in prompt format can move scores by large amounts (Sclar et al., 2024), so a shared format makes conditions comparable but not independent of that format.

3.1 TASKS

A task needs an exactly checkable state that fits in one turn as an image and as text, and a known shortest solution at every level of difficulty. Scene exploration benchmarks are not suitable here because the state is revealed continuously through a stream of observations (Lin et al., 2025; Yang et al., 2025). We use two puzzles that differ in structure (Table 5 in Appendix A). **Tower of Hanoi.** The puzzle has three poles and one to six rings, and only the top ring of a pole may move. Our initial and goal states are arbitrary legal configurations, as in Pereira & Zuidema (2026), rather than the classic full-transfer setting with its well-known recursive solution (Hinz et al., 2018). Under *Rule* a larger ring may never rest on a smaller one. *Free* is our own control condition with no size rule. **Rubik's cube.** We use a $3 \times 3 \times 3$ Rubik's cube with only the twelve quarter-turn moves and do not allow rotations of the entire cube. A problem starts from a solved or a scrambled cube, and its goal lies an odd number of quarter turns away, from 1 to 9. We count quarter turns only (Agostinelli et al., 2019). In that metric every move switches between two classes of positions (Rokicki & Davidson, 2014), so it changes the distance to the goal by exactly one.

The two puzzles stress different abilities. Every cube turn is legal and moves eight pieces simultaneously, so the model must reconstruct the resulting state from its sequence of moves. A Hanoi move names one ring and its new pole, so the state follows from the model's own moves as long as it tracks which of them were carried out. Most Hanoi moves expressible in the notation are illegal, and a legal move does not necessarily reduce the distance to the goal. A seeded generator builds six problems for each of 22 cells. The cells are 6 ring counts times 2 rule conditions for Hanoi and 5 depths times 2 start conditions for the cube. Exhaustive search gives the shortest solution length L^* of each of the 132 problems, from 1 to 51. A run may use at most $B = \max(2L^*,\ L^* + 2)$ turns (Appendix A).

3.2 FOUR INPUT ENCODINGS

The encoding of the supply is our control since it separates reading the supply from keeping the state. In the **image** encoding, the model receives rendered images of the initial and goal states. Hanoi uses one picture per state, and the cube uses two pictures that together show all six faces. In the **symbolic** encoding, the model receives the initial and goal states as compact strings defined by a fixed grammar. In the **image and symbolic**, encoding it receives both. In the **image and text** encoding, it receives the pictures and a description in plain sentences. The first three encodings use exactly the same system prompt, byte for byte. The fourth encoding uses plain sentences for every

[1] We thus avoid calling a failure to maintain the state "forgetting".

Table 2: **What SSD measures for each step.** Every probe runs on a discarded copy and is scored against a simulator. Running a new model on the same messages yields each of these measures.

Step	What SSD asks or changes	What it reports
Reading	the configuration at turn zero in each encoding	objects placed from the supply
Updating	place recall and the configuration after the model's own moves	the gap to move recall and the displaced objects placed
Noticing	the legality of the last move and the state after a rejected move	"legal" answers to legal and illegal moves and the match to the state with the move carried out
Choosing	the next move among four options and the state just before a failing move	optimal moves and the failures that follow an exact state
Cause	Remind with a placebo and Mask with marker and replay controls	the change in the next decision
Controls	four encodings, two directions and a comprehension gate	differences paired by problem

rule, question, option and reply. Its option sets, keys and scoring are identical to those of the other encodings (Appendix B.6).

3.3 One run

A run is one problem in one encoding and one direction. The system prompt gives the notation and the reply rules. The first user message provides the supply and task, followed by the rules, turn budget, and one example turn (Appendix C). The task asks the model to reach the goal in as few moves as it can. Each later turn request reads "Turn k of at most B. Your move?", the model replies with one move and the environment replies "ok.". The run ends when the goal is reached or the turn budget is exhausted. Any illegal move, such as placing a larger ring on a smaller one, is rejected and leaves the state unchanged. The state stays as it was and the turn still counts. The model is told this rule at turn zero, but the environment still replies "ok." because any other reply would reveal the state. The model must therefore infer whether a move was actually carried out. Replies that cannot be parsed as a single move are treated the same way. These are rare for proprietary models but more common for some open-weight models (Appendix A.5). Replies hold only a move, so the model tracks the state without visible working.

Directions. In a *forward* run the model moves from the initial state toward the goal. In a *backward* run the model stands on an anchor state that starts at the goal. At each turn, the model identifies the move that produced the anchor state, and the anchor then steps backward by one state using that move. The run ends when the anchor equals the initial state. Backward states are thus worked out, and forward states are produced by acting. With full states and moves that can be undone, the two directions are symmetric for a search procedure (Bonet & Geffner, 2001), so a gap between them is informative. **Comprehension gate.** A low score in the symbolic encoding may indicate that the model never understood the notation (Ivanova, 2025). Before the first turn, we therefore ask five multiple-choice questions about the supplied state on a fork. Backward runs include a sixth question about the goal. No move named in a check is an optimal first move, so the checks give no hint about the solution. A run is *admitted* if it answers every check correctly on the first try. A run that fails goes on with a flag and is left out of the main analysis (Section 4.1 gives the rates). Admission can differ between encodings, and dropping runs on a criterion that the encoding can affect would bias the comparison (Montgomery et al., 2018). Comparisons therefore use only problems admitted in every encoding.

3.4 Probes

Table 2 maps each step to what SSD measures. Four question families probe what the model can report about the current state, and a fifth is used only at the first failure (Table 6 in Appendix A.5). A correct answer does not imply that the state was stored, since it may be reconstructed from the supply and prior moves. **Recall** asks about moves at lag 1, 2, or 3. *Move recall* asks which move was made or which piece moved and is keyed to the model's written move, even if it was not carried out, so the visible record is sufficient. *Place recall* asks where a piece was after that move and therefore requires the state. **Recency** asks when a piece last moved, with rejected Hanoi moves excluded; on the cube, recall and recency refer to a face and can be answered from the record. **Configuration** asks for the full current state, scored by the fraction of rings on the correct pole or movable cube pieces in the correct slot, regardless of orientation. **Reasoning** asks whether the previous move reduced the distance to the goal, or whether it was legal under Rule and in backward runs, then asks the model

to choose among four next moves. On Hanoi, these are one optimal, one distance-preserving, one distance-increasing, and one illegal move. All other questions are keyed to the true state, where rejected moves never occurred.

Schedule. Reasoning probes run after every completed turn. The other families run at checkpoints fixed before any model runs. These are turns $\min(3, L^*)$, $\lceil L^*/2 \rceil$ and $\lceil 3L^*/4 \rceil$, together with turns L^* and $B - 1$. A configuration probe at turn zero measures how well the supply was read. Option order is a seeded shuffle (Zheng et al., 2024; Pezeshkpour & Hruschka, 2024), and one parser reads the written reply in every encoding (Appendix A.5).

3.5 Failures and interventions

Work on faithfulness tests a claimed cause by intervening and measuring the resulting change in behavior (Tutek et al., 2025; Chuang et al., 2026), and we intervene on the conversation history. **Failure.** The simulator marks the first failure of a run. Under Rule, this is the first illegal move. In the other conditions, it is either an illegal move or the first move that increases the distance to the goal. Unparseable replies are never treated as the first failure because they reflect reply format rather than state. **Belief and reality.** At the first failure, two forks ask for the full state immediately before and after the failing turn. We score each answer against both the true state and the state implied if all of the model's attempted moves had been carried out. The latter is used only for scoring and reveals whether the model reports the world as it is or as it intended it to be.

Remind. At the first failure, we copy the history twice and continue each copy as its own run. Before the next turn, one copy receives "Reminder. This is the current state." together with the true state in the original supply encoding. The other receives the placebo "Reminder. Please think about the current state before you move." (Figure 11), which contains no state and matches the reminder in position and opening words (Boot et al., 2013). If only the true reminder helps, the missing state was likely causal. If neither helps, the failure instead lies in move selection or rule use. In image conditions, the true reminder also introduces a new render, so part of any gain may come from the image itself. A configuration probe after the next move checks whether the reminder was incorporated; in the symbolic condition, it also tests applying one move to a given state.

Mask. We revisit a finished run and ask again for an optimal decision at turn k after replacing the model's move at an earlier turn j with "[move hidden]". Turn j *constrains* the decision if some move consistent with the gap would make the original decision no longer optimal. A mask is valid only when at least two moves fit the gap. A placebo mask hides a matched turn that does not constrain the decision; on the cube, nearly every earlier move constrains the decision, so no placebo is available. A *marker* control keeps the marker but reveals the hidden move in the next message, isolating the cost of the marker itself. A *replay* sends the unchanged history to measure how often the model changes its decision by chance (Atıl et al., 2025).

3.6 Models and analysis

We run proprietary models from three providers and open-weight models through a common runner with the same system prompt (Section 4.1). Our main open-weight models are Gemma 4 31B (Gemma Team, 2026), Qwen3.8 27B (Qwen Team, 2026b), and Qwen3.5 9B (Qwen Team, 2026a). We also evaluate InternVL3.5 14B (Wang et al., 2025b), GLM-4.6V Flash (GLM-V Team, 2025), and Kimi-VL A3B (Kimi Team, 2025), with and without thinking when supported. We include SpaceThinker (Remyx AI, 2025b) and SpaceQwen (Remyx AI, 2025a), the two strongest public spatial models in OmniSpatial (Jia et al., 2026), together with their base model Qwen2.5-VL 3B (Bai et al., 2025) (Appendix A.9). The full grid contains 1,056 runs per model from 22 cells $\times$ 6 problems $\times$ 4 encodings $\times$ 2 directions. We analyze turns up to L^*, which every run contains because the goal requires at least L^* moves. To avoid overweighting longer problems, we average within each problem before averaging across problems. We report 95 percent bootstrap intervals over problems, which preserve dependence among probes from the same problem (Miller, 2024). Models are reported individually, and we do not correct for multiple comparisons, so intervals that barely exclude zero provide weak evidence. We also report the *copy line*, which repeats the initial state as a no-state baseline (Cohen & Mooney, 2026; Hewitt & Liang, 2019).

Table 3: **What survives, per probe and encoding.** Each cell reports the percentage over turns up to L^* for problems admitted across all encodings, using one or both directions. The counts under the model name are problems, Hanoi first. *Read* is the configuration probe at turn zero. *Move* and *Place* are move recall and place recall. *Rec.* is recency. *Conf.* is the configuration after turn zero. *Next* is the share of optimal next moves on items that need the state. *Avg* averages the five post-turn-zero probes within each problem and then across problems containing all five probes (Table 11). The first two rows give the score of a chance answer and of a copy of the initial state on the same items. Dark and light shading mark the best and second best encoding of each model. Table 11 adds intervals, and Table 8 gives the pilot settings.

		Tower of Hanoi							Rubik's cube						
Model	Supply	Read	Move	Place	Rec.	Conf.	Next	Avg	Read	Move	Place	Rec.	Conf.	Next	Avg
Chance		–	28	33	35	–	30	–	–	25	25	31	–	25	–
Copy of the initial state		–	–	–	–	43	–	–	–	–	–	–	36	–	–
Proprietary models, full grid															
GPT-5.6 Terra, reasoning off	Symbolic	98	78	53	90	68	33	61	100	85	43	99	29	44	58
46/43 problems	Image	100	86	58	90	64	35	62	62	84	41	99	22	46	57
	Image + symbolic	100	84	68	92	71	44	68	100	82	41	99	31	34	55
	Image + text	100	87	65	97	77	48	69	100	90	24	99	31	26	51
Gemini 3.8 Flash, thinking low	Symbolic	100	99	89	98	93	60	84	100	100	54	100	43	24	61
72/53 problems	Image	100	98	84	98	88	58	81	68	95	46	100	26	19	54
	Image + symbolic	100	98	77	98	87	62	80	100	99	50	100	37	11	56
	Image + text	100	98	88	98	91	61	84	100	100	73	100	53	24	67

4 Results

4.1 Setup and how to read the results

Two proprietary models ran the full grid of 1,056 runs each, GPT-5.6 Terra (OpenAI, 2026) with reasoning off and Gemini 3.8 Flash (Google DeepMind, 2026) with thinking low. Three pilot settings ran 22 problems in the symbolic and image encodings, Flash with thinking high and Claude Sonnet 5 (Anthropic, 2026) with effort low or thinking off (Appendix B). The gate excluded 21 percent of Terra runs and 3 percent of Flash runs. Across all encodings, Terra admitted 46 of 72 Hanoi problems and 43 of 60 cube problems, while Flash admitted 72 and 53, respectively. All numbers are percentages, differences are in points paired by problem, and brackets give 95 percent bootstrap intervals over problems. The main pattern also holds when every run is included (Appendix B.1).

4.2 What survives

The model's record of its own moves survives in every encoding. The state that those moves produce survives less well. On Hanoi it is lost at moves that were not carried out, which the models often count as done, and on the cube within a few moves. (i) Questions answerable from the visible record remain high (Table 3), with both models recalling at least 78 percent of their moves across all encodings. (ii) Questions that need the state score lower. Place recall is below move recall in all 16 cells of model, task and encoding, although both ask about the same earlier turn. On the cube, the models place only 22 to 53 percent of pieces correctly, compared with 36 percent from simply copying the initial state. (iii) On the cube the state is lost within the first few moves, and it soon scores no better than a copy of the initial state (Figure 4 in Appendix B). For cube problems with $L^* \geq 7$, configuration accuracy falls below the copy line by turn 3 in seven of the eight model–encoding combinations. (iv) The two tasks lose the state in different ways (Table 12). On Hanoi the configuration probe places 91 to 95 percent of the rings correctly for Terra for up to 8 moves, as long as no move was rejected, and 98 to 100 percent for Flash for up to 16. After a rejected move this falls to 49 and 68 percent. Just after a first failure that is a rejected move we compare the answer with the true state and with the state that would hold had the move been carried out. Where the two differ, the answer matches the second on every ring in 65 and 41 percent of cases and the true state in 3 and 31 percent (Appendix B.4). On the cube, Terra and Flash place 18 and 92 percent of the pieces away from their start correctly on problems one move deep and 5 and 22 percent after two or three moves, so Terra barely computes the cube state. (v) Both full grid settings reason little, and more thinking narrows the gap, as it does for the largest open weight models (Section 4.6). With thinking high Flash keeps the full Hanoi state in the symbolic encoding on 12 problems (Table 8).

