FINDING THE MOVE IS NOT WINNING THE GAME: XIANGQIBENCH FOR CLOSED-LOOP EVALUATION OF LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Static evaluations credit a language model for naming the right move, but an agent must carry a plan through to a verified outcome while an opponent responds. We introduce **XiangqiBench**, an executable benchmark that measures this difference in Chinese chess: starting from 119 tactical endgames with forced mates supported by engine or checks-only search, an LLM agent must deliver checkmate against an engine defender. An interactive REPL interface separates real moves, state queries, and forward simulation, and we record 8,568 multi-turn trajectories from 12 frontier LLMs under two observation protocols. Three signals that look like competence each overstate closed-loop success. (i) *The Conversion Gap*: models play the stored reference first move in 26.1% of Sighted trials, yet only 13.9% of these trials end in a win. (ii) *The Consistency Gap*: the leading model reaches 38.7% pass@3 but only 5.9% pass^3, winning all three trials on 7 of the 46 positions it ever wins. (iii) *The Simulation Gap*: 32.3% of accepted simulation calls stop on an illegal move, and in 49.3% of comparable cases the real defender replies differently from the line the agent simulated; self-authored rollouts check legality but cannot anticipate the opponent. Finding the move is not winning the game: agent evaluations should score closed-loop outcomes and report reliability alongside coverage.

1 INTRODUCTION

Static evaluations such as puzzle solving and move prediction credit a language model for naming a good action (Chen et al., 2025). Agents are held to a different standard: they must carry a plan through to a verified outcome while an active adversary responds (Kolasani et al., 2025; Liu et al., 2026). Success then hinges on *conversion*: navigating an unfolding contingency tree to a verified goal. Whether static competence predicts conversion is rarely measured directly, because existing benchmarks often confound heuristic proposal, mental planning, and closed-loop execution.

Board-game endgames provide a controlled testbed for this decision loop: state transitions are exact, legality is executable, and outcomes are determined by verifiable terminal states (Toshniwal et al., 2022; Ruoss et al., 2024; Schultz et al., 2025). We study *endgame conversion* in Chinese chess (Xiangqi): starting from a tactical position in which engine or checks-only search reports a forced mate for Red, an LLM agent plays Red against an engine defender. Agreement with reference moves receives no score; the agent must sustain a winning continuation across multiple turns until checkmate (Grace et al., 2026). XiangqiBench exposes this process through an interactive Read–Evaluate–Print Loop (REPL) interface (Fig. 1) that separates real moves, state queries of the current board and legal moves, and a forward-simulation sandbox on a cloned board. Simulation is not an external search engine: the agent must supply both its own candidate moves and the hypothesized opponent replies. The tool therefore shows what the agent expects: does it anticipate the defender's replies, or does it check lines the real defender does not play?

We evaluate 12 frontier LLMs (Gemini 3.1 Pro (Google DeepMind, 2026), GPT-5.5 (OpenAI, 2026), Qwen 3.7 Max (Qwen Team, 2026), Claude Opus 4.7 (Anthropic, 2026), DeepSeek V4 Pro/R1/V3.2 (Guo et al., 2025; Liu et al., 2025; Xu et al., 2026), Seed 2.0 Pro (Seed, 2026), Kimi K2.5 (Team et al., 2026), Mistral Large 3 (Mistral, 2025), MiniMax M2.5 (Chen et al., 2026), and GPT-OSS

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

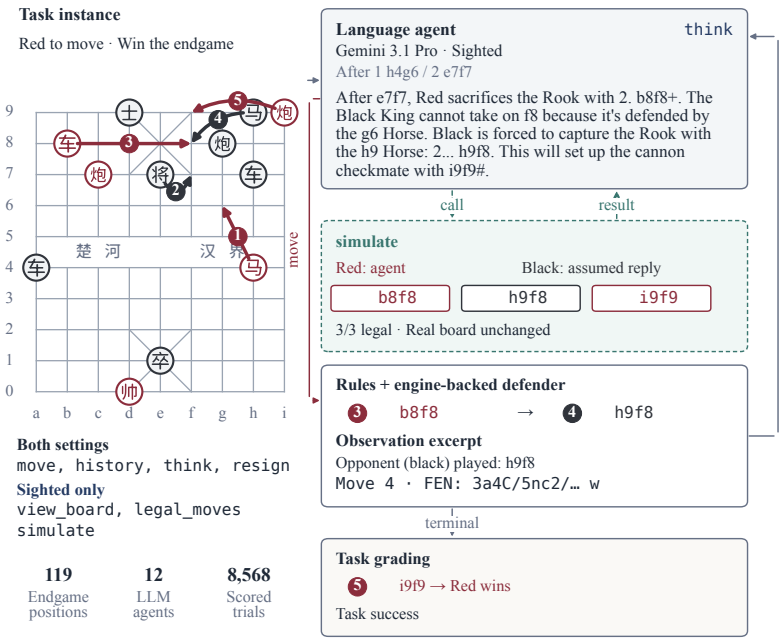


Figure 1: **XiangqiBench at a glance.** *Left:* a task instance with the real trajectory overlaid on the initial board (numbered plies; deep red for the agent, charcoal for the defender; curves separate annotations and are not piece paths), tool availability by setting, and the scale of the evaluation. *Right:* the second Red decision of Gemini 3.1 Pro in Sighted trial 3 on case SQYQ-251, after real plies 1–2 (Appendix I). The complete pre-simulation `think` response is shown; `simulate` then checks three model-specified plies, including an assumed Black reply, on a copy of the board. The defender's actual reply informs the next decision, and the terminal Red win determines task success.

120B (Agarwal et al., 2025); Appendix J) under two observation settings: *Sighted*, which provides board refreshes, legal-move feedback, and the simulation sandbox; and *Restricted observation*, which withholds board refreshes, state queries, and simulation to probe state tracking from move history. Our contributions are:

- **A benchmark for adversarial conversion:** 119 search-supported tactical endgames, an executable rules environment with an engine defender, and 8,568 recorded trajectories.
- **An instrumented decision loop:** we trace each trial from the first move through simulation legality and real-move adoption to defender divergence and the terminal outcome.
- **Three gaps between apparent and closed-loop competence:** a *Conversion Gap* (the reference first move rarely leads to a win), a *Consistency Gap* (pass@3 far exceeds pass^3), and a *Simulation Gap* (self-authored rollouts often break legality and mispredict the defender); a controlled ablation on the two strongest models suggests that much of the Restricted-observation drop reflects the setting's prompt and feedback rather than the missing board alone.

2 Related Work

Interactive agent evaluation. AgentBench, SWE-bench, WebArena, and OSWorld evaluate sequential tool invocation and computer control (Liu et al., 2024; Jimenez et al., 2024; Zhou et al., 2024; Xie et al., 2024). In game domains, GTBench, GameBench, GAMEBoT, KORGym, and TextArena provide rule-grounded strategic environments (Duan et al., 2024; Costarelli et al., 2024; Lin et al., 2025; Shi et al., 2025; Guertler et al., 2025). XiangqiBench complements them by isolating *adversarial conversion* of search-supported winning positions against an engine defender, scoring only the terminal outcome while recording intermediate planning and real-move commitment (Grace et al., 2026).

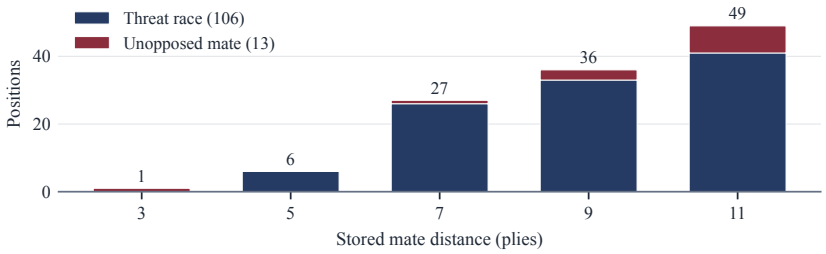


Figure 2: **Stored mate distances of the 119 positions, by category.** Distances are those reported at admission by Pikafish or the checks-only solver, stacked by category: 106 threat races (Black would also have a forced mate if it were to move) and 13 unopposed mates (no such counter-mate found). Distances are admission metadata, not limits on trajectory length during evaluation.

Game planning, state tracking, and simulation. Board games serve as testbeds for neural planning and representation (Toshniwal et al., 2022; Feng et al., 2023; Ruoss et al., 2024; Schultz et al., 2025). In chess, ChessArena and LLM Chess evaluate full games against other LLMs, random players, and engines, and ChessArena adds a blindfold mode that supplies only the move history (Liu et al., 2026; Kolasani et al., 2025). LMRL-Gym uses theoretical endgames as a multi-turn RL testbed (Abdulhai et al., 2025), while Xiangqi-R1 trains a 7B model in multiple stages to improve move legality and position analysis (Chen et al., 2025). We instead study generalist frontier LLMs on multi-turn endgame conversion and separate the decision loop into scratchpad simulation, real-move adoption, and resilience against engine counterplay.

Reliability metrics. Following Chen et al. (2021) and Yao et al. (2025), we use unbiased finite-sample estimators for pass@k (at least one success) and pass^k (all k attempts succeed); contrasting them reveals whether an agent that can reach a win also reaches it reliably.

3 XiangqiBench: Task Construction

Corpus. XiangqiBench curates 119 tactical compositions from classical Chinese-chess treatises in which Red can force checkmate: 116 from *Shi Qing Ya Qu*[1] (case identifiers SQYQ-n) and 3 from the classical *Jianghu* collection (JH-n). Each entry specifies an initial position in Forsyth–Edwards Notation (FEN) with Red to move; historical solution text is withheld from prompts and scoring. During evaluation, an executable rules adapter verifies every proposed action. Xiangqi legality involves non-local constraints: pieces must respect palace boundaries and river crossings, cannons require an intervening screen to capture, horses are blocked by adjacent pieces ("hobbling the horse's leg"), and generals cannot face each other across an open file.

Search-supported admission. Historical treatises can contain transcription errors or overlooked defenses, so source solutions are not trusted. Instead, each position is screened with **Pikafish** (Official Pikafish Contributors, 2026), a strong open-source Xiangqi engine derived from Stockfish that combines alpha–beta search with an NNUE evaluation network. For **admission**, a single-threaded, time-limited search is run from each position, and the position is admitted when the engine reports a forced mate for Red. The reported mate distance and the engine's preferred first move are stored as case metadata; the latter serves as the reference root move in Section 5. Composed positions that the engine does not confirm may instead be admitted by a checks-only mate search and are labelled separately (Appendix A). For **defense**, the same engine family plays Black during evaluation (Section 4). Because admission relies on bounded search rather than a complete proof tree, we describe the positions as *search-supported* rather than formally proved; all 119 positions were additionally checked manually to confirm that Red can force checkmate.