4.3 Direction. Forward and backward

The state survives better when the model produced it by acting than when it had to work it out backward from the goal. The record depends much less on the direction than the state

Table 4: **Interventions.** (a) Remind. Each cell is the true reminder minus the placebo, in points, on whether the next move after the first failure is optimal. A positive cell means that the true state helped more than the placebo line, and a cell near zero means that it did not. In the encodings with image, the reminder adds pictures and the placebo does not. (b) Mask. Each column under *Still optimal* gives the percent of decisions that stay optimal when the same decision is asked again. *Replay* sends the unchanged history, so it shows how often a model repeats itself. *Marker* hides one earlier move and gives it back in the next message, so a value close to Replay means that the marker alone costs little. *Mask* hides the move, and *Mask − marker* measures the cost of losing it. *Placebo* hides a nonconstraining move. A near-zero *Mask − placebo* indicates that the cost is insensitive to whether the hidden move matters. *Pairs* counts repeated decisions, and the cube has no placebo. Brackets show 95 percent bootstrap intervals over problems.

(a) Remind, reminder minus placebo

Model	Symbolic	Image	Image + symbolic	Image + text
GPT-5.6 Terra	+1 [−4, +6]	+3 [−1, +7]	+6 [+1, +11]	+9 [+4, +14]
Flash, thinking low	−1 [−10, +8]	+28 [+19, +38]	+19 [+10, +28]	+14 [+4, +25]

(b) Mask

			Still optimal				Difference	
Model	Task	Pairs	Replay	Marker	Mask	Placebo	Mask − marker	Mask − placebo
GPT-5.6 Terra	Hanoi	62	73	73	20	16	−53 [−68, −35]	+3 [−13, +20]
	Cube	81	74	79	48	–	−32 [−52, −7]	–
Flash, thinking low	Hanoi	399	86	84	60	70	−24 [−31, −17]	−10 [−16, −4]
	Cube	182	39	43	22	–	−21 [−30, −13]	–

does. We compare the two directions on the same problem and encoding (Table 9 in Appendix B). (i) Forward runs keep 8 to 31 more points of the configuration and 10 to 34 more points of place recall, for both models on both tasks, while move recall differs by at most 10 points. (ii) The gap remains before any rejected move, at 14 points $[7, 21]$ for Terra and 4 points $[2, 6]$ for Flash on Hanoi (Appendix B.4). On the cube both directions fail at the same turn, and forward runs keep 6 to 13 more points from solved and from scrambled starts alike. (iii) Search is as hard backward as forward (Section 3.3), so the gap lies in the models, though a backward run also asks the model to undo each move.

4.4 Cause and recovery. Is the lost state the cause?

On Hanoi the models often choose an illegal move although they can write the correct state when asked. When the supply included images, giving the state back improves the next move more than a placebo line does. With symbols the state is taken in and the next move still does not improve. (i) On Hanoi, a failing move is often a move that could not be made. In 85 to 90 percent of these failures the model wrote out the full state exactly on a fork just before the move, and in that state the move breaks the rules. The model thus had what it needed to see that the move was illegal and made it anyway, although whether it used that state when it moved is not shown. Asked on a fork after every move whether it was legal, Terra calls 86 percent of legal and 71 percent of illegal moves legal under Rule, and Flash 92 and 53 percent (Appendix B.4). By an explicit rule, most first failures on both tasks follow a state that the model could write exactly (Appendix B.5). (ii) In the encodings with pictures the reminder helps Terra by at most 9 points and Flash by 14 to 28 points (Table 4a), most consistently on the cube (Appendix B.3). The placebo shows no picture, so part of the gain may come from a new picture as such. (iii) In the symbolic encoding neither model gains over the placebo or over runs with no reminder, although one move later the state is 18 to 37 points more correct than after the placebo (Table 14). The state was taken in, so the failure in this encoding points to the choice of a move, although next move scores are low there for any state. (iv) Hiding one earlier move costs 21 to 53 points of optimal decisions against the marker control, while the marker itself costs nothing measurable (Table 4b). Little of this cost is specific to the move that matters, with no clear cost for Terra and 10 points $[4, 16]$ for Flash on Hanoi.

4.5 Control. Is the loss only a failure to read the supply?

No. The encoding matters when the supply is read, and after that the state falls in every encoding. At turn zero only the cube image encoding misreads the supply, placing 38 points $[29, 47]$ fewer pieces than the symbolic encoding for Terra and 32 points $[26, 38]$ fewer for Flash. With

symbols or words beside the same pictures both models read the supply at 100 percent, and the configuration still falls within the first turns (Appendix B.6).

4.6 Open weight models

With thinking off the two largest open weight models show the gap of the proprietary models, and with thinking on they keep the Hanoi state. Smaller models misread the pictures and often fail the gate, and the two spatial models score no higher than their base. We report the 14 settings over all runs (Table 7). The gate passes 80 to 91 percent of the Hanoi runs of Qwen3.8 27B and Gemma 4 31B and at most 77 percent for the others. At turn zero these two models place 93 to 97 percent of the rings from the Hanoi pictures, and no setting places more than half of the cube pieces from pictures. In the symbolic encoding 11 of the 14 settings score higher on move recall than on both place recall and the configuration. On the 66 problems of the thinking settings, thinking raises the share of displaced rings placed correctly to 95 to 98 percent for Qwen3.8 27B and Gemma 4 31B, 7 to 37 points more than without it. On the cube Gemma gains 21 to 52 points. On admitted runs whose every move could be read, both still keep 95 to 100 percent of the Hanoi state. Qwen3.5 9B gains less, GLM and Kimi gain little, and the thinking settings also sample while the others decode greedily (Appendix B.8). SpaceQwen and SpaceThinker score 2 to 25 points below Qwen2.5-VL 3B on average.

5 Discussion

Four steps, four fixes. The failures that SSD separates suggest different remedies. Reading fails for cube images, a problem of perception and not memory. Updating fails in models that reason little. The list of moves they wrote (their record) survives while the cube state it implies is soon lost. Theory suggests that fixed-depth transformers cannot track such compositions without a chain of thought (Merrill et al., 2024; Liu et al., 2023), and thinking helps the largest open weight models keep the state. Noticing fails on Hanoi, where the models often read the state off their record as if attempted moves had been carried out (Figure 1 and Appendix D). Feedback on failed actions should remove much of this loss, so a benchmark with such feedback would miss it. Choosing fails too since the models make illegal moves although they can write the correct state when asked. A single score over turns would mix these four causes. **Direction.** The record depends much less on the direction than the state does, so the gap lies in building the state from the record. Applying a move may be easier than undoing one.

Limitations. Our claims concern puzzles under a controlled supply, and arbitrary states and changed move names do not rule out knowledge of solving methods. A keyed answer is behavior and not a view of an internal state. Scores depend on one prompt format and on clean renders (Appendix A.10). Both full grid models reason little and track the state without visible working. The gate makes the admitted runs a selected set. The Remind placebo shows no picture, and Mask pairs concern the recent record. The model is told its turn budget, which reveals L^* and may make the task easier.

6 Conclusion

We asked what a vision-language model can still report about a state that it saw once after its own moves have changed it. The testbed answers this at any turn without touching the run and separates reading, updating, noticing and choosing. Forks make this possible. Every question runs on a copy of the conversation that is thrown away, so measuring never feeds the state back to the model. The same copies let us change the history to test a cause. In models that reason little, the record of their own moves survives better than the state it implies. With thinking on, the largest open weight models keep more of the state. The two tasks expose two separate failures. On Hanoi, the models often call an illegal move legal and write the state as if the move had been carried out. On the cube, they compose their moves into a state poorly. Knowing the state is not enough either. In most Hanoi failures that are illegal moves, the model could write the exact state just before and still chose the move, and a reminder in symbols was taken in without improving the next move. It therefore help agents without new observations to reason at each step, to be told when an action fails or to keep the state itself. Our puzzles are small and exactly checkable which is what lets each step be measured. The same protocol can be applied to any task whose state a simulator can check, and richer scenes, photographs and agents with feedback are natural next steps.

Reproducibility statement

Section 3 and Appendix A describe the protocol, the seeded problem generator, the independent validator and the frozen list of problem files. Appendix C shows every message that a model receives in each encoding. Section 3.6 gives the analysis, and Appendix A.9 gives the decoding of the local models. Every table comes from scripts that read the logged runs with a fixed bootstrap seed. The logs record the model version that each reply reports. The problems, the message library, the runner, the analysis scripts and the logs will be released upon acceptance.

AI use statement

We used the AI assistant, Claude Fable 5, to build the 3D simulators that render the synthetic pictures shown to the models, and no image generation model was used. It also assisted with some of the initial code for problem generation and was used to aid and polish the writing of the manuscript. We verified all of the code and made updates. Every answer is scored against a simulator key, so no language model graded any answer.

References

Forest Agostinelli, Stephen McAleer, Alexander Shmakov, and Pierre Baldi. Solving the Rubik's cube with deep reinforcement learning and search. *Nature Machine Intelligence*, 1(8):356–363, 2019. doi: 10.1038/s42256-019-0070-z.

Dhruv Anand and Ehsan Shareghi. Cube Bench: A benchmark for spatial visual reasoning in MLLMs, 2025. URL https://arxiv.org/abs/2512.20595.

Anthropic. System card: Claude Sonnet 5. https://www.anthropic.com/claude-sonnet-5-system-card, June 2026. Accessed 2026-09-26.

Berk Atıl, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, Zhe Wu, Lixinyu Xu, and Breck Baldwin. Non-determinism of "deterministic" LLM system settings in hosted environments. In *Proceedings of the 5th Workshop on Evaluation and Comparison of NLP Systems*, pp. 135–148, Mumbai, India, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.eval4nlp-1.12. URL https://aclanthology.org/2025.eval4nlp-1.12/.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, et al. Qwen2.5-VL technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Blai Bonet and Héctor Geffner. Planning as heuristic search. *Artificial Intelligence*, 129(1–2):5–33, 2001. doi: 10.1016/S0004-3702(01)00108-4.

Walter R. Boot, Daniel J. Simons, Cary Stothart, and Cassie Stutts. The pervasive problem with placebos in psychology: Why active control groups are not sufficient to rule out placebo effects. *Perspectives on Psychological Science*, 8(4):445–454, 2013. doi: 10.1177/1745691613491271.

Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. SFT or RL? An early investigation into training R1-like reasoning large vision-language models. *Transactions on Machine Learning Research*, 2025a. URL https://openreview.net/forum?id=wZI5qkQeDF.

Jingxuan Chen, Derek Yuen, Bin Xie, Yuhao Yang, Gongwei Chen, Zhihao Wu, Yixing Li, Xurui Zhou, Weiwen Liu, Shuai Wang, Kaiwen Zhou, Rui Shao, Liqiang Nie, Yasheng Wang, Jianye Hao, Jun Wang, and Kun Shao. SPA-Bench: A comprehensive benchmark for smartphone agent evaluation. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025b. arXiv:2410.15164.

Yanda Chen, Ruiqi Zhong, Narutatsu Ri, Chen Zhao, He He, Jacob Steinhardt, Zhou Yu, and Kathleen McKeown. Do models explain themselves? Counterfactual simulatability of natural language explanations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 7880–7904. PMLR, 2024. URL https://proceedings.mlr.press/v235/chen24bl.html.

Yu-Neng Chuang, Guanchu Wang, Chia-Yuan Chang, Ruixiang Tang, Shaochen Zhong, Fan Yang, Andrew Wen, Mengnan Du, Xuanting Cai, Vladimir Braverman, and Xia Hu. FaithLM: Towards faithful explanations for large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3802–3824, Rabat, Morocco, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.eacl-long.177. URL https://aclanthology.org/2026.eacl-long.177/.

Vanya Cohen and Raymond Mooney. MET-Bench: Multimodal entity tracking for evaluating the limitations of vision-language and reasoning models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2502.10886.

Iñaki Dellibarda Varela, Pablo Romero-Sorozabal, Eduardo Rocon, and Manuel Cebrian. Rethinking the Illusion of Thinking, 2025. URL https://arxiv.org/abs/2507.01231.

Deqing Fu, Ruohao Guo, Ghazal Khalighinejad, Ollie Liu, Bhuwan Dhingra, Dani Yogatama, Robin Jia, and Willie Neiswanger. IsoBench: Benchmarking multimodal foundation models on isomorphic representations. In *First Conference on Language Modeling (COLM)*, 2024. URL https://arxiv.org/abs/2404.01266.

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 physical intelligence under partial observations. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2512.23328.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Tobias Gerstenberg, Noah D. Goodman, David A. Lagnado, and Joshua B. Tenenbaum. A counterfactual simulation model of causal judgments for physical events. *Psychological Review*, 128(5): 936–975, 2021. doi: 10.1037/rev0000281.

GLM-V Team. GLM-4.5V and GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2025. URL https://arxiv.org/abs/2507.01006.

Gonzalo Gonzalez-Pumariega, Leong Su Yean, Neha Sunkara, and Sanjiban Choudhury. Robotouille: An asynchronous planning benchmark for LLM agents. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. arXiv:2502.05227.

Google DeepMind. Gemini 3.8 Flash model card. https://deepmind.google/models/model-cards/gemini-3-8-flash/, 2026. Accessed 2026-09-26.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Andreas M. Hinz, Sandi Klavžar, and Ciril Petr. *The Tower of Hanoi – Myths and Maths*. Birkhäuser, Cham, 2 edition, 2018. ISBN 978-3-319-73778-2. doi: 10.1007/978-3-319-73779-9.

Anna A. Ivanova. How to evaluate the cognitive abilities of LLMs. *Nature Human Behaviour*, 9(2): 230–233, 2025. doi: 10.1038/s41562-024-02096-z.

Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. OmniSpatial: Towards comprehensive spatial reasoning benchmark for vision language models. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2506.03135.