Why endgames. Like the chess-endgame task in LMRL-Gym (Abdulhai et al., 2025), we start from endgames rather than full games: the goal (checkmate) is fixed and reachable, so a failure

[1] https://www.xqipu.com/canjugupu/1547

Table 1: **REPL action space and observability contract.** Only $\mathcal{A}_{\text{env}}$ mutates the real board.

Class	Command & Operational Semantics	Sighted	Restricted
$\mathcal{A}_{\text{env}}$ (Transitions)	`move(from_to)`: Validate and commit a real move; trigger defender	✓	✓
	`resign()`: Surrender; terminate trial with defender win	✓	✓
$\mathcal{A}_{\text{query}}$ (Perception)	`view_board()`: Render ASCII board, FEN, active color, and piece census	✓	✗
	`get_legal_moves()`: Enumerate legal moves in the current state	✓	✗
	`get_history()`: Return the move sequence with piece annotations	✓	✓
$\mathcal{A}_{\text{sim}}$ (Hypothesis)	`simulate(moves)`: Execute alternating line on cloned state; verify legality	✓	✗
	`think(text)`: Record uncommitted deliberation	✓	✓

to mate is attributable to the agent; opening recall and attrition are removed; and a few precisely calculated moves decide the game. Unlike LMRL-Gym's few-piece theoretical endgames against a 1200-Elo engine, ours are composed tactical endgames (median 14 pieces) built on consecutive checks, mating threats, and pins, and a strong engine defends them, so wins cannot rely on blunders.

Position categories and horizon. Red can force mate in every position; the two categories differ only in Black's counterplay. At admission, the same engine or solver is also run with Black to move. If Black then has a forced mate of its own, the position is a *threat race*: Red must keep forcing, because a slow move can let Black execute its mate first. Otherwise it is an *unopposed mate*: no counter-mate was found, so a slow move forfeits tempo but does not by itself hand Black a win. Stored mate distances range from 3 to 11 plies (median 9; Figure 2). The 40-ply cap is more than three times the longest stored distance, leaving room for detours and recovery, and success already falls steeply within this range (Section 5); longer compositions would push nearly every model to zero, whereas this range still separates the strongest models.

4 Interaction and Evaluation Protocol

REPL action space. The agent plays Red against an engine defender for at most 40 plies. At each turn it receives an observation o_t (with interaction history h_t) and emits free text alongside a command from three classes (Table 1): transitions $\mathcal{A}_{\text{env}}$, which commit a move in ICCS coordinates (e.g., `move h2h9`) and trigger the defender's reply, or resign; read-only queries $\mathcal{A}_{\text{query}}$, which inspect the board, legal moves, or history without advancing the game; and hypotheses $\mathcal{A}_{\text{sim}}$, where `simulate` plays an alternating line of up to 8 model-supplied plies on a cloned board and reports its legality, and `think` records deliberation.

Observation settings. Under **Sighted** observation, the agent receives the updated board, FEN, and legal moves after every environment step and may use all queries and the simulator; this tests the agent as a system with external perception. Under **Restricted observation**, the board and FEN are shown once at the start; afterwards the agent receives only move coordinates and captures, `view_board`, `get_legal_moves`, and `simulate` are disabled, and it must track the position from the move history. The two settings also use different system prompts (Appendix A.3).

Engine defense and termination. After each accepted agent move, Black replies with a fixed-depth (depth-18), single-threaded Pikafish (Official Pikafish Contributors, 2026) search; environment fallbacks are described in Appendix A. A trial ends in an agent **win** if Red achieves checkmate or leaves Black without legal moves (stalemate is a loss for the stalemated player). It ends in **failure** upon Black checkmate, agent resignation, exhausting the 40-ply budget, exceeding 3 consecutive illegal moves, or threefold repetition. Under official rules, perpetual checking loses; our replay adjudicates perpetual checks with a simplified long-check rule (Appendix A.5).

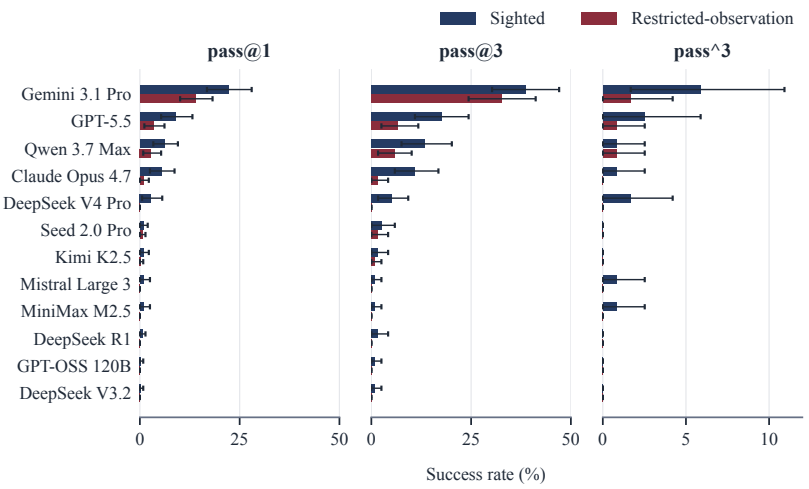


Figure 3: **Multi-turn endgame conversion across frontier LLMs.** Bars report mean pass@1 (single-trial success), pass@3 (at least one success in 3 attempts), and pass^3 (all 3 attempts successful) across 119 positions (357 trials per model/setting). Whiskers denote 95% case-clustered bootstrap intervals.

Metrics: exploration vs. consistency. Let c_i denote the successes on case i out of $n_i = 3$ trials, over $|\mathcal{I}| = 119$ positions. We report (Chen et al., 2021; Yao et al., 2025)

$$\widehat{\text{pass@}k} = \frac{1}{|\mathcal{I}|}\sum_{i=1}^{|\mathcal{I}|}\left[1 - \frac{\binom{n_i - c_i}{k}}{\binom{n_i}{k}}\right], \qquad \widehat{\text{pass\^{}k}} = \frac{1}{|\mathcal{I}|}\sum_{i=1}^{|\mathcal{I}|}\frac{\binom{c_i}{k}}{\binom{n_i}{k}}. \tag{1}$$

pass@k estimates the probability of at least one success in k attempts (exploratory reach), while pass^k estimates the probability that all k attempts succeed (consistency).

5 Evaluation Results

5.1 The Consistency Gap: Occasional Success vs. Reliable Conversion

Single-trial conversion (pass@1) is low for every model and observation setting (Figure 3). Under Sighted observation, Gemini 3.1 Pro leads at 22.1% pass@1 (79/357 trials), followed by GPT-5.5 at 9.0% (32/357) and Qwen 3.7 Max at 6.2% (22/357). Under Restricted observation every model declines: Gemini to 14.0%, GPT-5.5 to 3.4%, and Qwen to 2.8%, with nine models at most 0.8%.

Repeated attempts separate reaching a win from reproducing it. For Gemini 3.1 Pro under Sighted observation, pass@3 is 38.7% (46/119 positions) but pass^3 is 5.9% (7/119): of the 46 positions it wins at least once, it wins all three trials on only 7. For GPT-5.5 the corresponding counts are 21 and 3 positions (17.6% and 2.5%). pass^3 is zero in both settings for five models, and zero under Restricted observation for nine. Wins are also not spread evenly over positions. If Gemini's 22.1% per-trial rate held on every position independently, pass@3 would be 52.7% and pass^3 1.1%; the observed 38.7% and 5.9% indicate that wins concentrate on a subset of positions. Under Sighted observation, Mistral Large 3 and MiniMax M2.5 each win one position in all three trials and no other. Within the subset, outcomes vary from attempt to attempt: on the 46 positions Gemini wins at least once, it wins 57% of trials (79/138), and its split of 20, 19, and 7 positions with one, two, and three wins is close to the 16, 21, and 9 expected from independent trials at that rate. Competence is thus position-specific, and even where it exists a single run does not reveal it reliably.

5.2 The Conversion Gap: Finding the Move vs. Winning the Game

Success falls steeply with the stored mate distance (Figure 4a): Sighted pass@1, averaged over the 12 LLMs, is 11.1% at 5 plies, 6.1% at 7, 2.9% at 9, and 1.8% at 11. The 85 positions at 9 and 11 plies make up 71% of the benchmark.

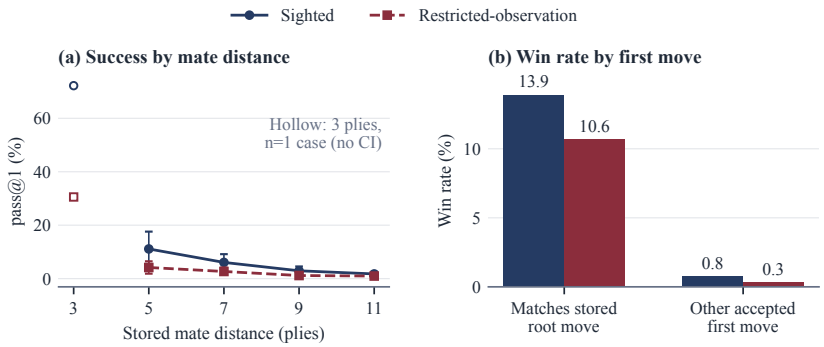


Figure 4: **Deconstructing the Conversion Gap.** (a) Mean pass@1 across the 12 LLMs by stored mate distance, per observation setting (95% case-bootstrap intervals; the 3-ply bin contains a single position). (b) Win rate conditioned on whether the agent's first move matches the stored reference move.

To locate where conversion breaks down, we compare the agent's first real move with the engine's stored reference move (Figure 4b). Across all Sighted trials, the first move matches the reference in 26.1% of cases, far above the 3.3% expected from choosing uniformly among legal root moves; Gemini 3.1 Pro agrees in 45.1% of trials and GPT-5.5 in 42.9%. A correct first move, however, rarely leads to a win. Trials that begin with the reference move are won only 13.9% of the time (155/1,117), so more than 86% of them are lost later in the game. Trials that begin with any other accepted move are won in 0.8% (Sighted) and 0.3% (Restricted observation); a different first move is not necessarily losing, since some positions may admit more than one winning first move. The reference move is thus close to necessary in practice (155 of the 178 Sighted wins begin with it) but far from sufficient. The same gap holds within models: Gemini agrees with the reference move in 45.1% of trials but converts 22.1%; GPT-5.5 agrees in 42.9% but converts 9.0%. We call this the **Conversion Gap**: an evaluation that scored only the first move would credit these models with roughly two to five times the success they achieve over the full game.

5.3 The Effect of Restricted Observation

Restricted observation lowers pass@1 for every model. Mean pass@1 drops from 4.15% in Sighted to 1.82% in Restricted observation, a reduction of 2.33 percentage points (95% case-bootstrap interval [1.42, 3.41]), or more than half. First-move agreement falls to 15.1%, and the mean rate of terminal invalid-action or no-command losses rises from 2.7% to 27.6% of trials (Section 6). Restricted observation, however, also changes the prompt. An ablation that fixes the prompt and removes the state refreshes and the commands separately (Appendix D) shrinks the drop: on positions they solved under Sighted, GPT-5.5 and Gemini without either win 38.1% and 41.7% of trials, against 39.7% and 48.3% with both and 9.5% and 20.0% under Restricted observation. The drop thus measures the full setting, not the missing board alone.