Changsoo Jung, Jack Fitzgerald, Ethan Seefried, Mariah Bradford, and Nathaniel Blanchard. Spatial Amsan: A benchmark for perception-grounded spatial reasoning and action evaluation in egocentric manipulation. In *Computer Vision – ECCV 2026*, Lecture Notes in Computer Science, pp. 565–582. Springer, 2026. doi: 10.1007/978-3-032-37261-1_32.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Paul Saldyt, and Anil B. Murthy. Position: LLMs can't plan, but can help planning in LLM-Modulo frameworks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 22895–22907, 2024.

Najoung Kim and Sebastian Schuster. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3835–3855. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.213. URL https://aclanthology.org/2023.acl-long.213/.

Kimi Team. Kimi-VL technical report, 2025. URL https://arxiv.org/abs/2504.07491.

K. Kotovsky, J. R. Hayes, and H. A. Simon. Why are some problems hard? Evidence from Tower of Hanoi. *Cognitive Psychology*, 17(2):248–294, 1985. doi: 10.1016/0010-0285(85)90009-X.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2505.06120. Oral.

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. *arXiv preprint arXiv:2307.13702*, 2023. URL https://arxiv.org/abs/2307.13702.

A. Lawsen. Comment on The Illusion of Thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity, 2025. URL https://arxiv.org/abs/2506.09250.

Belinda Z. Li, Zifan Carl Guo, and Jacob Andreas. (How) do language models track state? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34429–34452. PMLR, 2025. URL https://proceedings.mlr.press/v267/li25r.html.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.13382. Oral (notable top 5%).

Linjie Li, Mahtab Bigverdi, Jiawei Gu, Zixian Ma, Yinuo Yang, Ziang Li, Yejin Choi, and Ranjay Krishna. Unfolding spatial cognition: Evaluating multimodal models on visual simulations. In *International Conference on Learning Representations*, 2026.

Jingli Lin, Chenming Zhu, Runsen Xu, Xiaohan Mao, Xihui Liu, Tai Wang, and Jiangmiao Pang. OST-Bench: Evaluating the capabilities of MLLMs in online spatio-temporal scene understanding. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025), Datasets and Benchmarks Track*, 2025. URL https://arxiv.org/abs/2507.07984.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *International Conference on Learning Representations*, 2023.

Xinyi Lu, Christos Bechlivanidis, and Tobias Gerstenberg. Causal judgments in complex situations. In G. Boleda, I. Dautriche, M. Maldonado, C. Saldana, and T. Torrent (eds.), *Proceedings of the 48th Annual Conference of the Cognitive Science Society*, pp. 2064–2070, 2026. URL https://escholarship.org/uc/item/8fj8m8ww.

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35492–35506. PMLR, 2024. URL https://proceedings.mlr.press/v235/merrill24a.html.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*, 2024. URL https://arxiv.org/abs/2411.00640.

Jacob M. Montgomery, Brendan Nyhan, and Michelle Torres. How conditioning on posttreatment variables can ruin your experiment and what to do about it. *American Journal of Political Science*, 62(3):760–775, 2018. doi: 10.1111/ajps.12357.

OpenAI. GPT-5.6 system card. https://deploymentsafety.openai.com/gpt-5-6, June 2026. Covers GPT-5.6 Sol, Terra and Luna. Accessed 2026-09-26.

Daniel M. Oppenheimer, Tom Meyvis, and Nicolas Davidenko. Instructional manipulation checks: Detecting satisficing to increase statistical power. *Journal of Experimental Social Psychology*, 45 (4):867–872, 2009. doi: 10.1016/j.jesp.2009.03.009.

Turan Orujlu, Christian Gumbsch, Martin V. Butz, and Charley M. Wu. Causal process models: Reframing dynamic causal graph discovery as a reinforcement learning problem. In *Proceedings of the Fifth Conference on Causal Learning and Reasoning*, volume 323 of *Proceedings of Machine Learning Research*, pp. 1450–1480. PMLR, 2026. URL https://proceedings.mlr.press/v323/orujlu26a.html.

Devin Pereira and Willem Zuidema. Transformers struggle to use their emergent world models: Revisiting the Tower of Hanoi, and the Illusion of Thinking, 2026. URL https://arxiv.org/abs/2608.07077.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2006–2017, Mexico City, Mexico, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.130. URL https://aclanthology.org/2024.findings-naacl.130/.

Zekun Qi, Wenyao Zhang, Yufei Ding, Runpei Dong, Xinqiang Yu, Jingwen Li, Lingyun Xu, Baoyu Li, Xialin He, Guofan Fan, Jiazhao Zhang, Jiawei He, Jiayuan Gu, Xin Jin, Kaisheng Ma, Zhizheng Zhang, He Wang, and Li Yi. SoFar: Language-grounded orientation bridges spatial reasoning and object manipulation. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2502.13143.

Qwen Team. Qwen3.5-9B, model card. https://huggingface.co/Qwen/Qwen3.5-9B, 2026a.

Qwen Team. Qwen3.8-27B, model card. https://huggingface.co/Qwen/Qwen3.8-27B, 2026b.

Remyx AI. SpaceQwen2.5-VL-3B-Instruct, model card. https://huggingface.co/remyxai/SpaceQwen2.5-VL-3B-Instruct, 2025a.

Remyx AI. SpaceThinker-Qwen2.5VL-3B, model card. https://huggingface.co/remyxai/SpaceThinker-Qwen2.5VL-3B, 2025b.

Allen Z. Ren, Brian Ichter, and Anirudha Majumdar. Thinking forward and backward: Effective backward planning with large language models, 2024.

Tomas Rokicki and Morley Davidson. God's number is 26 in the quarter-turn metric. https://cube20.org/qtm/, 2014. Web page, result dated August 2014. Accessed 2026-09-21.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6c0e99d736da621403018ca7b32b1a4d-Abstract-Conference.html.

Chufan Shi, Cheng Yang, Yaokang Wu, Linghao Jin, Bo Shui, Taylor Berg-Kirkpatrick, and Xuezhe Ma. Are VLMs seeing or just saying? Uncovering the illusion of visual re-examination. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2605.15864. Oral.

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. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/9b26ad15462c81548c0689188d2e8018-Abstract-Conference.html.

Herbert A. Simon. The functional equivalence of problem solving skills. *Cognitive Psychology*, 7 (2):268–288, 1975. doi: 10.1016/0010-0285(75)90012-2.

Kexian Tang, Junyao Gao, Yanhong Zeng, Haodong Duan, Yanan Sun, Zhening Xing, Wenran Liu, Kai Chen, and Kaifeng Lyu. LEGO-Puzzles: How good are MLLMs at multi-step spatial reasoning? In *Computer Vision – ECCV 2026*, Lecture Notes in Computer Science, pp. 1–19. Springer Nature Switzerland, 2026. doi: 10.1007/978-3-032-37553-7_1.

Zhenwei Tang, Difan Jiao, Blair Yang, and Ashton Anderson. SEAM: Semantically equivalent across modalities benchmark for vision-language models. In *Second Conference on Language Modeling (COLM)*, 2025. URL https://arxiv.org/abs/2508.18179.

Shubham Toshniwal, Sam Wiseman, Karen Livescu, and Kevin Gimpel. Chess as a testbed for language model state tracking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36 (10):11385–11393, 2022. doi: 10.1609/aaai.v36i10.21390. URL https://ojs.aaai.org/index.php/AAAI/article/view/21390.

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

Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasović, and Yonatan Belinkov. Measuring chain of thought faithfulness by unlearning reasoning steps. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9935–9960, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.504. URL https://aclanthology.org/2025.emnlp-main.504/.

Keyon Vafa, Justin Y. Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. Evaluating the world model implicit in a generative model. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2f6a6317bada76b26a4f61bb70a7db59-Abstract-Conference.html.

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. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track*, 2023. arXiv:2206.10498.

Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Chi Wan, Yiping Lu, Zhengyuan Yang, Lijuan Wang, Ranjay Krishna, Jiajun Wu, Li Fei-Fei, Yejin Choi, and Manling Li. VAGEN: Reinforcing world model reasoning for multi-turn VLM agents. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025a. doi: 10.52202/085713-5754. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/fc6688d75adde86b9df910769c1d02e3-Abstract-Conference.html.

Qineng Wang, Baiqiao Yin, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, Saining Xie, Jiajun Wu, Li Fei-Fei, and Manling Li. Spatial mental modeling from limited views. In *The Fourteenth International Conference on Learning Representations*, pp. 82559–82622, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/858de961874816f894285ec02e50c7fe-Abstract-Conference.html.

Ruoyao Wang, Graham Todd, Ziang Xiao, Xingdi Yuan, Marc-Alexandre Côté, Peter Clark, and Peter Jansen. Can language models serve as text-based world simulators? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–17, 2024. doi: 10.18653/v1/2024.acl-short.1. URL https://aclanthology.org/2024.acl-short.1/.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025b. URL https://arxiv.org/abs/2508.18265.

Yue Wu, Xuan Tang, Tom M. Mitchell, and Yuanzhi Li. SmartPlay: A benchmark for LLMs as intelligent agents. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.01557.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_Thinking_in_Space_How_Multimodal_Large_Language_Models_See_Remember_CVPR_2025_paper.html.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/54dd9e0cff6d9214e20d97eb2a3bae49-Abstract-Conference.html.

APPENDIX

A Protocol details **17**
A.1 Problem set. 17
A.2 The first message 18
A.3 Comprehension gate 18
A.4 How forks work 18
A.5 Probe details 18
A.6 Intervention details 20
A.7 Replies and parsing 20
A.8 Admission test 20
A.9 Local models 20
A.10 Why rendered pictures 21
A.11 Logs 21

B Full results **21**
B.1 Robustness 21
B.2 Run outcomes 21
B.3 Interventions in detail 22
B.4 Checks on shortcuts, direction and belief 22
B.5 Which step a failure points to 23
B.6 Encoding details 23
B.7 Open weight models 23
B.8 Thinking on and off 24
B.9 Tables and figures. 24

C Prompts **24**

D Qualitative cases. **25**

E Extended related work **27**

A PROTOCOL DETAILS

This appendix gives the details that Section 3 leaves out. The prompts in Appendix C are copied from the library that builds every message, so they are exactly what a model receives.

A.1 PROBLEM SET

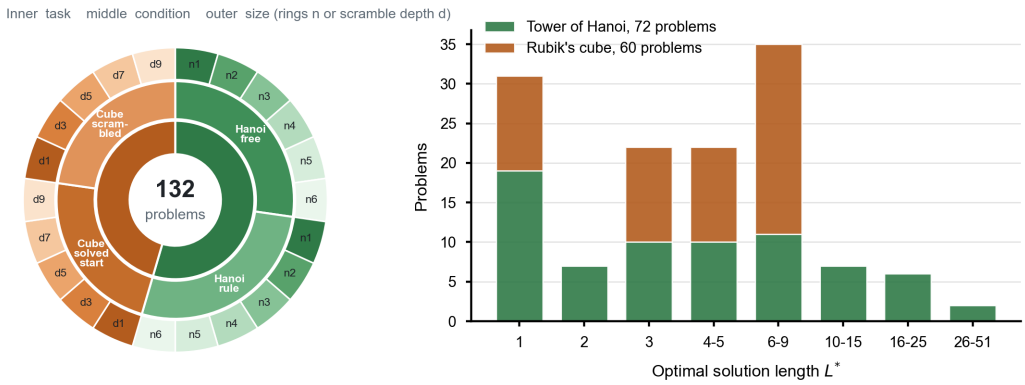


Figure 2: **The problem set.** Left. 132 problems in 22 cells of six problems each. Right. The optimal solution length L^* of every problem.

Figure 2 shows the problem set. A seeded generator builds the problems, so the set can be rebuilt bit for bit. For Hanoi, we sample legal initial and goal states and compute the optimal solution length with breadth-first search over the full state graph. For the cube, we generate targets by applying a fixed number of quarter turns and verify their distance with two-sided search. A problem is kept only if its optimal length equals the intended depth. A second program with its own search code checks every file again and shares no code with the generator. It performs 36 checks before the files are frozen under a checksum list. The runner aborts if any file no longer matches that list, and each run records the list checksum.

The Tower of Hanoi is a standard task in the study of human problem solving (Simon, 1975; Kotovsky et al., 1985). Our cube move names differ from the standard notation, so solving routines written in that notation do not carry over directly. The 132 files contain 126 distinct initial–goal pairs because the two Hanoi rule conditions are identical for one-ring problems. We set the budget to $B = \max(2L^*,\ L^* + 2)$. A wrong move requires two extra turns, one for the mistake and one to undo it. The floor $L^* + 2$ thus lets even the shortest problems survive one mistake, and twice the optimal length leaves room for several mistakes on longer problems. Earlier work sets step limits at 1.5 times the optimal length (Gonzalez-Pumariega et al., 2025) or at twice the length of a human run (Chen et al., 2025b). Every result can be recomputed for any smaller budget from the same logs.

Table 5: The two tasks.

	Tower of Hanoi	Rubik's cube
State	rings stacked on three poles	54 stickers, 20 movable pieces
Move	one ring from one pole to another	one quarter turn of one face
Legal moves	few under Rule	all twelve, always
How an error shows	a rule is broken	the goal gets farther away
Difficulty dial	1 to 6 rings, Rule or Free	1, 3, 5, 7 or 9 turns, solved or scrambled start
L^* in our set	1 to 51	equal to the depth

A.2 THE FIRST MESSAGE

Appendix C shows the system prompt and the first user message in every encoding, with examples from several Hanoi problems and one cube problem. The image-and-text prompt refers to the rendered pictures as images, and the models receive that wording. The worked example move is selected by a seeded rule from all well-formed moves except optimal first moves. It therefore demonstrates only the required reply format without revealing the state or hinting at the solution. In the encodings with pictures each picture follows a caption that names the state and the view. A short legend ties each ring letter or sticker letter to its color in the pictures.

A.3 COMPREHENSION GATE

Human experiments use instruction checks for a related purpose (Oppenheimer et al., 2009). Our checks run on a fork. If they stayed in the main history, the model's own correct answers would write the state into that history. In the image encoding, they would give away more than half of all ring positions. No check includes an optimal first move, preventing it from revealing the solution. Admission is based only on the first response because allowing a second attempt would increase the chance of passing by guessing.