6 Behavioral Analysis

6.1 How Models Use the REPL Interface

Figure 5 shows how each model distributes its actions across the REPL commands; three regularities emerge. **Simulation volume does not track success.** Under Sighted observation, `simulate` accounts for 45% of Gemini 3.1 Pro's actions, but also for 44% of DeepSeek V3.2's and 43% of DeepSeek V4 Pro's, whose pass@1 is 0.3% and 2.8%. Seven of the 12 models request simulation in at least 290 of their 357 Sighted trajectories (Table 4); only DeepSeek R1 (1.1% of actions) and GPT-OSS 120B (5.0%) largely forgo it. Line length does not separate the models either: Qwen 3.7 Max and Mistral Large 3 request the longest lines (mean 4.2 and 3.9 plies), whereas GPT-5.5 simulates a single ply in 71.9% of accepted calls (Table 5). Line legality separates the extremes but not the middle: GPT-5.5 and Gemini obtain at least one fully legal line in 353/356 and 299/332



Figure 5: **Action mix by model.** Each cell gives the share (%) of all agent REPL calls over complete trajectories spent on each command of Table 1; `simulate` includes budget-rejected calls, `move` counts accepted moves, and `rejected` counts rejected moves and the other rejected calls of Appendix Figure 11. Rows sum to 100%. Hatched cells are commands unavailable under Restricted observation. Red numbers give pass@1 (%).

simulating trajectories, against 14/213 for Mistral Large 3, yet DeepSeek V3.2 reaches 292/352 and still converts almost nothing. **Explicit state queries are rare and not concentrated in the strongest models.** Because the Sighted interface already returns the board and legal moves after every step, the two strongest models almost never call `view_board` or `get_legal_moves` (at most 2.2% of actions); the heaviest users are DeepSeek V3.2 (12% and 8.4%) and DeepSeek V4 Pro (8.6% and 5.8%). **Rejected actions dominate the weakest models under Restricted observation.** Without state refreshes, rejected actions remain at 5–9% of actions for Gemini, GPT-5.5, and Claude Opus 4.7, rise to 24–42% for the middle group, and reach 93% for Mistral Large 3 and 79% for MiniMax M2.5 (Figure 5). These two models contribute 2,101 of the 5,850 Restricted rejections and recover from under 5% of them, whereas Gemini 3.1 Pro, GPT-5.5, Claude Opus 4.7, and Qwen 3.7 Max recover from at least 88% in both settings (Appendix Figure 17; temporal patterns in Appendix F).

6.2 The Simulation Gap: Checking Legality Is Not Anticipating the Opponent

A natural hypothesis is that a forward simulation sandbox ($\mathcal{A}_{\text{sim}}$) lets models verify tactical lines before committing them. The 11,342 simulation requests issued under Sighted observation reveal two limits on what it verifies (Figure 6a,b); a controlled ablation shows that it nonetheless contributes to conversion (Figure 6c).

Legality falls with line length. Of 6,749 accepted simulation calls, 4,474 executed the complete candidate sequence legally, 2,178 (32.3%) stopped on an illegal move, and the remaining 97 ended early at a simulated terminal or the eight-ply cap. Single-ply requests are almost always legal (96.3% of 2,592 calls), but full-line legality falls to 47.6% for 3–4 plies (1,110/2,332) and 37.4% for 5–8 plies (357/955). Within their own scratchpads, models move pieces through palace boundaries, pins, and missing cannon screens.

Assumed opponent replies often differ from the defender's. Every multi-ply simulated line contains a defender reply chosen by the model itself. Among the 1,677 cases in which an agent adopted the first move of a fully legal line on the real board, the actual defender played a different reply in 49.3% (826/1,677), so the model's continuation was no longer the line it had checked. These data record disagreement, not the quality of the assumed reply; the sandbox verifies only the model's own hypothesis about the opponent, which the defender often does not share.

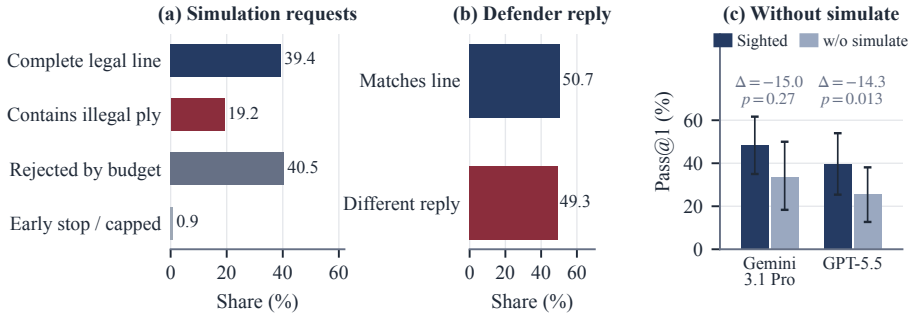


Figure 6: **Simulation checks legality but not the opponent, yet removing it costs wins.** (a) Outcomes of all 11,342 Sighted simulation requests; "early stop / capped" comprises 93 legal simulated terminals and four legal prefixes truncated at eight plies. (b) For the 1,677 adopted candidate lines with a legal assumed defender reply, whether the actual defender chose that reply. (c) Controlled ablation on positions each model solved under Sighted observation, three trials per position (Appendix D): Sighted with and without `simulate`; the ablation also disables `view_board` and `get_legal_moves`, which neither model uses when the state is refreshed. Bars: pass@1 with 95% case-bootstrap intervals; Δ: paired difference in points; p: exact sign test over positions.

The sandbox still helps the strongest models. These limits do not make simulation useless. We re-ran Gemini 3.1 Pro and GPT-5.5 under Sighted observation with and without `simulate`, using a shared prompt and a correct check field, on the 20 and 21 positions each had solved at least once in the main evaluation (Appendix D). The ablation also disables `view_board` and `get_legal_moves`, but with the state refreshed after every step both models almost never call them (at most 0.05 calls per turn, against 0.75 and 0.98 `simulate` calls). Without `simulate`, pass@1 falls from 48.3% to 33.3% for Gemini and from 39.7% to 25.4% for GPT-5.5 (Figure 6c). The paired drop is significant for GPT-5.5 ($p = 0.013$) and has the same sign but is not significant for Gemini ($p = 0.27$). Simulation thus helps the strongest models find wins even though it verifies only their own hypothesis about the opponent. Without state refreshes, disabling the same commands does not lower pass@1 further (Appendix D).

Failing agents are out-raced by the defender. How games end shows the cost of this gap (Figure 7). Wins rarely wander: 142 of 178 Sighted wins and 62 of 78 Restricted-observation wins end exactly at the stored mate distance. Failures end early instead: 77.8% of failed Sighted trials and 83.8% of failed Restricted-observation trials stop before that distance. Only 80 of the 8,568 trajectories reach ply 40: 76 end at the turn cap (29 Sighted and 47 Restricted-observation), and four end in a repetition draw on that ply. Most failures are lost on the board, and the share tracks the defender's counter-threat: 81.8% (Sighted) and 70.4% (Restricted) of failures in the 106 threat-race positions, against 28.2% and 29.8% in the 13 unopposed-mate positions. In the 81 positions whose defender threatens mate in one, 23.8% of Sighted and 30.0% of Restricted-observation trials end at ply 2: the agent's first move does not stop the threat, and the defender mates at once. Failing agents are thus not stalled but overtaken by replies they did not anticipate. Stronger models survive longer even when they lose: across the 12 models, the median relative length of failed trials rises with the number of wins (Spearman $\rho = 0.71$ Sighted, 0.89 Restricted observation).

7 Discussion

The results suggest two explanations for the gap between finding a good move and converting a won position. Both are interpretations consistent with the data rather than established causes.

Amortized policies vs. adversarial search. Converting a won position requires each move to hold up against the defender's best reply, i.e., an approximation of $V^*(s_t) = \max_{a_t} \min_{a_{\text{def}}} V^*(s_{t+1})$ over board states s_t. A language model instead samples each action from an amortized policy $P_\theta(a_t \mid o_t, h_t)$ shaped largely by pre-training. Such a policy can capture familiar motifs, as the 26.1% first-move agreement shows, but nothing in it checks each continuation against counterplay. If a model errs with probability ϵ_t at its t-th move, success on a line of H moves shrinks roughly as

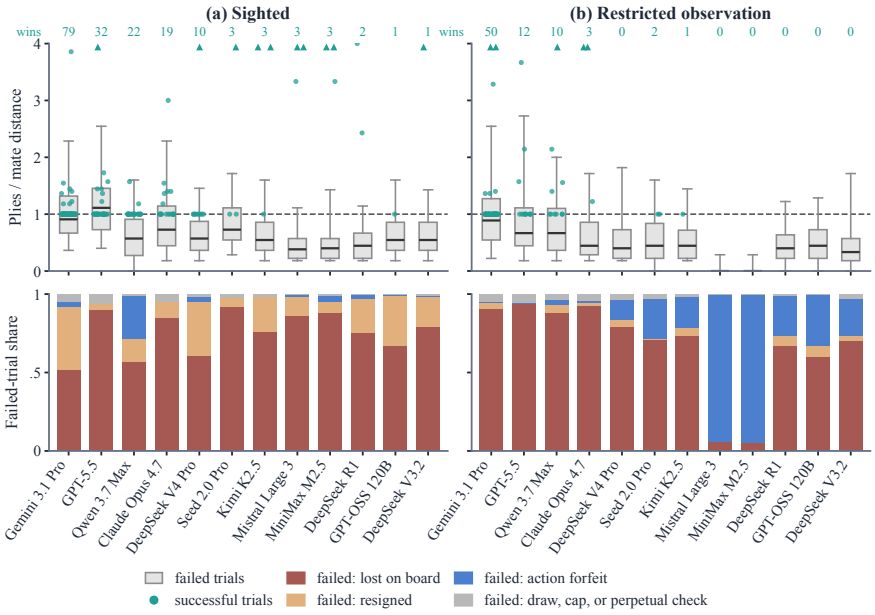


Figure 7: **Failed trajectories end early; successful ones end at the reference length.** Top: plies played divided by the stored mate distance of the case (dashed line: equal length). Boxes summarize failed trials (whiskers: 5th–95th percentiles), points show successful trials, and numbers above the panels give wins out of 357. Triangles mark the 15 wins longer than four times the mate distance, all in JH-084, where the defender loses by perpetual check. Bottom: how failed trials end; *lost on board* means the attacker was left without a legal move, and *action forfeit* covers invalid-action, no-command, and action-limit losses.

$\prod_{t=1}^{H}(1-\epsilon_t)$. This simple picture is consistent with the steep decline in pass@1 with mate distance and with the gap between Gemini's 45.1% first-move agreement and its 22.1% conversion. It is incomplete: errors are not independent across moves, and wins concentrate on particular positions.