A.4 HOW FORKS WORK

Figure 3 shows the forks of one real run. A fork copies the main-run messages up to a given turn and adds one question or intervention. The reasoning probe is the only exception, with two questions on the same fork. Each fork is sent as a new request containing the full copied conversation, so the model processes it in the same form as the main run. The main run itself holds only the first message, the model's moves, the turn requests and the "ok." replies. Probe forks and the two forks at the first failure ask one question and are then thrown away, so they never change the run. The Remind forks continue as runs of their own, one after the true state and one after the placebo line. The Mask forks are created after the run ends and revisit an earlier decision under a modified history. Forks therefore serve two purposes. They measure the state at any turn without leaving a trace in the run, and they let us change the history in a controlled way to test a cause.

A.5 PROBE DETAILS

Table 6: Probe families. Every probe runs on a fork and has a key from the simulator.

Family	What it asks	When it runs	Score
Recall	own action, moved piece or place of a piece at lag 1 to 3	checkpoints	correct option
Recency	the turn at which a piece last moved, or never	checkpoints	correct option, error in turns
Configuration	the full current state	turn zero and checkpoints	share of objects right
Reasoning	was the last move legal or did it bring the goal closer, then the next move	every turn	correct option
Belief	the full state before and after the first failure	first failure	share of objects right

Recall. The lag cycles through 1, 2 and 3 over the checkpoints of a run, and the problem number sets where the cycle starts. Each run cycles through every lag, and across problems each lag is evaluated at every checkpoint. Random sampling could omit a lag in short runs or confound a lag with a particular checkpoint. The schedule is fixed before any model runs, so every model gets the same questions. Both recall forms that exist for a task are asked at every checkpoint on separate forks, since a single form per checkpoint would tie each form to one lag. A fixed rule picks one form for the main grid at each checkpoint, and answers to the other form are kept apart and never pooled with it.

Configuration. On the cube the score leaves out the six center pieces, since the whole cube is never turned and they never move. An exact score requires every sticker to be correct. Replies that cannot

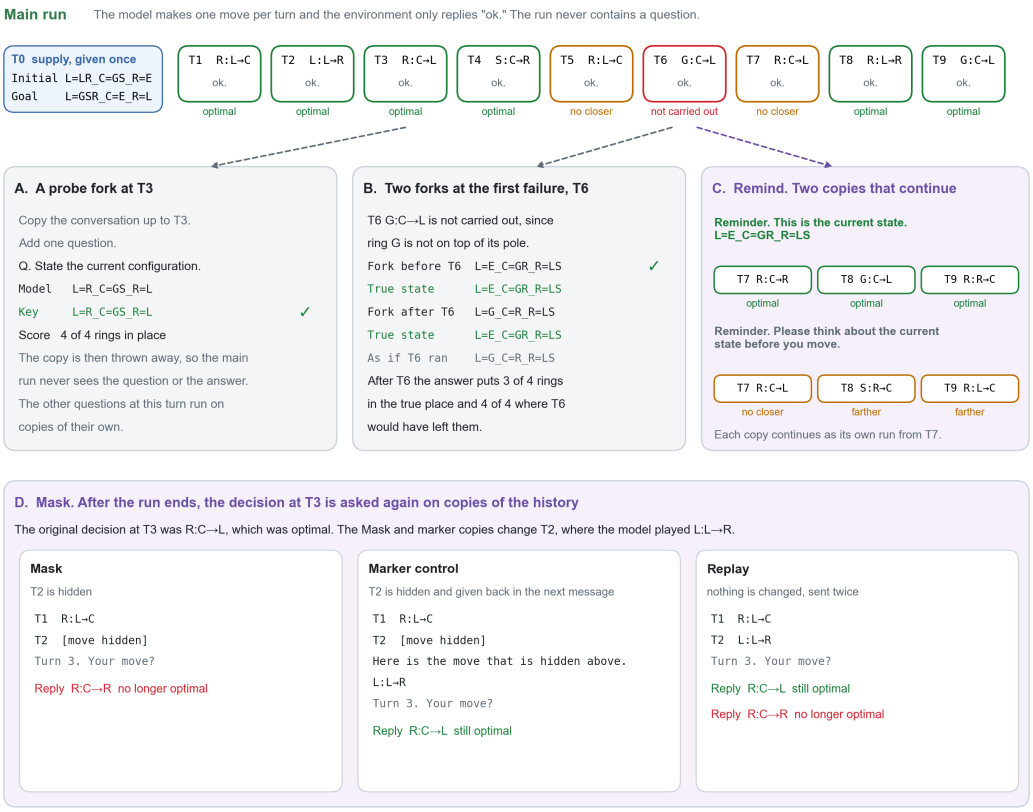


Figure 3: **How forks work, in one run.** Gemini 3.8 Flash with thinking low on a Hanoi problem with 4 rings in the symbolic encoding. Every move and reply comes from the logs, and the resulting state is computed from them. (A) A probe fork asks its question and is then discarded. (B) At the first failure, two forks ask for the state immediately before and after the failing move. (C) Remind continues two copies from the first failure, one with the true state and one with the placebo line. (D) Mask asks again for an earlier decision on copies with a changed history, next to the marker control and a replay.

be parsed as a single move are not carried out but still consume a turn. Up to L^* this happens in at most 3 percent of the moves of the proprietary models. It is common for some open weight models (Table 17).

Next move options. In the reasoning probe, the legality or distance question comes first since showing the next-move options would reveal its answer. The next move question offers one option from each of four classes. These are an optimal move, a move that keeps the distance to the goal, a move that increases it and an illegal move. At some states a class has no member. A fixed rule fills that slot with a move that increases the distance, and each option's class is logged with the item. Under Free, no move violates the size rule, so the illegal slot is always replaced this way. When no such move exists either, as with one ring, the item has fewer options and is scored apart. On the cube every quarter turn changes the distance to the goal by exactly one. Each of the twelve moves thus either brings the goal closer or moves it away, and none keeps the distance or is illegal. The options are one optimal move and three moves that increase the distance. Chance is then 25 percent, and Flash picks the optimal move less often than that.

Items solvable without the state. Some next move items can be solved from the option list and the last move in the record alone. Two cues make this possible on Hanoi. Every option moves a ring that is currently on top of its pole, so the option set reveals the top rings. Under Rule, this can expose the illegal option when it places a ring on a smaller top ring shown by another option. Because a shortest path never moves the same ring twice in succession, options that repeat the previous ring

can also be removed. For example, say the model just moved ring R. If two options move R again and a third is the illegal move, only the fourth survives. We mark every item that these two cues solve and leave it out of the next move score.

What the options reveal. The key is the letter of the correct option, and a seeded shuffle puts it on each letter about equally often. A model that always selects the same letter scores at chance, so letter bias adds noise without improving accuracy. Option lists within a fork also reveal true state information. For example, next-move options include only rings that are currently on top of their poles. Using random moves would hide this structure, make the correct option easier to identify, and fail to test the distinction between moves that reduce the distance to the goal and those that do not. These facts never reach the main run, since the fork is thrown away, but they can help the answer to that one question.

A.6 Intervention details

Shi et al. (2026) find that an outside instruction restores the use of an image when self-prompted looking does not, and Remind extends this logic to a state that is no longer in view. The placebo line matches the reminder in position and opening words but contains no state information. In image-based encodings, the reminder includes new renders of the current state produced by the same renderer as the supplied images. For every state on a stored shortest solution, these renders match the corresponding stored image pixel for pixel. At the first failure a third fork asks the model why it made the move. We do not score that answer as evidence and only compare it with the cause found by the interventions (Chen et al., 2024). For Mask the simulator tries every move that could stand in the gap and keeps those that leave the rest of the record possible. Mask uses only the part of a run before any rejected move, and every Mask reply is scored at the true state because only the record changed. A mask requires at least two moves that fit the gap, preventing the model from reconstructing the record uniquely. The marker control restores the hidden move in a user message rather than in the model's original turn. Changes in the decision are interpreted relative to the replay change rate. On the cube, every legal move fits a gap and changes the state, so nearly every earlier turn constrains the decision. The cube therefore has no placebo, and cube masks are read against the marker and the replay.

A.7 Replies and parsing

One parser reads replies in every encoding. For a lettered question it accepts a single letter with harmless marks around it. These are bold or italic markup, quotation marks, a final period and lead-in words such as "Option" or "The answer is". The parser accepts only the option letter, not the option text. Replies that violate this format are logged for a secondary analysis. For a move, it removes the same harmless marks and then applies the strict move grammar. A reply that holds a move and other words counts as unreadable in every encoding. Every accepted form is logged with the reply.

A.8 Admission test

Before a model enters the study it must pass a short test with about a dozen cheap calls. The test checks four things. The model can process a history it did not generate, answer a question added to a copy of that history, and continue from a version in which an earlier turn has been edited. It accepts several pictures in the first message. The test also sends the same request three times and records how often the reply is identical. That number is a rate and not a pass mark, since replies can differ between identical requests (Atıl et al., 2025).

A.9 Local models

SpaceQwen is trained from Qwen2.5-VL 3B. SpaceThinker is trained from VLAA-Thinker (Chen et al., 2025a), which is itself trained from Qwen2.5-VL 3B. The best specialized model in OmniSpatial, SoFar (Qi et al., 2025), has released no weights for its vision-language model. Local models without thinking use greedy decoding with a fixed seed, so that their runs can be replayed. Local models with thinking sample at temperature 0.6 and top-p 0.95 with a fixed seed. This is one setting for all thinking models, and it is lower than the temperature of 1.0 that the Qwen model cards

give for thinking on general tasks (Qwen Team, 2026a;b). Settings with thinking run three of the six problems in every cell, or 66 problems. Gemma 4 31B, Qwen3.8 27B and Qwen3.5 9B run their thinking settings in the symbolic and image encodings only. A model's own suggested system prompt is not used. Text that a local model marks as hidden reasoning is removed before its reply is read or replayed. For the proprietary models sampling settings stay at each provider's default.

A.10 Why rendered pictures

The pictures are rendered and not photographed, for three reasons. First, every probe needs an exact key, and a renderer draws exactly the state that the simulator holds. Second, the reminder in the encodings with pictures has to show the true current state at the first failure. We render the states on a stored shortest solution of each problem in advance, but after the model's own moves the current state is usually not one of them. A renderer draws any state on demand, while a photograph would need someone to set up that state by hand during the run. Third, a change of picture would mostly affect reading, and SSD measures that step on its own. In the main run the pictures enter only at turn zero, and the configuration probe at turn zero measures how well they were read. The control in Section 4.5 shows that the main findings do not rest on how the pictures were read. Clean renders with a fixed view and even light are likely close to a best case for reading. We expect photographs to lower reading at turn zero, most of all on the cube. We have not tested this.

A.11 Logs

Every call is logged with its time and its delay. The log also holds the reply, the reason the model stopped and the token counts. Every score line carries the options, the key and the chosen option with its class. It carries the lag, the form and the probed piece. It also names the problem and its cell, the encoding, the direction and L^*. A copy of the message library is saved with every run. Pictures are logged as file paths with checksums and never as image data. The proprietary models may change over time, so we record the model version that each reply reports.

B Full results

B.1 Robustness

The gate is itself noisy. We ran the excluded runs again, which asked the gate a second time. Of the runs that failed the first time, 30 percent passed the second time. This was 67 of 224 runs for Terra and 11 of 35 for Flash. They stay excluded, since a run is admitted on its first answer. With every run included, Terra solves 4 to 10 points fewer Hanoi runs in each encoding, and the order of the encodings does not change. The cube results and the Flash results change little. Table 13 repeats Table 3 on every complete run. Move recall stays above place recall and above the configuration in all 16 cells of model, task and encoding.

B.2 Run outcomes

Table 12 splits the Hanoi configuration probe by whether every move so far was carried out. Runs with a rejected move differ in other ways too, so this split describes and does not isolate a cause. The table also scores only the displaced objects, those that are no longer where they started. Repeating the initial state gets none of them right, so this score shows how much of the change the model followed. On the cube Terra gets few displaced pieces right even after one move. Flash gets 92 percent right after one move and 22 percent after two or three. The column for one move holds only the problems of depth one, so the four cube columns do not cover the same problems. Both full grid models answer at least 90 percent of the recency questions in every encoding. Rejected moves are common on Hanoi. Of the moves up to L^*, 53 percent of Terra's and 31 percent of Flash's broke the rules and were not carried out, and 57 and 34 percent of the runs have at least one such move. The recency questions include items on forward Hanoi runs in which the model tried a move of the probed ring after the ring last moved and that move was rejected. On those items Terra scores 8 percent $[2, 15]$ and Flash 47 percent $[29, 67]$, against 95 and 99 percent on the other forward items. Terra then names the turn of the rejected move in 59 percent of them. The next move needs the state and a plan, and it is the weakest probe on Hanoi. With thinking high Flash keeps 84 percent

of the cube pieces in the symbolic encoding, on 10 problems. On the cube place recall is 24 to 73 percent. Terra keeps 64 to 77 percent of the Hanoi rings in place and Flash 87 to 93 percent. On cube problems with $L^* \geq 7$ the configuration is down to 10 to 36 percent by turn 3, and only the image and text encoding of Flash stays above the copy line there at every checkpoint. On Hanoi Terra is below the copy line from turn 3 on, while Flash stays above it in every encoding. On the cube Flash picks an optimal next move on 11 to 24 percent of the items, below the chance level of 25 percent. Between directions Terra's place recall on Hanoi differs by 34 points $[17, 50]$, and its recency on Hanoi is higher backward by 15 points $[9, 21]$. Runs fail early. On the longer cube problems 88 to 100 percent of the runs fail at turn 1, because the first move already increases the distance to the goal. Over all problems the first failure comes at 23 to 51 percent of L^*. Terra solves 36 to 42 percent of its Hanoi runs and 3 to 10 percent of its cube runs. Flash solves 62 to 66 percent and 20 to 21 percent. Each range spans the four encodings. Terra picks an optimal next move on 33 to 48 percent of the Hanoi items and on 26 to 46 percent of the cube items, and Flash on 58 to 62 percent of the Hanoi items. On Hanoi forward runs fail later than backward runs, by 22 to 26 percent of L^*. On the cube the first failure comes at the same point in both directions (Table 9).

B.3 Interventions in detail

In the encodings with pictures every Remind interval excludes zero except Terra's image encoding, at $+3$ $[-1, 7]$. Table 4a pools the two tasks. Split by task, Flash gains 22 to 24 points over the placebo on the cube in each encoding with pictures, and every interval excludes zero. On Hanoi Flash gains 33 points $[18, 47]$ in the image encoding. On Hanoi in the image and symbolic encoding and the image and text encoding the placebo alone raises Flash by 17 points $[4, 30]$ and 16 points $[0, 31]$ against runs with no reminder. There the reminder adds only 14 points $[-6, 34]$ and 3 points $[-13, 20]$ over the placebo. For Terra the gains stay below 13 points in every cell. In the symbolic encoding the reminder gains at most 7 points against runs with no reminder, and every interval includes zero. When Remind gives the true state in symbols, Terra's answer one move later scores 60 percent on the cube and is exact in 6 of its 95 runs, or 7 percent as a mean over problems. On Hanoi Flash loses 10 points more $[4, 16]$ when the hidden move matters than when it does not. Table 15 gives the pilot settings.

B.4 Checks on shortcuts, direction and belief

We run six checks on the admitted runs. First, always answering the latest completed turn would score 17 to 37 percent on recency. The models score 91 to 100 percent, pooled over encodings. Second, on forward Hanoi runs with symbolic moves, reading a ring's pole off the record as if every move had been carried out would score 85 percent on Terra's place recall items and 88 percent on Flash's. Terra scores 68 percent $[57, 79]$, and Flash scores exactly 88 percent. Third, on Hanoi probes before any rejected move, forward runs keep 14 points $[7, 21]$ more of the configuration than backward runs for Terra and 4 points $[2, 6]$ more for Flash. Fourth, when the first failure on Hanoi is a rejected move, the configuration written just before it is 97 percent $[95, 99]$ right for Terra and 98 percent $[96, 99]$ for Flash. Just after it these fall to 69 percent $[65, 72]$ and 79 percent $[74, 83]$. Fifth, before any rejected move the Hanoi configuration falls by at most 4 points with the number of moves carried out. Terra keeps 91 to 95 percent for up to 8 moves and Flash 98 to 100 percent for up to 16. Sixth, the cube direction gap holds for both start conditions. For Terra forward runs keep 12 points $[9, 16]$ more of the configuration from a solved start and 13 points $[9, 17]$ from a scrambled start. For Flash the gaps are 9 points $[4, 16]$ and 6 points $[1, 11]$.

Three more counts concern Hanoi runs whose first failure is a rejected move, at any turn. In the image and text encoding the answer names only the pole of each ring, so there a state counts as exact when every ring is on its right pole. The state written just before was exact in 90 percent $[85, 94]$ of Terra's failures and 85 percent $[78, 92]$ of Flash's. In forward runs under Free the question after a move asks whether it reduced the distance, and under Rule it asks whether the move was legal. On Rule runs, asked right after, Terra calls the move legal in 83 percent $[74, 91]$ of these failures and Flash in 37 percent $[26, 47]$. Over all turns under Rule, Terra calls 86 percent $[80, 91]$ of legal and 71 percent $[63, 77]$ of illegal moves legal. Flash calls 92 percent $[90, 94]$ and 53 percent $[46, 59]$ legal. For Terra the answer after a failure thus mostly reflects a general tendency to say yes. We then take the cases where the true state and the state with the move carried out differ. Both matches count a ring as right when it is on the right pole, as the configuration score does. The state written just

after such a move matches the state with the move carried out on every ring in 65 percent $[52, 77]$ of Terra's cases and 41 percent $[31, 50]$ of Flash's. It matches the true state in 3 percent $[0, 9]$ and 31 percent $[22, 41]$. In forward runs Terra's answer matches the state with the move carried out in 78 percent $[65, 89]$ of cases and Flash's in 67 percent $[58, 77]$, against 0 and 29 percent $[19, 39]$ for the true state.

B.5 Which step a failure points to

A fixed rule assigns every first failure of an admitted run to one step. The failure points to *reading* if the configuration written at turn zero did not place every object right. It points to *updating* if it did but the state written just before the failing move was not exact. It points to *choosing* if that state was exact and the move still failed. Noticing is measured after a rejected move and is not part of the rule. Both probes run on forks, so an exact state means that the model could write it when asked. For the image and text encoding, exact is defined as in Appendix B.4. On Hanoi reading accounts for at most 1 percent of first failures, updating for 12 and 13 percent and choosing for 87 percent for both Terra and Flash. On the cube the shares are 12 and 14 percent for reading, 4 and 7 percent for updating and 84 and 79 percent for choosing. Most cube runs fail at turn 1, before any update, and the loss of the cube state that Table 12 shows comes after the first failure.

B.6 Encoding details

After turn zero the gap from reading shrinks. On the cube configuration the image encoding stays 7 points $[1, 14]$ behind the symbolic encoding for Terra and 18 points $[11, 24]$ for Flash. On the configuration the other encodings stay within 10 points of the symbolic encoding, and the sign of the difference depends on the model. On Hanoi every encoding reads the supply correctly, and later every encoding stays within 9 points of the symbolic encoding on the configuration. Against the symbolic encoding, words beside the pictures help Terra on Hanoi by 9 points $[3, 14]$ on the configuration. They help Flash on the cube by 9 points $[2, 17]$ on the configuration and by 19 points $[6, 32]$ on place recall. The same words cost Terra 19 points $[1, 35]$ of place recall on the cube. We therefore do not rank the encodings in general. The fourth encoding changes the form of questions and replies as well as the supply, so a difference from the image encoding can come from either. We did not run a control that separates the two, so the fourth encoding is read with this caveat.

B.7 Open weight models

With symbols the open weight settings place up to 100 percent of the rings and of the pieces at turn zero. In the symbolic encoding Qwen3.8 27B, Gemma 4 31B and Qwen3.5 9B with thinking place more rings than a copy of their initial state would, and the first two also more cube pieces. Every other setting places no more objects than that copy. We report the interventions for the proprietary models only. Of the eight cells of task and encoding, the gap to Qwen2.5-VL 3B excludes zero in six for SpaceQwen and in all eight for SpaceThinker. SpaceThinker was trained through VLAA-Thinker, and it samples while its base decodes greedily. In the symbolic encoding the open weight models without thinking recall 60 to 91 percent of their Hanoi moves and 28 to 51 percent of the ring places. On the cube they recall 45 to 87 percent of their moves and 26 to 49 percent of the piece places. SpaceThinker is the one model whose move recall on the cube is below chance, in the symbolic and image encodings. Move recall is above both place recall and the configuration on both tasks in 11 of the 14 settings. The exceptions are Qwen3.8 27B and Qwen3.5 9B with thinking and SpaceThinker. A copy of the model's own initial state would place 44 to 95 percent of the rings and 27 to 82 percent of the pieces. This copy line is often high because many moves were rejected or unreadable, so the true state stayed near the start. In the symbolic encoding and without thinking the next move is at 36 to 46 percent on Hanoi against a chance level of 35. With thinking Qwen3.8 27B and Gemma 4 31B reach 77 and 75 percent. Some models often reply in a form that the parser cannot read (Table 17). For Kimi-VL this holds for 35 to 69 percent of the moves, and for Qwen3.8 27B with thinking for 71 percent of its cube moves. A reply that is not a move is never a first failure, so we do not compare failure times for these models. On the admitted runs InternVL recalls 75 percent of its Hanoi moves and gets 41 percent of the rings right in the configuration on 29 problems. GLM with thinking reaches 88 and 34 percent on 24 problems. Of the runs that did not finish, 24 hit the context limit of the server, 19 for Gemma 4 31B with thinking and 5 for SpaceThinker. Another 51 runs of Kimi-VL with thinking stopped at the gate under an earlier rule.

B.8 THINKING ON AND OFF

We compare each open weight model that has both settings with thinking on and off on the 66 problems in the symbolic and image encodings, which every thinking setting ran, over all runs. The comparison scores the configuration on the displaced objects only, because rejected or unreadable moves keep the true state near the start and so inflate the plain configuration score. On Hanoi thinking raises the share of displaced rings placed correctly to 95 to 98 percent, by 27 points $[19, 36]$ and 37 points $[27, 46]$ for Qwen3.8 27B in the symbolic and image encodings, and by 7 points $[2, 12]$ and 10 points $[5, 17]$ for Gemma 4 31B. On the cube Gemma gains 52 points $[43, 61]$ and 21 points $[12, 33]$. Qwen3.8 27B with thinking leaves most of its cube moves unreadable, so its cube scores rest on few displaced pieces. For Qwen3.5 9B the Hanoi gains are 15 points $[-2, 31]$ and 19 points $[1, 37]$, and on the cube they are within 6 points of zero. For GLM-4.6V Flash and Kimi-VL, thinking brings no clear gain. On admitted runs in which every move could be read, Qwen3.8 27B and Gemma 4 31B with thinking keep 95 to 100 percent of the Hanoi configuration. There thinking adds 17 points $[6, 30]$ and 38 points $[24, 52]$ of displaced rings for Qwen3.8 27B and 2 points $[0, 6]$ and 6 points $[-3, 19]$ for Gemma 4 31B, whose readable runs without thinking already place 94 and 76 percent. Gemma gains 50 points $[36, 64]$ and 30 points $[12, 49]$ of displaced cube pieces. The comparison also changes decoding, since the thinking settings sample at temperature 0.6 and the others decode greedily. Runs of the thinking settings outside these problems come from earlier starts and are kept in the raw logs only.

B.9 TABLES AND FIGURES

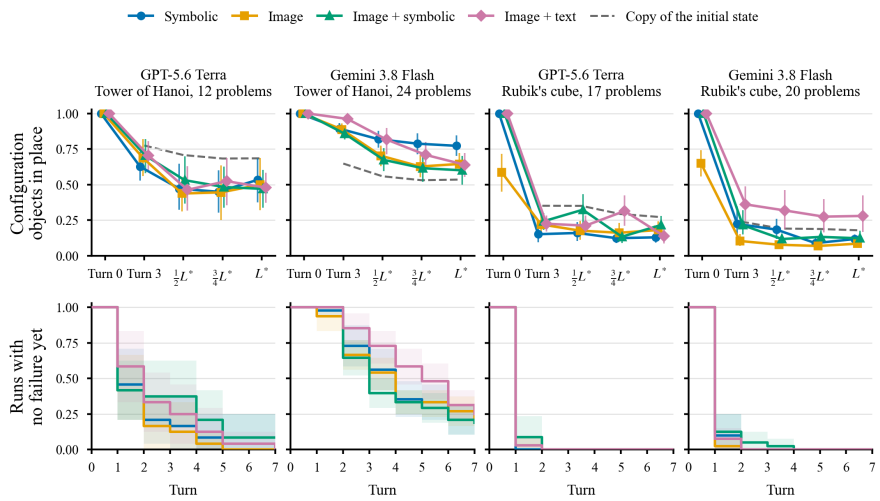


Figure 4: **The state over the run.** Problems with $L^* \geq 7$ only, so every run reaches every turn shown and each line averages over the same problems. Top. Share of objects in place when the model writes out the current state, at the fixed checkpoints. The dashed line is the score of a copy of the initial state on the same questions, pooled over the four encodings. Bottom. Percent of runs with no failure yet. Bars and bands are 95 percent intervals over problems.

C PROMPTS

Figures 5 to 11 show every kind of message that a model receives in a forward run. The text is taken from the requests that the message library built for a keyless test run, so it is exactly what a model reads. A panel shows a whole message or the part of a message that its title names. Pictures appear where they appear in the message. Every request resends the whole conversation, the system prompt included. When the system prompt says that its rules are stated once, it means that the turn requests do not repeat them, since a turn request holds only the turn number and the budget. The ring line of the Hanoi system prompt in Figure 5 names all six rings and says how many of them the problem

Table 7: **Open weight models, all runs.** Runs that failed the gate are included, since the gate excluded many runs of these models. *Gate* is the percent of runs that passed the gate on the first try. $Read_S$ and $Read_I$ are the configuration probe at turn zero in the symbolic and in the image encoding. The other columns are the probes of Table 3 in the symbolic encoding, which every model ran. *Copy* is the score on *Conf.* of a copy of the model's own initial state. It is high because many moves were rejected or unreadable, so the true state often stayed near the start. The small numbers after a name are the Hanoi and cube problems with complete runs. The thinking settings marked † ran 36 Hanoi and 30 cube problems. Dark and light shading mark the best and second best setting in each column. Table 16 gives every encoding and Table 17 the share of replies that could not be read.