Simulation checks legality, not the opponent. The agent writes both sides of every simulated line. A third of accepted simulations stop on an illegal move, and when a legal line began with the move actually played, the real defender chose a different reply in about half of the cases (Section 6). This does not show that the imagined replies were bad, since several defenses can be equally strong, but it does show that the sandbox cannot supply the opponent's choices. Agents that rely on self-authored rollouts in adversarial settings still need an external source of opponent behavior.

Implications for agent evaluation. (i) **Evaluate closed-loop conversion:** agreement on the first move does not imply the ability to finish the game, so benchmarks should include play against an active opponent. (ii) **Report reliability alongside coverage:** pass@k and pass^k diverge sharply here and should be reported together. (iii) **Vary one affordance at a time:** Restricted observation more than halves pass@1, yet a controlled ablation attributes much of that drop to its prompt and feedback rather than the missing board, so observation settings should differ only in the affordance under test.

8 Conclusion

We introduced **XiangqiBench**, an executable benchmark in which language agents must convert search-supported winning Xiangqi endgames into checkmate against an engine defender. Across 8,568 trajectories from 12 frontier LLMs, finding the move is not winning the game: first-move agreement, pass@3, and self-authored simulation each overstate closed-loop success. Closing these gaps likely takes more than better single-step proposals; agents must win reliably and anticipate how the opponent responds. XiangqiBench provides a controlled testbed for measuring both.

AI USE STATEMENT

We used AI tools to help debug code, proof-read paper, and improve readability. We did not use them to generate data, formulate hypotheses or claims, or design the methodology. All state transitions, legality checks, and game outcomes were computed by the deterministic rules engine and Pikafish, never by a generative AI tool. All AI-assisted code was executed and verified, AI-revised text was rewritten and checked by the authors. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT.

The protocol and metrics are specified in Section 4; task construction, defender, record schema, rules replay, and all known provenance gaps are in Appendix A; models and prompts are in Appendices J and B.

REFERENCES

Marwa Abdulhai, Isadora White, Charlie Victor Snell, Charles Sun, Joey Hong, Yuexiang Zhai, Kelvin Xu, and Sergey Levine. LMRL Gym: Benchmarks for multi-turn reinforcement learning with language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 126–153. PMLR, 2025. URL https://proceedings.mlr.press/v267/abdulhai25a.html.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Anthropic. Claude Opus 4.7, 2026. URL https://www.anthropic.com/news/claude-opus-4-7.

Aili Chen, Aonian Li, Baichuan Zhou, Bangwei Gong, Binyang Jiang, Boji Dan, Changhao Zhang, Changqing Yu, Chao Wang, Cheng Ma, et al. The minimax-m2 series: Mini activations unleashing max real-world intelligence. *arXiv preprint arXiv:2605.26494*, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Yuhao Chen, Shuochen Liu, Yuanjie Lyu, Chao Zhang, Jiayao Shi, and Tong Xu. Xiangqi-R1: Enhancing spatial strategic reasoning in LLMs for chinese chess via reinforcement learning. *arXiv preprint arXiv:2507.12215*, 2025. URL https://arxiv.org/abs/2507.12215.

Anthony Costarelli, Mat Allen, Roman Hauksson, Grace Sodunke, Suhas Hariharan, Carlson Cheng, Wenjie Li, Joshua M. Clymer, and Arjun Yadav. GameBench: Evaluating strategic reasoning abilities of LLM agents. *arXiv preprint arXiv:2406.06613*, 2024. URL https://arxiv.org/abs/2406.06613.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. GTBench: Uncovering the strategic reasoning capabilities of LLMs via game-theoretic evaluations. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0885. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/3191170938b6102e5c203b036b7c16dd-Abstract-Conference.html.

Xidong Feng, Yicheng Luo, Ziyan Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Mguni, Yali Du, and Jun Wang. ChessGPT: Bridging policy learning and language modeling. In *Advances in Neural Information Processing Systems*, volume 36, pp. 7216–7262, 2023.

Google DeepMind. Gemini 3.1 Pro, 2026. URL https://deepmind.google/models/gemini/pro/.

Mikaela Grace, Jeremy Hadfield, Rodrigo Olivares, and Jiri De Jonghe. Demystifying evals for AI agents. Anthropic Engineering, January 2026. URL https://www.anthropic.com/engineering/demystifying-evals-for-ai-agents. Published January 9, 2026.

Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. TextArena. *arXiv preprint arXiv:2504.11442*, 2025. URL https://arxiv.org/abs/2504.11442.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Sai Kolasani, Maxim Saplin, Nicholas Crispino, Kyle Montgomery, Jared Quincy Davis, Matei Zaharia, Chi Wang, and Chenguang Wang. LLM Chess: Benchmarking reasoning and instruction-following in LLMs through chess. *arXiv preprint arXiv:2512.01992*, 2025. URL https://arxiv.org/abs/2512.01992.

Wenye Lin, Jonathan Roberts, Yunhan Yang, Samuel Albanie, Zongqing Lu, and Kai Han. GAME-BoT: Transparent assessment of LLM reasoning in games. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 7656–7682, 2025. doi: 10.18653/v1/2025.acl-long.378. URL https://aclanthology.org/2025.acl-long.378/.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Jincheng Liu, Sijun He, Jingjing Wu, Xiangsen Wang, Yang Chen, Zhaoqi Kuang, Siqi Bao, and Yuan Yao. ChessArena: A chess testbed for evaluating strategic reasoning capabilities of large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 7901–7954, 2026. doi: 10.18653/v1/2026.acl-long.360. URL https://aclanthology.org/2026.acl-long.360/.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zAdUB0aCTQ.

Mistral. Introducing mistral 3. 2025. URL https://mistral.ai/news/mistral-3/.

Official Pikafish Contributors. Pikafish: A free and strong UCI xiangqi engine. https://github.com/official-pikafish/Pikafish, 2026. Accessed 2026-07-31.

OpenAI. Introducing GPT-5.5, 2026. URL https://openai.com/index/introducing-gpt-5-5/.

Qwen Team. Qwen3.7: The agent frontier, May 2026. URL https://qwen.ai/blog?id=qwen3.7.

Anian Ruoss, Grégoire Delétang, Sourabh Medapati, Jordi Grau-Moya, Li Kevin Wenliang, Elliot Catt, John Reid, Cannada A. Lewis, Joel Veness, and Tim Genewein. Amortized planning with large-scale transformers: A case study on chess. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2102. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/78f0db30c39c850de728c769f42fc903-Abstract-Conference.html.

Philipp Schmid. Pass@k vs. Pass^k: Understanding agent reliability. Technical blog post, March 2025. URL https://www.philschmid.de/agents-pass-at-k-pass-power-k. Published March 24, 2025.

John Schultz, Jakub Adamek, Matej Jusup, Marc Lanctot, Michael Kaisers, Sarah Perrin, Daniel Hennes, Jeremy Shar, Cannada A. Lewis, Anian Ruoss, Tom Zahavy, Petar Veličković, Laurel Prince, Satinder Singh, Eric Malmi, and Nenad Tomasev. Mastering board games by external and internal planning with language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53581–53644. PMLR, 2025. URL https://proceedings.mlr.press/v267/schultz25a.html.

Bytedance Seed. Seed2. 0 model card: Towards intelligence frontier for real-world complexity. *arXiv preprint arXiv:2607.00248*, 2026.

Jiajun Shi, Jian Yang, Jiaheng Liu, Xingyuan Bu, Jiangjie Chen, Junting Zhou, Kaijing Ma, Zhoufutu Wen, Bingli Wang, Yancheng He, Liang Song, Hualei Zhu, Shilong Li, Xingjian Wang, Wei Zhang, Ruibin Yuan, Yifan Yao, Wenjun Yang, Yunli Wang, Siyuan Fang, Siyu Yuan, Qianyu He, Robert Tang, Yingshui Tan, Wangchunshu Zhou, Zhao-Xiang Zhang, Zhoujun Li, Wenhao Huang, and Ge Zhang. KORGym: A dynamic game platform for LLM reasoning evaluation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5384. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ebfa4297cd6419f64efe86f657ba49d0-Abstract-Conference.html.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Ziwei Chai, Y Charles, HS Che, Cheng Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Shubham Toshniwal, Sam Wiseman, Karen Livescu, and Kevin Gimpel. Chess as a testbed for language model state tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11385–11393, 2022. doi: 10.1609/aaai.v36i10.21390.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, 2024.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R. Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

A Benchmark and Protocol Details

A.1 Case Schema and Construction

Each case record contains a stable identifier, the source and source index, a title, the normalized FEN, the attacking color, a verification label, one stored root move, the reported mate distance in plies, a pseudo-legal root-action count, a position category, piece tags, and an optional counter-mate distance for the side-flipped position. Model outputs are not used to admit individual cases.

For engine-admitted cases, a single-threaded Pikafish process runs a fixed-time search from the position. The case is admitted when the final reported score is a positive mate, and the engine's final preferred move is stored as the root move. Neither the principal variation nor the search log is retained, and the root move was not re-validated by the rules adapter. The construction defaults are 800 ms for the original position and 300 ms for the side-flipped diagnostic, but case records do not

store the parameters actually used. When Pikafish rejects a composed position, a checks-only mate search may admit it, and such cases are labelled separately. Faithful reconstruction would therefore require archived engine and network versions, search logs, and per-case search settings.

The 119 positions are partitioned into 106 threat races and 13 unopposed mates (Section 3). The category is determined by a second search from the same position with Black to move: a threat race is one in which that search finds a forced mate for Black, and an unopposed mate is one in which it does not. Because this search is depth- or time-limited, "unopposed" means that no counter-mate was found within that limit. Pikafish-admitted positions use a time-limited engine search for this check; solver-admitted positions use the solver's counter-threat depth limit.

A.2 Defender and Trajectory Records

On each defender turn, the environment requests a depth-18, single-threaded Pikafish search. The released environment also includes fallbacks that keep a trial running when Pikafish is unavailable. At initialization, if no Pikafish binary is found, it uses Fairy-Stockfish at the same depth; if no engine is found at all, a rule-based defender plays the whole trial. During play, the rule-based defender also answers any single turn on which the engine call raises an error or returns an illegal move. The rule-based defender scores every legal Black move by a depth-5 forced-mate search for Red and prefers, in order, moves with no mate found, moves with the longest mate distance, and checking moves, breaking remaining ties by move string. All 8,568 retained trials record an engine defender at initialization. Records do not identify the engine binary, build, or network file, or whether any individual defender move came from the rule-based fallback.

Each trajectory record contains the initial board and FEN, the move list, the position before and after every move, the legal moves available at each turn, the prompt shown at each interaction, the model's text and commands, token usage when returned by the provider, environment feedback, and the terminal result. Records do not include provider endpoints, response identifiers, full request parameters, or engine search logs. For simulation, the submitted command and the rendered feedback are stored, but not the structured per-ply result; our analysis therefore reads the requested length from the command and recovers per-ply validity from the feedback text.

A.3 Prompts and Command Interface

Both settings use the same command interface. The central action is a coordinate move, e.g., `move h2h9`.

The Sighted system prompt describes an endgame, asks for a forcing checkmate, and lists board display, legal moves, move history, and simulation. Its first user message contains the initial board, FEN, and legal moves. After each move, the feedback can include the board, FEN, legal moves, captures, and check status. The dominant configuration permits one explicit board view and one simulation of at most eight model-supplied plies per turn.

The Restricted-observation system prompt is a generic blindfold xiangqi prompt. It describes the standard starting position and states that the board is not shown, yet the first user message then presents the actual endgame board and FEN once. Thereafter, the model receives move, capture, and check fields without a board, FEN, or legal moves, and the board-view, legal-move, and simulation commands are unavailable. In both settings the check field always read `No` because of the defect described in Appendix A.5. This prompt mismatch is one reason we treat the setting comparison as descriptive; Appendix D tests it with a shared prompt. Both system prompts and the message templates are reproduced in Appendix B.

A.4 Termination, Configuration Variation, and Record Selection

All retained trajectories use a 40-ply interaction cap. In the dominant configuration, a turn allows eight actions, up to three rejected moves, one simulation, one board view, and five consecutive responses without a command. In all 4,284 retained Sighted trajectories, the system prompt stated that exceeding the action limit would force a move, whereas the configuration actually forfeited the trial (Appendix B); the Restricted-observation prompt does not contain this statement, and 25 Sighted trials end by action-limit forfeit. In addition, 75 retained Sighted records use a four-action limit

and two use a five-rejection limit. Scoring treats exhausted rejection or no-command budgets and a turn cap without a win as failures. Records ending in API errors, timeouts, or exceptions, records without a determined winner that are not turn-cap outcomes, and certain action-budget outcomes are excluded from scoring; the excluded set therefore mixes infrastructure and protocol events.

Within each of the 2,856 model–case–setting cells, scored records are ordered by their recorded start time (ties and missing times are ordered by record identifier), and the first three are retained; every cell contains at least three. This yields 8,568 trajectories. Eight additional scored records in three Sighted cells of Mistral Large 3, all failures, are omitted by this rule; the retained trials in these cells also fail, so no per-case success rate changes. Because an API request that exhausts its retries leaves no record, neither the balanced matrix nor the number of extra records indicates how many requests were attempted.

A.5 Rules Replay and Repetition Adjudication

Check flag. The collection environment computed its check flag by calling the rules library's `is_checking()` predicate after applying a move. That predicate asks whether the *side to move* attacks the opposing general. After a move, the side to move is the opponent of the player who just moved, and a legal move never leaves the mover's own general attacked, so the predicate is false after every accepted move. The flag therefore never signalled check: all 57,000 `Check` fields in the messages shown to agents read `No`, as does every per-step `Check` field in accepted simulation output, including after checking and mating moves. The same flag also fed two terminal rules. Every position in which the side to move had no legal move was labelled stalemate, and the collection-time rule that three consecutive checks lose could never fire. Board transitions, move legality, and the winner do not depend on this flag and are unaffected: a side with no legal move loses whether or not it is in check.

Replay. Our replay re-executes all accepted actions, discarding 423 synthetic forfeit entries that repeat the prior move without changing the position. Independent replay with the `cchess` library (version 1.25.5) confirms that the archived check flags are false negatives and partitions the 6,137 terminals recorded as stalemate into 6,034 checkmates and 103 stalemates; the winners of these trials are unchanged.

Repetition. Because the check flag was always false, the collection environment declared every threefold repetition a draw. All reported outcomes instead use the long-check rule of the released environment: at threefold repetition, if exactly one side gave check on every one of its moves since the earliest of the three occurrences, that side loses; any other repetition is a draw. We replay the 138 retained repetition trajectories with the released environment; every replay reproduces the archived positions ply by ply, and only terminal labels change. The rule assigns 43 outcomes to perpetual check. The 23 committed by the attacker (16 Sighted, seven Restricted-observation) remain failures; the 20 committed by the defender (14 and six) become attacker wins. Eighteen of these 20 come from JH-084, one of the two solver-admitted cases, in which the engine-backed defender repeatedly checks. Under the archived all-draw labels, aggregate wins would be 164 Sighted and 72 Restricted-observation, mean pass@1 3.83% and 1.68%, and their difference 2.15 points ([1.38, 2.99]). The rule is simpler than official long-check and long-chase adjudication. More broadly, agents never received a correct check signal, which may have changed their play throughout each trajectory; neither labelling resamples behavior.

A.6 Metrics and Uncertainty

For each model and observation setting, pass@1 equals the mean of 357 binary trajectory outcomes. With three trials per case, observed pass@2/pass@3 use the subset estimator of Chen et al. (2021) for at least one success, and observed pass^2/pass^3 use the all-subsets-success estimator of Yao et al. (2025). In particular, pass@3 is the fraction of 119 cases with at least one win and pass^3 the fraction with three wins. Appendix tables report $k = 1, 2, 3$; the data do not support extrapolation beyond three. The task-level plug-in $(c_i/n_i)^k$, often used for intuitive reporting (Schmid, 2025), is a different finite-sample estimator and is not our primary measure. For the aggregate protocol comparison, the bootstrap samples 119 case identifiers with replacement and carries all 12 models, both observation settings, and three trajectories for every sampled case. The reported percentile

interval uses 4,000 resamples with seed 2027, quantifying case-sampling variation across the 119 evaluation cases.

B PROMPT TEMPLATES

This appendix reproduces the prompts seen by the agent (Red) in the retained trajectories. The system prompts are copied from the trajectory records, and the release includes the script that generates this appendix from them. The Restricted-observation prompt is identical across all 4,284 trajectories. The Sighted prompt shown is used in 4,209 trajectories; the remaining 75 differ only in stating a four-action rather than an eight-action turn limit. Per-turn messages are shown as templates, with *⟨angle-bracketed placeholders⟩* for content that changes from turn to turn. For typesetting, the Chinese glosses that accompany English terms in the original (e.g., piece names and the bilingual turn indicator) are omitted, the board diagram, which is drawn with Chinese piece glyphs, is replaced by a placeholder, and Unicode punctuation is rendered in ASCII; the wording is otherwise unchanged.

Two statements in these prompts differ from the evaluated configuration and scoring. The Sighted prompt says that exceeding the per-turn action limit forces a move, whereas the configuration forfeits the trial (Appendix A). Both prompts describe the 40-ply limit as a draw, which is scored as a failure to convert. The Restricted-observation prompt states that three consecutive checks lose, a rule that never fired during collection and is not used in scoring (Appendix A.5). Finally, although the templates allow `Check: Yes`, every `Check` field in the retained trajectories reads `No` because of the defect described in Appendix A.5.

B.1 SYSTEM PROMPTS

Sighted system prompt

```
You are solving a Chinese Chess (Xiangqi) ENDGAME.

This is NOT the opening -- it is a sharp endgame position and it is YOUR move.
You are the side to move. Your task is to find the precise forcing sequence
that CHECKMATES the opponent. Play exactly: in these positions a single
slack, non-forcing move can let the opponent escape or launch a decisive
counterattack. There is no draw available -- play to win by checkmate.

================================================================================
1. GAME RULES
================================================================================

Board: 9 columns (a-i) x 10 rows (0-9), divided by a River.
Red occupies rows 0-4 (bottom), Black occupies rows 5-9 (top).
Each side has a Palace -- the 3x3 area around the King.

Pieces and Movement:
  K/k  King     -- One step orthogonally, confined to the Palace. The two
                   Kings may NEVER face each other on the same column with
                   no piece between them (Flying General).
  A/a  Advisor  -- One step diagonally, confined to the Palace.
  B/b  Elephant -- Two steps diagonally. Cannot cross the River.
                   Blocked if the midpoint is occupied.
  N/n  Horse    -- "L" shape. Blocked if the first
                   orthogonal square is occupied.
  R/r  Rook     -- Any distance orthogonally; cannot jump.
  C/c  Cannon   -- Moves like a Rook; to CAPTURE it must jump exactly one
                   piece (the "cannon platform").
  P/p  Pawn     -- Forward one step; after crossing the River also sideways.
                   Never backward.

Win / Loss:
  - Checkmate: opponent's King is in check with no legal escape -> you WIN.
  - Stalemate: opponent has no legal move (even if not in check) -> opponent
    LOSES (unlike Western chess).
  - Move limit: if the game runs past 40 half-moves it is scored as a
    draw (i.e. you failed to convert) -- so do not dawdle.

================================================================================
2. MOVE FORMAT (ICCS)
================================================================================
```

```
Each move is exactly 4 characters: <from_col><from_row><to_col><to_row>
  Columns: a b c d e f g h i  (left -> right)
  Rows:    0 1 2 3 4 5 6 7 8 9 (bottom -> top, Red's back rank = 0)
Examples: h2e2 (cannon h2->e2), b0c2 (horse b0->c2), a0a1 (rook a0->a1).

================================================================================
3. BOARD STATE -- WHAT TO TRUST
================================================================================

You get THREE views; trust them in this order:
  1. PIECE POSITIONS census -- coordinate-keyed ground truth of every piece. This
     is authoritative; do NOT rely on memory (a common failure is misreading a
     glyph, e.g. confusing ⟨King glyph⟩ with ⟨Elephant glyph⟩).
  2. FEN string -- compact full-board encoding (ranks listed row 9 -> row 0;
     N=Horse, B=Elephant).
  3. ASCII board -- Chinese glyphs; useful for intuition but easy to misread.
Reconcile your mental model with the census/FEN before every move.

================================================================================
4. AVAILABLE TOOLS
================================================================================

Each turn you MUST eventually call `move` (or `resign`) to end your turn. Before
that you SHOULD `think`, and you MAY use auxiliary tools:

  think(reasoning)      -- **ALWAYS call this first.** Record your analysis of the
                           mating idea, candidate forcing moves, and the lines you
                           calculated. Does not end your turn.
  move(from_to)         -- Play a move. Ends your turn.
  resign()              -- Give up (you lose). Only if the position is truly lost.
  view_board()          -- Show the board with FEN (max 1/turn).
  simulate(moves)       -- Test a hypothetical line WITHOUT affecting the real
                           game. YOU supply both sides: [your_move, assumed_reply,
                           your_move, ...]. Each entry is one half-move. Returns per
                           ply: what moved/captured, check, FEN, and the number of
                           legal replies. Max 8 half-moves,
                           1/turn. Use it to verify a forced mate
                           before committing.
  get_legal_moves()     -- List all legal moves in the current position.
  get_history()         -- Show the full move history.

Recommended flow: think -> (simulate to verify the mate) -> move

After each `move` the environment returns objective facts only (it never tells
you who is winning -- assessing the position is YOUR job): what moved/captured,
whether it is check, material balance, the updated FEN, and the legal moves for
the next position.

================================================================================
5. CONSTRAINTS
================================================================================

  - You can ONLY act via the provided commands/tools; plain text is rejected.
  - You MUST call `think` BEFORE `move`.
  - A turn ends only on `move` or `resign`. Auxiliary tools do not end the turn.
  - If you exceed 8 actions in a turn without moving, a move is
    forced on your behalf.
  - Illegal moves are rejected with the legal-move list; choose a legal move.
  - Keep the attack forcing. Quiet moves usually throw away the win.


================================================================================
INTERACTION PROTOCOL (read carefully)
================================================================================

You control the game through a command-line interface, like a terminal.
This is the ONLY way you may act. Ignore any mention elsewhere of JSON tool
calls or `function(args)` syntax -- in this game you act by typing commands.

On each of your turns:
  1. Think out loud in plain text (your private reasoning -- the opponent never
     sees it).
  2. Then issue EXACTLY ONE command inside a fenced code block:

```bash
move h2e2
```

The environment runs your command and replies with terminal output (prefixed
```

```
with `$ <command>`), then it is your turn again until you end it with a move.

Available commands (one per block):
  move <from><to>            Play a move in ICCS format (e.g. `move h2e2`). ENDS your turn
    ↪ .
  simulate <m1> <m2> ...     Forward-test a hypothetical line you supply for BOTH
                             sides: [your move, assumed opponent move, ...]
                             (e.g. `simulate h2e2 h9g7 b0c2`).
  view_board                 Show the board, FEN, and the ground-truth piece list.
  legal_moves                List all legal moves right now.
  think <reasoning>          Record private reasoning (does NOT end your turn).
  history                    Show the move history.
  resign                     Forfeit the game. ENDS your turn.

Rules:
  - Put your command in a ```bash``` block. Only the LAST block is executed.
  - One command per turn-step. To look before you leap, send `view_board` or
    `simulate ...` first, read the output, then send `move ...`.
  - If you emit no command block, you will be asked again.
  - Illegal or unknown commands are rejected with the list of legal moves; keep
    trying -- you do NOT lose your turn for one mistake, but repeated invalid
    commands in a single turn will be resolved against you.
```

Restricted-observation system prompt

```
You are playing Chinese Chess (Xiangqi) in BLINDFOLD mode.
You interact with the game environment by typing commands in a terminal (see INTERACTION
    ↪ PROTOCOL below).
You CANNOT see the board, FEN, or piece census at any point except the initial state. You
    ↪ must track the
entire position in your own reasoning, using only the move history and the
per-move diffs you are given. Your memory and mental board-tracking ARE the test.

============================================================================
1. GAME RULES
============================================================================

Board: 9 columns (a-i) x 10 rows (0-9), divided by a River.
Red occupies rows 0-4 (bottom), Black occupies rows 5-9 (top).
Each side has a Palace -- the 3x3 area around the King.

Pieces and Movement:
  King     -- One step orthogonally, confined to the Palace. The two
              Kings may NEVER face on the same file with nothing between
              them (Flying General).
  Advisor  -- One step diagonally, confined to the Palace.
  Elephant -- Exactly two steps diagonally. CANNOT cross the
              River. Blocked if the midpoint is occupied.
  Horse    -- "L" shape. Blocked if the first orthogonal step is
              occupied.
  Rook     -- Any distance orthogonally. Cannot jump.
  Cannon   -- Moves like a Rook; to CAPTURE it must jump exactly one
              piece (the platform).
  Pawn     -- Before the River: one step forward only. After crossing:
              one step forward OR sideways. Never backward.

Win / Loss / Draw:
  - Checkmate: opponent's King in check with no legal escape -> you win.
  - Stalemate: opponent has no legal move but is NOT in check -> opponent
    LOSES (unlike Western chess).
  - Perpetual check: checking 3+ times in a row -> the checking side loses.
  - Threefold repetition -> draw (or the aggressor loses).
  - Move limit: exceeding 40 half-moves -> draw.

============================================================================
2. COORDINATES & INITIAL POSITION
============================================================================

Columns a-i (left->right), rows 0-9 (bottom->top). Red's back rank is row 0,
Black's back rank is row 9. Moves are 4-char ICCS, e.g. 'h2e2'.

Red:   King e0 | Advisors d0 f0 | Elephants c0 g0 | Horses b0 h0 | Rooks a0 i0
       Cannons b2 h2 | Pawns a3 c3 e3 g3 i3
Black: King e9 | Advisors d9 f9 | Elephants c9 g9 | Horses b9 h9 | Rooks a9 i9
       Cannons b7 h7 | Pawns a6 c6 e6 g6 i6

============================================================================
3. CONSTRAINTS
```

```
================================================================================

- You have NO board viewer, NO simulator, and NO legal-move list. The ONLY
  game-changing action is `move`. (You may also `think` to record reasoning.)
- After each move you get only a diff: what moved/was captured, check, material.
  After the opponent moves you get the same diff for their move -- never a board.
- If you make 3 illegal moves in a single turn, you FORFEIT.
- You cannot see your opponent's reasoning -- only their moves.


================================================================================
INTERACTION PROTOCOL (read carefully)
================================================================================

You control the game through a command-line interface, like a terminal.
This is the ONLY way you may act. Ignore any mention elsewhere of JSON tool
calls or `function(args)` syntax -- in this game you act by typing commands.

On each of your turns:
  1. Think out loud in plain text (your private reasoning -- the opponent never
     sees it).
  2. Then issue EXACTLY ONE command inside a fenced code block:

```bash
move h2e2
```

The environment runs your command and replies with terminal output (prefixed
with `$ <command>`), then it is your turn again until you end it with a move.

BLINDFOLD: there is no board viewer, simulator, or legal-move list. Track the position
    ↪ from the move history and per-move diffs alone. Repeated illegal moves in one turn
    ↪ forfeit the game.

Available commands (one per block):
  move <from><to>   Play a move in ICCS format (e.g. `move h2e2`). ENDS your turn
    ↪ .
  think <reasoning>   Record private reasoning (does NOT end your turn).
  history   Show the move history.
  resign   Forfeit the game. ENDS your turn.

Rules:
  - Put your command in a ```bash``` block. Only the LAST block is executed.
  - If you emit no command block, you will be asked again.
  - Illegal or unknown commands are rejected; repeated invalid commands in a single turn
    ↪ will be resolved against you.
```

B.2 PER-TURN MESSAGES

In both settings the first user message shows the initial position once. Afterwards, Sighted feedback repeats the full state after every ply, while Restricted-observation feedback reports only the move itself.

Sighted | first user message

```
Game started. You are playing as RED.

Initial board:

⟨board diagram with file letters a-i and rank numbers 0-9⟩

Current turn: Red (Uppercase)
Move format: ⟨gloss of the ICCS format, e.g. h2e2 moves the piece on h2 to e2⟩

FEN: ⟨FEN⟩ w

Legal moves (⟨n⟩): ⟨comma-separated ICCS moves⟩

Type one command inside a ```bash``` block to act (e.g. `move h2e2`). You may `view_board`,
    ↪ `simulate ...`, or `legal_moves` first, then `move` to end your turn.
```

Sighted | after the agent's move, then after the defender's reply

```
Move executed: ⟨move⟩
Result: Red ⟨piece⟩ ⟨from⟩->⟨to⟩ ⟨no capture | captures piece⟩
Capture: you captured ⟨piece⟩.                          ⟨only on a capture⟩
Check: ⟨Yes - you are giving check!  | No⟩
Material -- Red: R⟨n⟩ N⟨n⟩ B⟨n⟩ A⟨n⟩ C⟨n⟩ P⟨n⟩ | Black: ⟨same
     fields⟩ | Balance: ⟨side⟩ ⟨value⟩ (nominal)
Board after your move:

⟨board diagram⟩

Current turn: Black (Lowercase)
Move format: ⟨gloss of the ICCS format⟩
FEN: ⟨FEN⟩ b
Legal moves (⟨n⟩): ⟨Black's legal moves⟩

──────────────────────────────────────────────────────────────────────────────
Opponent (black) played: ⟨move⟩ -- Black ⟨piece⟩ ⟨from⟩->⟨to⟩ ⟨no capture | captures piece⟩
Check: ⟨Yes | No⟩
You LOST your ⟨piece⟩ on this move.                ⟨only on a capture⟩
Move ⟨k⟩ -- FEN: ⟨FEN⟩ w

Current board:

⟨board diagram⟩

Current turn: Red (Uppercase)
Move format: ⟨gloss of the ICCS format⟩

Piece positions (ground truth -- trust this over your memory):
  Red : ⟨piece-type list with coordinates, e.g. King e0 | Rook i5 | Cannon g9,h5⟩
  Black: ⟨piece-type list with coordinates⟩

Legal moves (⟨n⟩): ⟨comma-separated ICCS moves⟩

It is now your turn. Call a tool to proceed.
```

Restricted observation | first user message

```
Game started. You are playing as RED.

Initial board:

⟨board diagram with file letters a-i and rank numbers 0-9⟩

Current turn: Red (Uppercase)
Move format: ⟨gloss of the ICCS format, e.g. h2e2 moves the piece on h2 to e2⟩

FEN: ⟨FEN⟩ w

BLINDFOLD: this is the ONLY time you are shown the board. From now on you get only per-
    ↪ move diffs -- no board, FEN, or legal moves. Track the position yourself.

Type one command inside a ```bash``` block to act (e.g. `move h2e2`). You may `think ...`
    ↪ first, then `move` to end your turn. There is no board viewer or legal-move list.
```

Restricted observation | after the agent's move, then after the defender's reply

```
Move executed: ⟨move⟩
Result: Red ⟨piece⟩ ⟨from⟩->⟨to⟩ ⟨no capture | captures piece⟩
Capture: you captured ⟨piece⟩.                          ⟨only on a capture⟩
Check: ⟨Yes - you are giving check!  | No⟩
Material -- Red: R⟨n⟩ N⟨n⟩ B⟨n⟩ A⟨n⟩ C⟨n⟩ P⟨n⟩ | Black: ⟨same
     fields⟩ | Balance: ⟨side⟩ ⟨value⟩ (nominal)

──────────────────────────────────────────────────────────────────────────────
Opponent (black) played: ⟨move⟩ -- Black ⟨piece⟩ ⟨from⟩->⟨to⟩ ⟨no capture | captures piece⟩
Check: ⟨Yes - you are in check!  | No⟩
⟨No capture.  | You LOST your piece on this move.⟩
It is now your turn.
```