	Tower of Hanoi									Rubik's cube								
Model	Gate	$Read_S$	$Read_I$	Move	Place	Rec.	Conf.	Copy	Next	Gate	$Read_S$	$Read_I$	Move	Place	Rec.	Conf.	Copy	Next
Chance	–	–	–	27	33	35	–	–	35	–	–	–	25	25	32	–	–	25
Open weight models																		
Qwen3.8 27B, thinking off 72/60	84	100	94	86	48	79	55	*71*	37	39	100	45	87	49	99	51	*52*	45
Qwen3.8 27B, thinking on† 36/30	91	100	97	90	99	99	98	*50*	77	62	100	50	100	100	100	96	*82*	5
Gemma 4 31B, thinking off 72/60	81	100	95	91	51	90	54	*71*	46	69	100	36	85	38	100	27	*27*	22
Gemma 4 31B, thinking on† 32/30	80	100	93	100	95	100	96	*44*	75	88	92	37	98	81	100	64	*38*	44
Qwen3.5 9B, thinking off 72/60	49	87	60	70	41	57	45	*84*	43	35	97	7	56	35	95	26	*30*	43
Qwen3.5 9B, thinking on† 36/30	59	25	18	85	88	91	56	*53*	54	69	75	9	89	43	88	30	*54*	36
InternVL3.5 14B 72/60	56	84	65	71	48	56	39	*78*	37	18	100	34	79	44	92	36	*43*	45
GLM-4.6V Flash, thinking off 72/60	44	89	18	60	42	38	46	*83*	40	4	79	30	63	35	70	32	*45*	22
GLM-4.6V Flash, thinking on† 36/30	77	35	33	81	44	69	33	*74*	38	41	53	33	88	48	92	16	*50*	40
Kimi-VL A3B Instruct 72/60	24	23	23	83	38	57	38	*93*	38	3	42	4	45	26	54	28	*60*	37
Kimi-VL A3B Thinking† 26/27	16	6	2	38	6	38	5	*86*	36	5	12	2	75	28	62	9	*82*	17
Specialized spatial models and their base																		
SpaceThinker Qwen2.5-VL 3B† 36/30	5	35	6	38	20	25	43	*95*	35	1	81	5	17	7	41	38	*52*	30
SpaceQwen2.5-VL 3B 72/60	8	65	8	65	28	19	53	*93*	36	0	61	2	51	30	36	29	*41*	28
Qwen2.5-VL 3B (base) 72/60	11	56	17	73	36	23	53	*92*	37	5	81	5	76	29	55	39	*55*	43

Table 8: **Table 3 for the pilot settings.** They ran 22 problems in the symbolic and image encodings. They are small, and we read them for direction only.

		Tower of Hanoi							Rubik's cube						
Model	Supply	Read	Move	Place	Rec.	Conf.	Next	Avg	Read	Move	Place	Rec.	Conf.	Next	Avg
Chance		–	26	33	36	–	32	–	–	25	25	30	–	25	–
Copy of the initial state		–	–	–	–	38	–	–	–	–	–	–	52	–	–
Proprietary models, pilot grid (22 problems)															
Gemini 3.8 Flash, thinking high	Symbolic	100	100	100	100	100	91	98	100	100	100	100	84	44	85
12/10 problems	Image	100	100	100	100	99	91	98	72	100	78	100	46	50	73
Claude Sonnet 5, effort low	Symbolic	100	88	88	100	84	79	90	100	100	54	90	31	79	69
8/8 problems	Image	94	62	75	100	70	71	76	49	96	36	98	21	60	61
Claude Sonnet 5, thinking off	Symbolic	100	100	100	83	62	34	71	100	100	20	69	53	44	62
6/6 problems	Image	100	100	50	92	62	0	58	49	100	25	75	29	78	60

uses, from 1 to 6. The sentence that defines a lag appears in the system prompt and again at the start of every recall question, so that each question read on its own fork fixes which move lag 1 means. We set this before any model ran.

D QUALITATIVE CASES

Figure 12 shows three real cases from the logs. Case A shows a cube that is misread at turn zero in the image encoding. Case B is a run of Flash on Hanoi with 6 rings under Rule. Just before its fourth move it writes the state exactly. The move would put the sky blue ring on the smaller red ring, so it is not carried out. Asked on a fork just after, the model writes the state as if the move had been carried out, with the larger ring on the smaller one. It also calls the move legal. The state it writes is not the goal state, so it is not simply copying the goal. Case C shows the two copies of the Remind intervention for one run. The copy that receives a picture of the current state makes three optimal moves in its first four turns. The placebo copy makes a rejected move at once, as the original run did.

[System Prompt, Tower of Hanoi]

Poles L, C, R (left, center, right), always written in that order.
State text has the form L=<rings>_C=<rings>_R=<rings>. Each pole lists its rings bottom to top with no spaces. An empty pole is written E.
Rings from largest to smallest are G, L, S, R, O, Y. This problem uses the first 3 of them.
In a move X:Y->Z the first letter is always a ring, the last two are always poles.
One move per turn, format: ring:from->to
Turns are numbered from 1. Every turn request shows the turn number.
Lag 1 is the move you just completed; lag 2 is the move immediately before it; lag 3 the one before that.

Each turn, reply with exactly one move token in the form ring:from->to and nothing else.
A question that supplies lettered options is answered with exactly one capital letter and nothing else.
The letter is the label in front of the option. It is not the text of the option.
A question that asks for the state is answered with the state text and nothing else.
If you are not sure, still give your best answer in the required form. Never send an empty reply. Never ask a question.
Output notation only, no explanation. Replies are pure notation.
This contract is stated here once and is never repeated when a turn is requested.

[System Prompt, Rubik's Cube]

Faces TF (top), DF (down, on the table), FF (front), RF (right), BF (back), LF (left).
Seen from above, FF, RF, BF, LF go around the cube counterclockwise.
Opposites: TF/DF, FF/BF, RF/LF. The cube is never reoriented as a whole.
Colors: single-letter tokens W, G, R, B, O, Y. W is the solved color of TF, G of FF, R of RF, B of BF, O of LF, Y of DF.
Sticker address: <face> R<row> C<column>. The center of any face is R2 C2.
Row 1 touches TF for the four side faces, BF for TF, FF for DF.
Column 1 touches LF for TF, FF and DF; FF for RF; RF for BF; BF for LF.
State text: six blocks, one block per line, in the order TF, FF, RF, BF, LF, DF. Each block is "<face>: t t t / t t t / t t t", where each t is one color token, row 1 first, column 1 leftmost.
Moves: <face>+1 = one quarter turn of that face clockwise seen from outside it, <face>-1 = counterclockwise.
Twelve moves total. There is no half-turn token.
One move per turn, format: <face>+1 or <face>-1
Turns are numbered from 1. Every turn request shows the turn number.
Lag 1 is the move you just completed; lag 2 is the move immediately before it; lag 3 the one before that.

Each turn, reply with exactly one move token in the form <face>+1 or <face>-1 and nothing else.
A question that supplies lettered options is answered with exactly one capital letter and nothing else.
The letter is the label in front of the option. It is not the text of the option.
A question that asks for the state is answered with the state text and nothing else.
If you are not sure, still give your best answer in the required form. Never send an empty reply. Never ask a question.
Output notation only, no explanation. Replies are pure notation.
This contract is stated here once and is never repeated when a turn is requested.

Figure 5: **System prompts.** Each is shared by the symbolic encoding, the image encoding and the image and symbolic encoding.

Table 9: **Direction, forward minus backward.** Points, paired by problem and encoding over every pair admitted in both directions, with 95 percent intervals from a bootstrap over problems. *Problems* counts the problems with at least one such pair. *Move* to *Next* are the probes of Table 3. *Solved* is the percent of runs that reach the goal or, backward, the initial state. t_f/L^* is the turn of the first failure as a percent of L^*. A positive value favors the forward direction. Table 10 gives the pilot settings.

Model	Task	Problems	Move	Place	Rec.	Conf.	Next	Solved	t_f/L^*
GPT-5.6 Terra	Hanoi	66	+10 [+5, +15]	+34 [+17, +50]	−15 [−21, −9]	+31 [+25, +38]	−5 [−14, +5]	+15 [+8, +22]	+22 [+15, +30]
	Cube	60	−4 [−11, +3]	+14 [+4, +23]	0 [−1, +1]	+12 [+9, +15]	+22 [+15, +28]	+3 [0, +7]	0 [−1, +2]
Flash, thinking low	Hanoi	72	+3 [+1, +5]	+10 [+5, +15]	−3 [−5, −1]	+12 [+8, +15]	+14 [+9, +20]	+17 [+11, +24]	+26 [+18, +34]
	Cube	60	+2 [0, +3]	+11 [+1, +19]	+1 [0, +1]	+8 [+4, +11]	+21 [+17, +26]	+1 [0, +2]	0 [−1, +2]

Table 10: **Table 9 for the pilot settings.** Points, forward minus backward, with 95 percent intervals from a bootstrap over problems. Intervals over fewer than 10 problems are not shown.

Model	Task	Problems	Move	Place	Rec.	Conf.	Next	Solved	t_f/L^*
Flash, thinking high	Hanoi	12	0	0	0 [0, 0]	+1 [0, +3]	−8	−4 [−12, 0]	−20
	Cube	10	0	+16	0 [0, 0]	+12 [−8, +31]	+20	+10 [0, +30]	+1
Sonnet 5, effort low	Hanoi	8	−12	+12	+3	+33	+33	+38	+4
	Cube	9	−5	+32	+12	−1	+27	+11	+15
Sonnet 5, thinking off	Hanoi	7	0	−33	−5	+45	−7	+21	+8
	Cube	9	0	+4	−6	0	+3	+11	+16

E Extended related work

This appendix gives the longer review that Section 2 summarizes.

Keeping an explicit state. A learned world model that keeps a sparse graph of object interactions predicts physical scenes better than dense baselines at long horizons (Orujlu et al., 2026). It suggests one way for agents to keep the state itself.

Spatial mental simulation. MindCube, LEGO-Puzzles and STARE ask models to simulate spatial changes from pictures, each within a single question (Wang et al., 2026; Tang et al., 2026; Li et al., 2026). In our runs the change comes from the model's own moves over many turns, with no picture after the supply.

Evidence for the three problems. Each problem named in Section 1 has direct evidence. New observations arrive during the run in many evaluations. VAGEN returns a new image after every action (Wang et al., 2025a), OST-Bench adds new frames in every round (Lin et al., 2025), and agents in CubeBench can ask to see the cube again (Gao et al., 2026). Questions and answers that stay in the conversation shape later turns. When an instruction is revealed over several turns, models commit early to their own guesses and rarely recover (Laban et al., 2026). Appendix A.3 estimates how much our own checks would reveal if they stayed in the run. A model's account of its own behavior is weak evidence. Explanations in a chain of thought can leave out a biasing feature of the input that changed the answer (Turpin et al., 2023). How much an answer depends on the stated reasoning varies widely between tasks (Lanham et al., 2023). Explanations also predict the model's answers on changed inputs poorly (Chen et al., 2024).

State tracking in language models. One line of work gives a model a start state and then a list of operations and asks about the resulting state. Kim & Schuster (2023) do this with boxes and objects and find that among the models they prompt without fine-tuning only those pretrained on much code track the state beyond a simple baseline. Toshniwal et al. (2022) and Li et al. (2023) train transformers on chess and Othello move lists and find that the models come to track the board. Li et al. (2025) study the mechanisms that language models learn when trained to compose permutations. In all of these the model watches operations that someone else chose within a single input. In our runs the state changes through the model's own moves over many turns. We also ask about it at every turn without changing the run. Pereira & Zuidema (2026) report in a recent preprint that reasoning models encode the Hanoi state well at the end of the prompt and then lose it during the chain of thought. Their evidence comes from probes on the activations of open weight text mod-

Table 11: **Table 3 with intervals.** Each cell gives the mean and, in brackets, the 95 percent interval from a bootstrap over problems. The columns are those of Table 3. Intervals over fewer than 10 problems are not shown. *Avg* leaves out the problems that lack one of the five probes. These are all problems with $L^* \leq 2$, which have no place recall, and a few longer ones. They score higher, so *Avg* in Table 3 is up to 6 points below the mean of its five columns. The chance and copy rows average over the cells of each table the score that a chance answer or a copy of the initial state gets on the items of that cell. Here they also average the pilot cells, so they differ from those of Table 3.

		Tower of Hanoi							Rubik's cube						
Model	Supply	Read	Move	Place	Rec.	Conf.	Next	Avg	Read	Move	Place	Rec.	Conf.	Next	Avg
Chance		–	27	33	36	–	31	–	–	25	25	31	–	25	–
Copy of the initial state		–	–	–	–	41	–	–	–	–	–	–	43	–	–
Proprietary models, full grid															
GPT-5.6 Terra, reasoning off	Symbolic	98 [95,100]	78 [67,88]	53 [38,67]	90 [84,95]	68 [61,75]	33 [24,42]	61 [56,66]	100 [100,100]	85 [76,92]	43 [30,56]	99 [97,100]	29 [23,36]	44 [36,52]	58 [54,62]
46/43 problems	Image	100 [100,100]	86 [75,95]	58 [42,72]	90 [83,95]	64 [57,71]	35 [25,45]	62 [56,69]	62 [53,70]	84 [75,92]	41 [28,53]	99 [97,100]	22 [17,26]	46 [38,55]	57 [52,61]
	Image + symbolic	100 [100,100]	84 [75,92]	68 [55,81]	92 [87,96]	71 [64,79]	44 [33,56]	68 [62,74]	100 [100,100]	82 [73,90]	41 [28,54]	99 [98,100]	31 [25,38]	34 [25,43]	55 [51,60]
	Image + text	100 [100,100]	87 [77,95]	65 [52,77]	97 [94,99]	77 [69,84]	48 [36,59]	69 [64,74]	100 [100,100]	90 [82,97]	24 [14,35]	99 [98,100]	31 [26,37]	26 [17,36]	51 [48,54]
Gemini 3.8 Flash, thinking low	Symbolic	100 [100,100]	99 [98,100]	89 [84,93]	98 [96,99]	93 [90,95]	60 [51,69]	84 [82,87]	100 [100,100]	100 [100,100]	54 [42,65]	100 [100,100]	43 [34,53]	24 [16,31]	61 [58,65]
72/53 problems	Image	100 [100,100]	98 [96,100]	84 [76,91]	98 [96,99]	88 [85,92]	58 [50,66]	81 [78,85]	68 [62,74]	95 [89,99]	46 [34,57]	100 [99,100]	26 [18,35]	19 [12,27]	54 [51,57]
	Image + symbolic	100 [100,100]	98 [96,100]	77 [70,85]	98 [96,99]	87 [83,91]	62 [53,70]	80 [77,84]	100 [100,100]	99 [97,100]	50 [38,62]	100 [100,100]	37 [28,46]	11 [6,16]	56 [53,59]
	Image + text	100 [100,100]	98 [96,100]	88 [80,93]	98 [97,99]	91 [88,94]	61 [53,68]	84 [81,88]	100 [100,100]	100 [100,100]	73 [62,83]	100 [99,100]	53 [44,62]	24 [17,31]	67 [63,71]
Proprietary models, pilot grid (22 problems)															
Gemini 3.8 Flash, thinking high	Symbolic	100 [100,100]	100	100	100 [100,100]	100 [100,100]	91 [83,97]	98	100 [100,100]	100	100	100 [100,100]	84 [69,97]	44	85
12/10 problems	Image	100 [100,100]	100	100	100 [100,100]	99 [97,100]	91 [83,99]	98	72 [56,86]	100	78	100 [100,100]	46 [31,62]	50	73
Claude Sonnet 5, effort low	Symbolic	100	88	88	100	84	79	90	100	100	54	90	31	79	69
8/8 problems	Image	94	62	75	100	70	71	76	49	96	36	98	21	60	61
Claude Sonnet 5, thinking off	Symbolic	100	100	100	83	62	34	71	100	100	20	69	53	44	62
6/6 problems	Image	100	100	50	92	62	0	58	49	100	25	75	29	78	60

Table 12: **Where the state is lost.** Admitted runs, turns up to L^*, means over problems with 95 percent intervals from a bootstrap over problems. *Hanoi configuration* splits the configuration probe by whether every move so far was carried out. *Displaced objects* are the rings or pieces that are no longer where they started in the true current state. A model that simply repeats the initial state gets every other object right and every displaced object wrong. The share of displaced objects placed correctly thus shows how much of the change the model followed. For example, if two of four rings have left their starting poles, the configuration score counts all four rings and this score counts only those two.