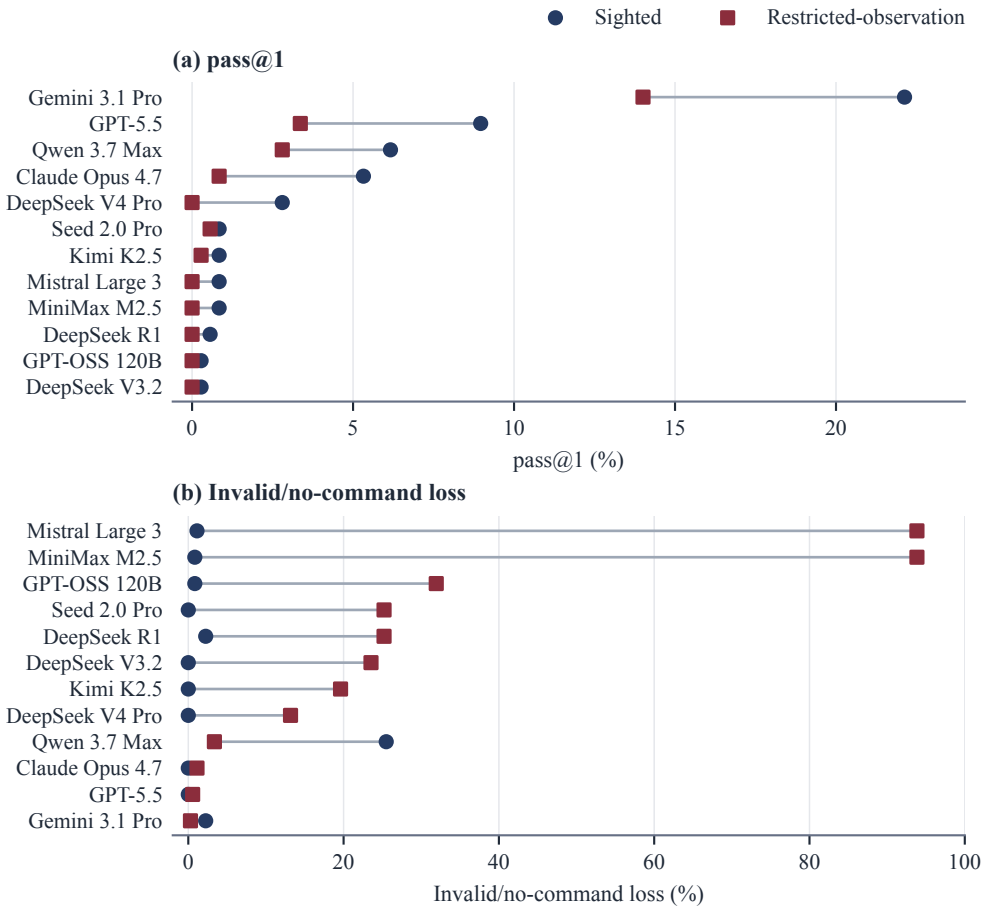


Figure 8: **The setting difference varies across models.** (a) Pass@1; (b) terminal invalid/no-command loss rate. Each point uses 119 cases and three retained trials per case in each setting. These compare the complete evaluation settings, not an isolated memory intervention.

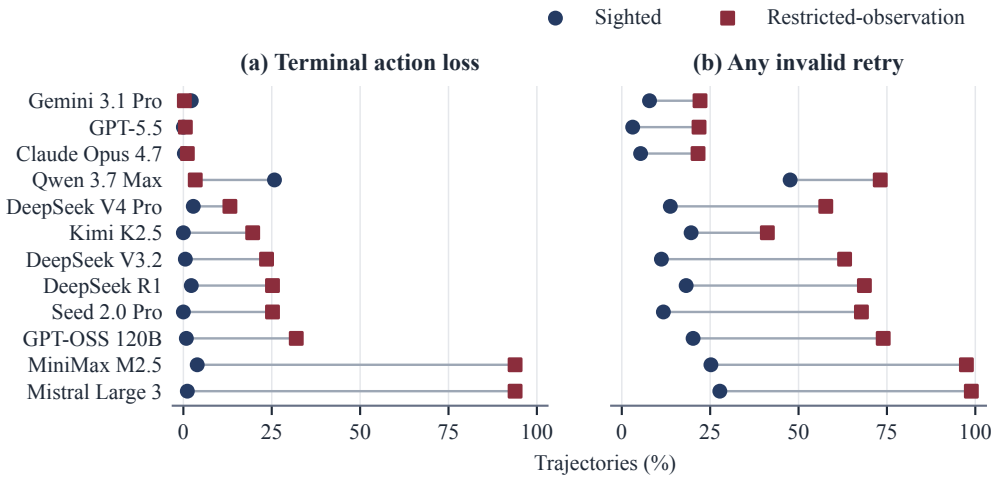


Figure 9: **Action-level failures concentrate under Restricted observation.** Left: terminal action losses (invalid-action, no-command, or action-limit); right: trajectories with any rejected move attempt. Each point has 357 trials. Because a rejected attempt can be followed by recovery, these are different measures of interaction failure, not interchangeable indicators of strategic quality.

C Additional Setting and Failure Diagnostics

Game length. Section 6.2 summarizes Figure 7. No win ends before the stored mate distance, and the median failed trial lasts 6 plies under Sighted and 4 under Restricted observation. Of all failed trials, 76.1% (Sighted) and 66.0% (Restricted) are lost on the board. Resignation ends 18.4% of failed Sighted trials (40.3% for Gemini 3.1 Pro) but only 3.2% under Restricted observation, where action forfeits end 28.2%; Mistral Large 3 and MiniMax M2.5 forfeit 310 and 313 of their 357 trials before playing a move.

D Observation Ablation

Restricted observation differs from Sighted in more than what the agent observes. It removes the state refreshes and the query and simulation commands, but it also uses a different system prompt (Appendix A.3), and in both settings the check field was defective (Appendix A.5). To separate these factors, we ran a smaller ablation after the main evaluation.

Design. All arms share one endgame system prompt. Its rules and move-format sections are identical to the Sighted prompt; only the paragraphs that describe the observation and list the commands change with the arm. The check field is correct, and the prompt states the configured consequence of exceeding the action limit (forfeit). Two factors are crossed: *state refreshes*, i.e., the board, FEN, and legal moves attached to every move result, rejection, and defender notice; and *query commands* (`view_board`, `legal_moves`, `simulate`). Arm S has both; S-NT keeps the refreshes and removes the commands; R-T removes the refreshes and keeps the commands, so the agent must request the state; R removes both. Without refreshes, the first message shows the board and FEN once, and later feedback reports only the move, capture, check, and material. The defender, budgets, and scoring match the main evaluation.

Models and positions. We use Gemini 3.1 Pro and GPT-5.5, the two strongest models in the main evaluation. A position that a model never solves scores zero in every arm, so each model is evaluated on positions it won at least once under Sighted: 20 of Gemini's 46, drawn by mate distance with a fixed seed, and all 21 of GPT-5.5's. Every model–position–arm cell has three scored trials; trials lost to API errors were rerun. GPT-5.5 was served through a different API gateway than in the main evaluation, with the same model version. Because the positions were selected on main-evaluation Sighted wins, the main-evaluation Sighted rates on them are biased upward. We therefore compare arms within the ablation and report the main evaluation only for reference.

Table 2: **Observation ablation.** Pass@1 over three trials per position. Solved: positions won at least once. Rej.: share of move attempts rejected. View, Legal, Sim.: query calls per agent turn. The main-evaluation rows use the same positions; their Sighted rate is biased upward by selection.

Model	Arm	Refresh	Commands	Pass@1 (%)	Solved	Rej. (%)	View	Legal	Sim.
Gemini 3.1 Pro	S	✓	✓	48.3	17/20	1.7	0.00	0.00	0.75
	S-NT	✓	✗	33.3	11/20	0.7	–	–	–
	R-T	✗	✓	35.0	14/20	4.0	0.32	0.19	1.00
	R	✗	✗	41.7	15/20	6.9	–	–	–
	Main Sighted			51.7					
	Main Restricted			20.0					
GPT-5.5	S	✓	✓	39.7	15/21	0.3	0.01	0.05	0.98
	S-NT	✓	✗	25.4	10/21	1.9	–	–	–
	R-T	✗	✓	34.9	13/21	2.0	0.38	0.67	1.03
	R	✗	✗	38.1	15/21	3.6	–	–	–
	Main Sighted			50.8					
	Main Restricted			9.5					

Findings. Each comparison below rests on 20 or 21 positions; the intervals are 95% case-bootstrap intervals and the p-values are exact sign tests over positions. (i) Removing the query commands lowers pass@1 for both models (S minus S-NT: 15.0 points for Gemini, [0.0, 30.0]; 14.3 points

for GPT-5.5, [1.6, 23.8], $p = 0.013$). The agents lose simulation, which they use on most S turns. (ii) Removing the refreshes but keeping the commands lowers Gemini's pass@1 by 13.3 points ([0.0, 25.0], $p = 0.039$) and GPT-5.5's by 4.8 points ([−11.1, 19.0]). Both models then request the state, with 0.32 and 0.38 board views and 0.19 and 0.67 legal-move requests per turn, which they almost never do when the state is refreshed. (iii) With neither refreshes nor commands, GPT-5.5 wins 38.1% of trials and Gemini 41.7%, close to S (S minus R: 1.6 points, [−12.7, 15.9], and 6.7 points, [−6.7, 21.7]) and far above the 9.5% and 20.0% they reach on the same positions under Restricted observation. Without refreshes, removing the commands does not lower pass@1 further (R minus R-T: 3.2 and 6.7 points, neither significant), so the command effect in (i) holds only when the state is refreshed; with three trials per position we do not interpret this interaction further. Most of the Restricted drop reported in Section 5 therefore reflects the setting as a whole, including its prompt and feedback, rather than withholding the board alone. The ablation does not identify which of these remaining differences matters most.

E Tool-Use and Failure Diagnostics

The simulation tables report setting coverage, per-model requests and accepted calls, request length, and mutually exclusive turn patterns; the failure tables separate terminal action losses from recoverable invalid attempts. GPT-5.5 predominantly submits one-ply simulations, whereas several peers more often submit multi-ply continuations; this describes action patterns and is not evidence that either style causes better play. Across all models, complete-line legality decreases from 2,496/2,592 (96.3%) for one-ply requests to 1,110/2,332 (47.6%) for three–four plies and 357/955 (37.4%) for five–eight plies. Longer requests create more opportunities for an illegal move, but length, model, case, and decision state vary together.

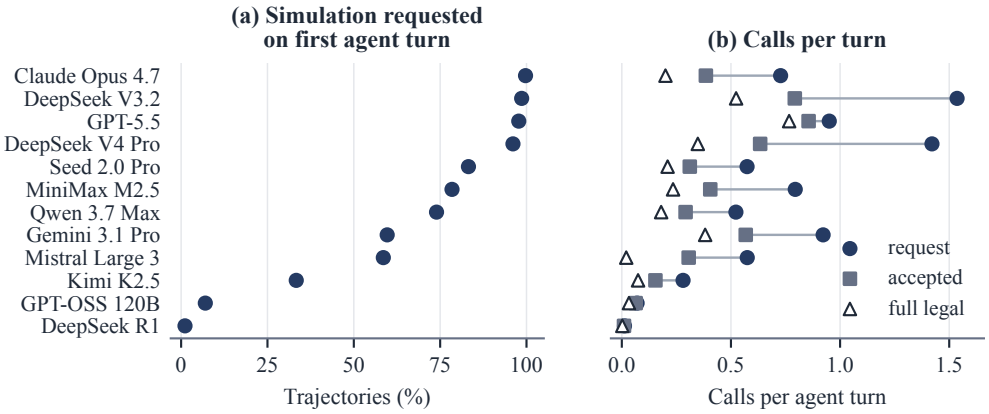


Figure 10: **Simulation use varies widely across models.** (a) Trajectories requesting simulation on the first agent turn (357 per model). (b) Simulation requests, accepted calls, and complete legal candidate lines per agent turn (15,318 turns). Requests include budget-rejected calls, and acceptance does not imply a legal line. Simulation is unavailable under Restricted observation.