	Hanoi configuration		Displaced objects right		Cube displaced pieces right, by turns taken			
Model	No rejected move	After a rejected move	Hanoi	Cube	1	2 to 5	4 to 5	6 to 9
Terra	93 [91, 95]	49 [44, 53]	77 [71, 81]	7 [5, 10]	18 [11, 26]	5 [4, 6]	4 [3, 5]	4 [3, 5]
Flash	99 [98, 99]	68 [64, 72]	90 [86, 93]	34 [26, 42]	92 [87, 96]	22 [19, 25]	15 [13, 17]	11 [10, 13]

els inside one reply. Our evidence is behavioral, covers many turns and both proprietary and open weight vision-language models, and varies the input encoding. We see the two as converging lines of evidence.

Spatial memory and decline over turns in multimodal models. Lin et al. (2025) find that accuracy falls as an exploration of a scene gets longer. Yang et al. (2025) find that spatial reasoning over video stays below human level. Jia et al. (2026) find, on single-image questions, that models trained for spatial questions gain little on a broad spatial benchmark. In the first two benchmarks new frames arrive at every step or the full video is given. They motivate our question and do not answer it, because a model never has to act on a state that it no longer sees.

Encoding of the input. With the input in view, models do better with text than with images (Fu et al., 2024; Tang et al., 2025), and image and text together usually beat the image alone (Fu et al., 2024). Cohen & Mooney (2026) vary whether a list of actions is given as text or as images and also find text ahead. In their tasks with several steps the initial state is always text and the model does not act. Gao et al. (2026) and Anand & Shareghi (2025) supply a Rubik's cube in several encodings, and in both the model can see the cube again during the run. None of these asks whether a gap between encodings grows or shrinks over a run after a supply that is given only once.

Puzzles as tests of reasoning and planning. Planning benchmarks found that the language models of 2023 often failed to produce valid plans (Valmeekam et al., 2023), and some argue that they need an outside checker (Kambhampati et al., 2024). Shojaee et al. (2025) report that reasoning models collapse on the Tower of Hanoi beyond a certain size. A later comment questions the scoring of full move lists (Lawsen, 2025), and Dellibarda Varela et al. (2025) find that failures remain with

Table 13: **Table 3 on every run.** Every complete run of the full grid models, with the runs that failed the gate included. The chance and copy rows are computed on these runs in the same way.

		Tower of Hanoi							Rubik's cube						
Model	Supply	Read	Move	Place	Rec.	Conf.	Next	Avg	Read	Move	Place	Rec.	Conf.	Next	Avg
Chance		–	27	33	34	–	30	–	–	25	25	31	–	25	–
Copy of the initial state		–	–	–	–	49	–	–	–	–	–	–	35	–	–
Proprietary models, full grid															
GPT-5.6 Terra, reasoning off	Symbolic	99	84	48	89	63	35	60	100	84	35	98	27	37	54
72/60 problems	Image	99	84	48	88	60	32	59	54	85	33	99	19	45	55
	Image + symbolic	100	88	52	89	67	36	63	100	82	31	100	29	31	52
	Image + text	100	88	60	92	72	41	65	100	89	27	99	31	25	51
Gemini 3.8 Flash, thinking low	Symbolic	100	99	89	98	93	61	85	100	100	52	100	42	20	60
72/60 problems	Image	100	98	84	98	88	58	81	65	97	48	100	24	17	54
	Image + symbolic	100	98	78	98	87	62	81	100	99	51	100	35	10	55
	Image + text	100	98	88	98	92	61	84	100	100	72	100	50	24	66

Table 14: **One move after Remind, symbolic encoding.** *Remind* and *Placebo* are the configuration probe after the next move of each copy, as the percent of objects in place. *Difference* is their paired difference. On runs whose next move was carried out, *Copy* is the score of simply repeating the reminded state and *Exact* is the percent of runs in which the model wrote the new state without error. Means are over problems with 95 percent intervals from a bootstrap over problems. A cube quarter turn moves 8 of the 20 pieces, so repeating the given state always scores 60 percent.

		All first failures				Next move carried out			
Model	Task	Problems	Remind	Placebo	Difference	Runs	Remind	Copy	Exact
GPT-5.6 Terra	Hanoi	44	74 [67, 81]	57 [49, 64]	+18 [+14, +23]	22	98 [95, 100]	73 [68, 77]	78 [57, 95]
	Cube	57	60 [57, 63]	33 [28, 38]	+27 [+23, +31]	95	60 [57, 63]	60 [60, 60]	7 [2, 13]
Flash, thinking low	Hanoi	34	99 [98, 100]	79 [72, 85]	+20 [+14, +27]	55	100 [99, 100]	77 [75, 79]	99 [96, 100]
	Cube	48	78 [73, 81]	40 [34, 46]	+37 [+31, +44]	92	78 [73, 81]	60 [60, 60]	36 [26, 47]

stepwise prompting. Ren et al. (2024) report lower success when a language model writes a whole plan backward. We ask for one move per turn, so the length of a reply never limits a run. Wu et al. (2024) let a model play Hanoi turn by turn and describe the state to it again at every step. We do not rank models on solving. We use exact states and known optimal lengths to score what a model can say about the current state at every turn.

Measuring without self-report. Studies of faithfulness do not rely on what a model says about itself. They change the model or the input and measure the change in the answer. They remove a reasoning step from the weights (Tutek et al., 2025) or contradict the explanation in the prompt (Chuang et al., 2026). Shi et al. (2026) show that self-prompted looking does not restore the use of an image and that an instruction from the user does. Our reminder goes one step further, because in our runs there is no current view to look back at. Work on probing warns that a probe score mixes what the probe can do on its own with what the model holds (Hewitt & Liang, 2019). We therefore report chance levels and a copy of the initial state, a baseline that needs no state. Studies of human causal judgment separate whether a factor made a difference from how it shaped the outcome (Gerstenberg et al., 2021; Lu et al., 2026). Our Mask intervention hides one earlier move and so asks the first kind of question.

Table 15: **Interventions on the pilot grid.** The same measures as Table 4, for the three settings that ran 22 problems in the symbolic and image encodings. They are small, and we read them for direction only. Intervals over fewer than 10 problems are not shown.

(a) Remind, reminder minus placebo

Model	Symbolic	Image	Image + symbolic	Image + text
Flash, thinking high	−6	+15 [−30, +55]	–	–
Sonnet 5, effort low	+6 [−6, +22]	0 [0, 0]	–	–
Sonnet 5, thinking off	0 [0, 0]	+7 [0, +18]	–	–

(b) Mask

			Still optimal				Difference	
Model	Task	Pairs	Replay	Marker	Mask	Placebo	Mask − marker	Mask − placebo
Flash, thinking high	Hanoi	64	93	100	85	92	−15	−6
	Cube	37	32	26	20	–	−5	–
Sonnet 5, effort low	Hanoi	13	94	90	65	78	−25	−13
	Cube	9	78	89	78	–	−11	–
Sonnet 5, thinking off	Hanoi	1	50	0	0	0	0	0
	Cube	20	80	98	82	–	−16	–

[First User Message, Symbolic Encoding]

Initial: L=L_C=GS_R=E
Goal: L=GS_C=L_R=E

Your task. Change the Initial state into the Goal state.
You make one move per turn. Use as few moves as you can.
Rule: none (free variant). Any top ring may be placed on any pole. Only a top ring may move.
A move that cannot be made is not carried out.
This covers a move whose ring is not the top ring of its from pole and a reply that is not a move token.
The rings then stay where they were. The turn still counts. You are not told when this happens.
After every move the environment replies with the constant token "ok."
It carries no information: no legality feedback, no success feedback, no state, ever.
Budget: at most 10 turns.
The run ends by itself when the Goal state is reached or when the budget is used up.
As long as you are asked for a move, reply with one move.

env> Turn 1 of at most 10. Your move?
model> S:L->R
env> ok.
The exchange above is an example. Your run has not started; your first move will be turn 1.
The move above illustrates the reply format. It is not a hint about this problem.
The move in the example may or may not be possible in this problem.

Turn 1 of at most 10. Your move?

Figure 6: **The first message of a run, symbolic encoding, Tower of Hanoi.** It holds the supply, the task, the rules and one example turn.

Table 16: **Open weight models, every encoding, all runs.** As in Table 3, with runs that failed the gate included. The copy line differs by model, and Table 7 gives it for the symbolic encoding.

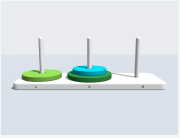
		Tower of Hanoi							Rubik's cube						
Model	Supply	Read	Move	Place	Rec.	Conf.	Next	Avg	Read	Move	Place	Rec.	Conf.	Next	Avg
Chance		–	27	33	35	–	34	–	–	25	25	31	–	25	–
Open weight models															
Qwen3.8 27B, thinking off	Symbolic	100	86	48	79	55	37	56	100	87	49	99	51	45	63
72/60 problems	Image	94	85	54	85	52	34	56	45	85	38	94	18	39	52
	Image + symbolic	100	88	58	84	61	37	59	100	86	42	91	35	32	54
	Image + text	100	81	73	92	63	56	67	100	84	41	100	31	52	60
Qwen3.8 27B, thinking on	Symbolic	100	90	99	99	98	77	91	100	100	100	100	96	5	80
36/30 problems	Image	97	99	97	100	97	72	93	50	97	68	91	33	25	59
Gemma 4 31B, thinking off	Symbolic	100	91	51	90	54	46	61	100	85	38	100	27	22	52
72/60 problems	Image	95	90	45	87	56	40	58	36	95	36	100	17	27	53
	Image + symbolic	100	91	54	89	62	44	63	100	90	41	100	25	12	52
	Image + text	100	89	68	91	72	43	68	100	89	43	100	33	9	52
Gemma 4 31B, thinking on	Symbolic	100	100	95	100	96	75	91	92	98	81	100	64	44	73
32/30 problems	Image	93	100	91	97	89	58	83	37	100	53	100	30	40	63
Qwen3.5 9B, thinking off	Symbolic	87	70	41	57	45	43	48	97	56	35	95	26	43	47
72/60 problems	Image	60	76	42	71	43	34	48	7	71	30	85	22	40	47
	Image + symbolic	92	75	36	69	51	35	50	99	69	31	87	34	40	49
	Image + text	86	67	38	86	54	42	51	97	67	24	100	30	57	53
Qwen3.5 9B, thinking on	Symbolic	25	85	88	91	56	54	74	75	89	43	88	30	36	55
36/30 problems	Image	18	91	78	84	49	32	64	9	100	28	88	4	6	44
InternVL3.5 14B	Symbolic	84	71	48	56	39	37	44	100	79	44	92	36	45	57
72/60 problems	Image	65	80	48	62	37	43	48	34	94	34	90	23	49	56
	Image + symbolic	79	78	58	63	45	37	48	99	91	38	88	48	47	61
	Image + text	97	51	68	81	55	48	54	100	75	37	99	26	48	54
GLM-4.6V Flash, thinking off	Symbolic	89	60	42	38	46	40	42	79	63	35	70	32	22	43
72/60 problems	Image	18	47	42	49	39	35	39	30	61	30	70	14	23	38
	Image + symbolic	58	46	45	46	45	39	41	64	59	35	68	30	23	41
	Image + text	71	45	48	72	54	48	47	73	62	41	87	27	18	45
GLM-4.6V Flash, thinking on	Symbolic	35	81	44	69	33	38	49	53	88	48	92	16	40	56
36/30 problems	Image	33	76	40	66	29	42	44	33	84	32	91	15	41	53
	Image + symbolic	29	80	38	68	36	46	45	65	79	56	95	22	43	58
	Image + text	8	76	60	88	61	50	60	64	100	49	99	37	47	65
Kimi-VL A3B Instruct	Symbolic	23	83	38	57	38	38	54	42	45	26	54	28	37	34
72/60 problems	Image	23	66	34	47	18	43	41	4	62	30	51	21	46	39
	Image + symbolic	37	73	39	53	32	37	47	41	52	46	49	32	39	42
	Image + text	29	49	54	71	32	28	45	27	95	41	80	35	18	54
Kimi-VL A3B Thinking	Symbolic	6	38	6	38	5	36	25	12	75	28	62	9	17	44
26/27 problems	Image	2	40	45	46	3	26	33	2	69	21	72	4	24	39
	Image + symbolic	8	33	52	58	4	42	30	28	75	28	70	23	18	42
	Image + text	8	57	60	64	41	44	48	11	93	42	77	18	30	55
Specialized spatial models and their base															
SpaceThinker Qwen2.5-VL 3B	Symbolic	35	38	20	25	43	35	28	81	17	7	41	38	30	27
36/30 problems	Image	6	22	28	23	10	31	23	5	21	3	38	9	22	19
	Image + symbolic	39	43	25	15	40	35	31	54	35	0	32	42	24	23
	Image + text	8	39	16	48	39	35	30	11	31	2	56	13	29	25
SpaceQwen2.5-VL 3B	Symbolic	65	65	28	19	53	36	39	61	51	30	36	29	28	33
72/60 problems	Image	8	74	36	25	10	33	33	2	69	31	43	3	35	36
	Image + symbolic	64	64	45	21	52	37	41	47	64	28	40	32	47	41
	Image + text	10	47	38	45	51	34	39	0	62	30	43	34	27	38
Qwen2.5-VL 3B (base)	Symbolic	56	73	36	23	53	37	42	81	76	29	55	39	43	48
72/60 problems	Image	17	67	36	31	20	40	36	5	67	32	62	5	34	40
	Image + symbolic	50	73	44	30	48	40	47	68	72	31	63	35	49	48
	Image + text	21	32	49	57	51	37	41	33	55	38	78	24	29	42

Table 17: **Gate and readable replies of the open weight models.** *Gate* is the percent of runs that passed the gate on the first try. *Unread* is the percent of moves in the main run that the parser could not read. *Lost* counts runs that did not finish. For Gemma 4 31B with thinking and SpaceThinker these hit the context limit of the server, and for Kimi-VL with thinking they stopped at the gate under an earlier rule. The thinking settings marked † ran 66 problems.

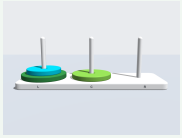
	Tower of Hanoi			Rubik's cube		
Model	Gate	Unread	Lost	Gate	Unread	Lost
Open weight models						
Qwen3.8 27B, thinking off	84	4	0	39	12	0
Qwen3.8 27B, thinking on†	91	21	0	62	71	0
Gemma 4 31B, thinking off	81	10	0	69	0	0
Gemma 4 31B, thinking on†	80	8	18	88	8	1
Qwen3.5 9B, thinking off	49	5	0	35	0	0
Qwen3.5 9B, thinking on†	59	35	0	69	50	0
InternVL3.5 14B	56	12	0	18	0	0
GLM-4.6V Flash, thinking off	44	13	0	4	0	0
GLM-4.6V Flash, thinking on†	77	9	0	41	3	0
Kimi-VL A3B Instruct	24	57	0	3	35	0
Kimi-VL A3B Thinking†	16	62	38	5	69	13
Specialized spatial models and their base						
SpaceThinker Qwen2.5-VL 3B†	5	33	1	1	31	4
SpaceQwen2.5-VL 3B	8	31	0	0	0	0
Qwen2.5-VL 3B (base)	11	24	0	5	0	0

[Image Encoding, Tower of Hanoi]

In these pictures ring G is green, ring L is lime, ring S is sky blue.



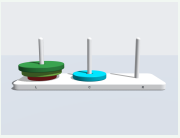
Initial state



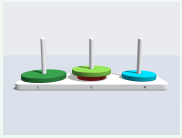
Goal state

[Image and Symbolic Encoding, Tower of Hanoi]

Initial: L=RLG_C=S_R=E
Goal: L=G_C=RL_R=S
The pictures and the text show the same two states.
In these pictures ring G is green, ring L is lime, ring S is sky blue, ring R is red.



Initial state



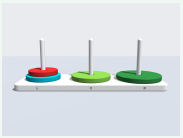
Goal state

[Image and Text Encoding, Tower of Hanoi]

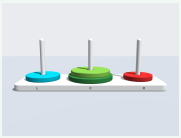
Here are the start state and the goal state. Each state is given in words and in one photo. The words and the photo show the same state.
In the photos, each ring has the color it is named by.
Each photo shows a letter under each pole. You do not need these letters.

Start state.
The left pole holds, from bottom to top, the sky blue and red rings.
The middle pole holds only the lime ring.
The right pole holds only the green ring.

Goal state.
The left pole holds only the sky blue ring.
The middle pole holds, from bottom to top, the green and lime rings.
The right pole holds only the red ring.



The next photo shows the start state



The next photo shows the goal state

[Image Encoding, Rubik's Cube]

Each state is shown in two pictures.
View a is taken from the front right, looking down. It shows face TF on top, face FF on the left and face RF on the right.
View b is taken from the back left, looking up from below. It shows face BF on the left, face LF on the right and face DF at the bottom.
In these pictures W is white, G is green, R is red, B is blue, O is orange, Y is yellow.



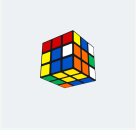
Initial state, view a



Initial state, view b



Goal state, view a



Goal state, view b

Figure 7: **How the other encodings supply the states.** Each panel is the opening of the first message. The task, the rules and the example turn follow as in Figure 6, and as plain sentences in the image and text encoding (Figure 8).

[System Prompt, Image and Text Encoding]

This puzzle has three poles and some rings.
The poles are the left pole, the middle pole and the right pole. They are always named in that order.
Each ring is named by its color.
This problem uses 4 rings. From biggest to smallest they are green, lime, sky blue and red.
Each state says which rings are on each pole. Rings on a pole are always listed from the bottom ring to the top ring.
You make one move per turn.
The first turn is turn 1. Every turn request shows the turn number.
Lag 1 is the move you just completed. Lag 2 is the move right before it. Lag 3 is the move before that.

Each turn, reply with exactly one move sentence and nothing else.
Every move sentence has this form. Move the COLOR ring from the OLD pole to the NEW pole.
Write the color of the ring in place of COLOR. Write left, middle or right in place of OLD and in place of NEW.
OLD is the pole the ring leaves. NEW is the pole the ring goes to.
When a question gives lettered options, reply with exactly one capital letter and nothing else.
The letter is the label in front of the option. It is not the words of the option.
When a question asks for the state, reply with the state sentences it asks for and nothing else.
If you are not sure, still give your best answer in the form that was asked for. Never send an empty reply. Never ask a question.
Do not explain. Do not add any other words.
These reply rules are stated once, here. They are not repeated when a turn is requested.

[Task and Rules, Image and Text Encoding] *the rest of the first message*

Your task. Change the start state into the goal state.
You make one move per turn. Use as few moves as you can.
Rules. Only the top ring of a pole may move. Never put a bigger ring on top of a smaller ring.
If a move cannot be made, it is not carried out.
This covers a move that breaks the rules. It covers a move of a ring that is not on top of the pole it leaves. It covers a reply that is not a move sentence.
The rings then stay where they were. The turn still counts. You are not told when this happens.
After every move the environment says "ok." and nothing more.
That reply tells you nothing. It does not say whether your move was allowed. It does not say whether your move helped. It never shows the state.
You have at most 30 turns.
The run ends by itself when the goal state is reached or when your turns are used up.
As long as you are asked for a move, reply with one move sentence.

Here is one example turn.
The environment says, "Turn 1 of at most 30. Your move?"
You reply, "Move the sky blue ring from the left pole to the right pole."
The environment says, "ok."
The exchange above is an example. Your run has not started. Your first move will be turn 1.
The move above shows the reply form. It is not a hint about this problem.
The move in the example may or may not be possible in this problem.

Turn 1 of at most 30. Your move?

Figure 8: **The image and text encoding uses no notation.** Its system prompt, and the task and rules of its first message, are plain sentences.

[Comprehension Gate] *one of five checks, before turn 1*

Before turn 1 there are five short checks about what you were given above. Answer each with one capital letter.
In the Initial state, name the top ring of pole R. Answer with the letter of the correct option.
A) G
B) L
C) S
D) E

[Recall, Action Form]

Lag 1 is the move you just completed; lag 2 is the move immediately before it; lag 3 the one before that. Consider the move you made at lag 1. Which move was it? Answer with the letter of the correct option.
A) L:L->R
B) S:C->L
C) S:C->R
D) L:R->L

[Recall, Spatial Form]

Lag 1 is the move you just completed; lag 2 is the move immediately before it; lag 3 the one before that. Immediately after your move at lag 1, which pole held ring L? Answer with the letter of the correct option.
A) C
B) R
C) L

[Recency]

Before now, when did ring S last move? Your most recent completed move counts. Answer with the letter of the correct option.
A) NEVER
B) turn 1
C) turn 3
D) turn 2

[Configuration] *also asked at turn 0 and at the first failure*

State the current configuration in the state grammar (L=..._C=..._R=...).

[Reasoning, Legality or Distance, then Next Move] *two questions on one copy*

Environment
Did your most recent move reduce the distance to the goal? Answer with the letter of the correct option.
A) yes
B) no

Model
B

Environment
Which single legal move brings you closer to the goal? Answer with the letter of the correct option.
A) L:L->C
B) S:C->R

Figure 9: **The questions, symbolic encoding, Tower of Hanoi.** Each is asked on a fork that is thrown away after the answer is scored.

[Recall, Spatial Form, Image and Text Encoding]

Lag 1 is the move you just completed. Lag 2 is the move right before it. Lag 3 is the move before that. Right after your move at lag 1, which pole held the sky blue ring? Answer with the letter of the correct option.
Option A. The right pole.
Option B. The middle pole.
Option C. The left pole.

[Reasoning, Image and Text Encoding]

Environment
Did the rules allow your most recent move? Answer with the letter of the correct option.
Option A. Yes.
Option B. No.

Model
A

Environment
Which one move is allowed by the rules and brings you closer to the goal? Answer with the letter of the correct option.
Option A. Move the red ring from the middle pole to the left pole.
Option B. Move the red ring from the middle pole to the right pole.
Option C. Move the sky blue ring from the left pole to the right pole.
Option D. Move the sky blue ring from the left pole to the middle pole.

[Configuration, Rubik's Cube]

State the current configuration. Write exactly six lines, one per face, in the order TF, FF, RF, BF, LF, DF. Each line has the form "<face>: t t t / t t t / t t t", where each t is one color token, row 1 first and column 1 leftmost. Reply with the six lines and nothing else.

Figure 10: **The same questions in the image and text encoding, and the configuration question on the cube.**

[Why, Not Scored] *asked at the first failure*

In one or two short sentences, say why you chose your most recent move.

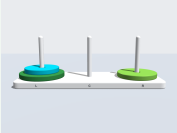
[Remind, True Reminder]

Model
G:L->C

Environment
ok.
Reminder. This is the current state.
L=L_C=GS_R=E
Turn 2 of at most 10. Your move?

[Remind, True Reminder, Image Encoding]

Environment
ok.
Reminder. This is the current state.



Current state

Turn 6 of at most 10. Your move?

[Remind, Placebo]

Model
G:L->C

Environment
ok.
Reminder. Please think about the current state before you move.
Turn 2 of at most 10. Your move?

[Mask] *the model reply at turn j is hidden*

Environment
ok.
Turn 2 of at most 12. Your move?

Model
[move hidden]

Environment
ok.
Turn 3 of at most 12. Your move?

[Mask, Marker Only] *the hidden reply is given back*

Environment
ok.
Turn 2 of at most 12. Your move?

Model
[move hidden]

Environment
ok.
Here is the move that is hidden above.
G:L->R
Turn 3 of at most 12. Your move?

Figure 11: **The question at the first failure and the two interventions.** Remind continues copies of the run. Mask asks again for one decision.

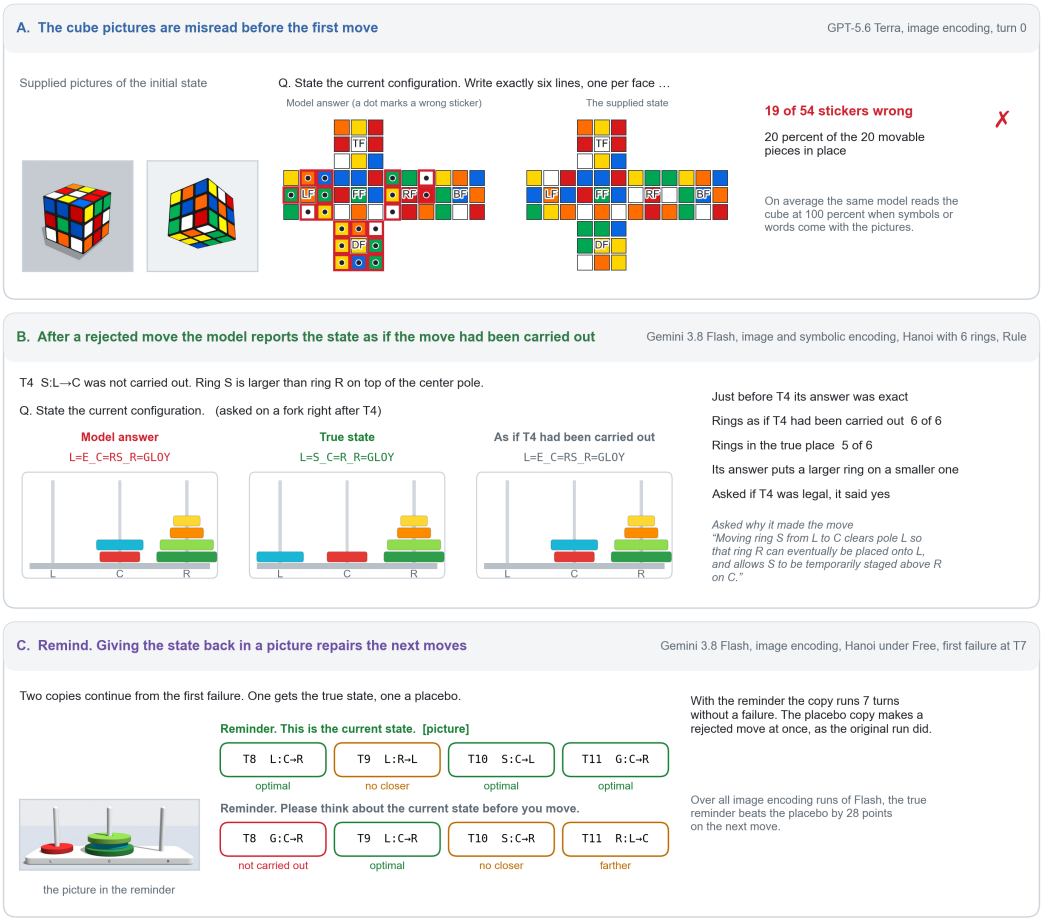


Figure 12: **Three cases from the logs.** Every answer is the model reply as logged. The pictures in A and C are those the model received, and we draw every other state from the logged text.