Tables 8–11 characterize the first real move, comparing against the uniform baseline among all legal candidate moves. For models with high early forfeit rates under Restricted observation (e.g., Mistral Large 3 and MiniMax M2.5), profiles condition on trajectories that successfully registered an initial accepted move.

Table 3: Explicit simulation use by protocol. All entries count trajectories. Behavior rows overlap; acceptance of a simulation call does not imply full-line legality.

Trajectory measure	Sighted	Restricted-observation
Trajectory denominator	4284	4284
Requested simulation	3098	0
At least one accepted simulation call	3098	0
At least one full legal requested line	2155	0
At least one mid-line invalid step	1639	0
At least one capped request	32	0
Only budget/protocol rejected	0	0

Table 4: Sighted explicit simulation use by model. Any and Full count trajectories out of 357: any request and at least one full legal original request, respectively. Accepted and Rejected count calls; all rejections here are budget rejections.

Model	Any	Full	Accepted	Rejected
Gemini 3.1 Pro	332	299	1081	675
GPT-5.5	356	353	1727	191
Qwen 3.7 Max	290	188	360	284
Claude Opus 4.7	357	241	595	529
DeepSeek V4 Pro	354	266	810	1006
Seed 2.0 Pro	328	222	426	359
Kimi K2.5	156	76	180	148
GPT-OSS 120B	63	33	70	7
DeepSeek R1	8	2	9	2
DeepSeek V3.2	352	292	890	835
Mistral Large 3	213	14	242	212
MiniMax M2.5	289	169	359	345

Table 5: Original request length among accepted Sighted simulation calls. Length is measured in plies. Two or more plies include model-supplied assumed opponent replies, not engine search. Length and position differences prevent interpreting legality rates as planning ability.

Model	Median	Mean	1 ply (%)	$\geq$2 plies (%)
Gemini 3.1 Pro	3	3.57	12.5	87.5
GPT-5.5	1	1.52	71.9	28.1
Qwen 3.7 Max	4	4.20	9.2	90.8
Claude Opus 4.7	3	3.38	5.5	94.5
DeepSeek V4 Pro	3	3.12	27.0	73.0
Seed 2.0 Pro	2	2.20	46.7	53.3
Kimi K2.5	3	3.33	24.4	75.6
GPT-OSS 120B	1.5	1.86	50.0	50.0
DeepSeek R1	3	3.44	22.2	77.8
DeepSeek V3.2	2	1.92	48.4	51.6
Mistral Large 3	3	3.93	3.7	96.3
MiniMax M2.5	1	1.68	58.5	41.5

Table 6: Mutually exclusive agent-turn patterns, assigned by the first applicable row in the displayed priority order. Entries are counts (percent), with 15,318 Sighted and 12,387 Restricted-observation agent turns. Recovery means only that a real move followed a rejected or aborted simulation, not that the strategy improved. Direct execution excludes explicit auxiliary actions but does not imply the absence of automatically supplied state information.

Turn pattern (priority order)	Sighted n (%)	Restricted-observation n (%)
No executed move	898 (5.9)	1,319 (10.6)
Simulation rejected/aborted, then recovery	4,386 (28.6)	0 (0.0)
Simulate, then adopt tested root	1,447 (9.4)	0 (0.0)
Simulate, then choose different root	774 (5.1)	0 (0.0)
Observe, then move	234 (1.5)	15 (0.1)
Direct execution	6,988 (45.6)	10,581 (85.4)
Other explicit action, then move	591 (3.9)	472 (3.8)

Table 7: Mutually exclusive observed failure classes. Opponent terminal collects the remaining failures; on replay, each ends with the attacker checkmated or stalemated. Classes follow the released long-check rule (Appendix A.5).

Observed failure class	Sighted	Restricted-observation
All observed failures	4106	4206
Invalid-action limit	12	1178
No command/tool call	105	6
Perpetual check by the attacker	16	7
Repetition draw	39	56
Action limit	25	0
Turn cap	29	47
Resignation	756	135
Opponent terminal (attacker cannot move)	3124	2777

Table 8: Root-action composition in the Sighted setting. Entries are observed percent (percentage-point difference from uniform choice among true legal root moves). For each model, the reference uses exactly the cases with an accepted first move. Check and capture overlap; quiet means neither. These are action patterns, not personality labels.

Model	Accepted	Check	Capture	Quiet
Gemini 3.1 Pro	353	95.5 (+84.8)	25.5 (+18.7)	4.2 (-80.8)
GPT-5.5	357	95.8 (+85.1)	25.5 (+18.7)	3.6 (-81.3)
Qwen 3.7 Max	287	88.9 (+77.9)	27.2 (+20.3)	8.7 (-76.0)
Claude Opus 4.7	356	84.6 (+73.8)	29.8 (+23.0)	10.7 (-74.3)
DeepSeek V4 Pro	350	81.4 (+70.7)	27.7 (+21.0)	14.6 (-70.4)
Seed 2.0 Pro	357	88.0 (+77.2)	33.3 (+26.6)	7.3 (-77.7)
Kimi K2.5	357	72.8 (+62.1)	26.3 (+19.5)	19.9 (-65.1)
GPT-OSS 120B	352	71.9 (+61.2)	20.7 (+14.0)	22.2 (-62.8)
DeepSeek R1	352	51.4 (+40.7)	34.7 (+27.8)	34.7 (-50.3)
DeepSeek V3.2	355	70.4 (+59.7)	41.1 (+34.3)	15.8 (-69.2)
Mistral Large 3	355	37.5 (+26.7)	25.6 (+18.8)	46.5 (-38.5)
MiniMax M2.5	344	34.9 (+24.2)	27.0 (+20.3)	52.6 (-32.4)

Table 9: Root piece-choice differences (percentage points) in the Sighted setting. Differences are relative to uniform legal choice on the same accepted-first-move case distribution. Other comprises general, advisor, and elephant. Intervals are omitted for readability, and no multiple-comparison correction is applied. These exploratory contrasts are not evidence of piece-specific playing strength.

Model	Rook	Cannon	Horse	Pawn	Other
Gemini 3.1 Pro	+10.9	-22.5	+3.2	+10.8	-2.5
GPT-5.5	+16.1	-17.5	-5.7	+8.7	-1.6
Qwen 3.7 Max	+4.7	-17.8	+6.2	+8.8	-1.9
Claude Opus 4.7	+4.7	-22.2	+11.2	+9.3	-3.0
DeepSeek V4 Pro	+11.7	-21.9	+7.7	+4.5	-1.9
Seed 2.0 Pro	+24.2	-27.3	-4.0	+9.9	-2.7
Kimi K2.5	+4.3	-16.9	+6.9	+8.7	-3.0
GPT-OSS 120B	+16.5	-16.6	+5.4	-2.7	-2.6
DeepSeek R1	+17.8	-22.3	+2.1	+3.5	-1.1
DeepSeek V3.2	+11.8	-21.6	+2.5	+9.9	-2.7
Mistral Large 3	+1.7	-12.0	+14.7	-1.6	-2.8
MiniMax M2.5	+12.0	-25.8	+11.9	+3.2	-1.4

Table 10: Root-action composition in the Restricted-observation setting. Entries are observed percent (percentage-point difference from uniform choice among true legal root moves). For each model, the reference uses exactly the cases with an accepted first move. Check and capture overlap; quiet means neither. These are action patterns, not personality labels.

Model	Accepted	Check	Capture	Quiet
Gemini 3.1 Pro	356	82.6 (+71.9)	26.7 (+19.9)	10.4 (-74.6)
GPT-5.5	357	72.5 (+61.8)	37.8 (+31.0)	13.7 (-71.3)
Qwen 3.7 Max	352	69.9 (+59.1)	26.7 (+19.9)	18.8 (-66.2)
Claude Opus 4.7	355	58.9 (+48.1)	45.1 (+38.3)	19.7 (-65.2)
DeepSeek V4 Pro	341	43.4 (+32.7)	36.1 (+29.3)	34.0 (-50.9)
Seed 2.0 Pro	334	55.7 (+45.0)	35.3 (+28.7)	32.0 (-53.0)
Kimi K2.5	340	52.1 (+41.4)	46.5 (+39.6)	25.3 (-59.7)
GPT-OSS 120B	320	61.3 (+50.7)	41.9 (+35.0)	24.1 (-61.0)
DeepSeek R1	325	51.7 (+40.8)	42.5 (+35.7)	28.3 (-56.5)
DeepSeek V3.2	303	36.0 (+25.3)	29.4 (+22.9)	50.2 (-35.1)
Mistral Large 3	47	12.8 (+5.1)	0.0 (-3.7)	87.2 (-2.7)
MiniMax M2.5	44	27.3 (+18.9)	4.5 (-0.5)	72.7 (-15.7)

Table 11: Root piece-choice differences (percentage points) in the Restricted-observation setting. Differences are relative to uniform legal choice on the same accepted-first-move case distribution. Other comprises general, advisor, and elephant. Intervals are omitted for readability, and no multiple-comparison correction is applied. These exploratory contrasts are not evidence of piece-specific playing strength.

Model	Rook	Cannon	Horse	Pawn	Other
Gemini 3.1 Pro	+11.4	-19.4	+2.5	+7.6	-2.2
GPT-5.5	+11.6	-17.5	-1.2	+8.2	-1.1
Qwen 3.7 Max	+0.5	-15.5	+5.2	+9.2	+0.6
Claude Opus 4.7	+5.4	-27.8	+10.4	+13.0	-1.1
DeepSeek V4 Pro	+7.7	-19.5	+4.4	+6.3	+1.1
Seed 2.0 Pro	+18.7	-16.1	-6.8	+7.0	-2.8
Kimi K2.5	+4.9	-21.5	+4.3	+13.8	-1.6
GPT-OSS 120B	+18.9	-24.1	-0.5	+7.6	-1.8
DeepSeek R1	+25.2	-25.6	-1.3	+2.8	-1.1
DeepSeek V3.2	+7.0	-22.6	+8.4	+2.4	+4.8
Mistral Large 3	-2.3	+5.9	-5.5	+3.9	-1.9
MiniMax M2.5	+1.2	-4.4	-1.0	+2.2	+2.0

F REPL-Call Composition Along Trajectories

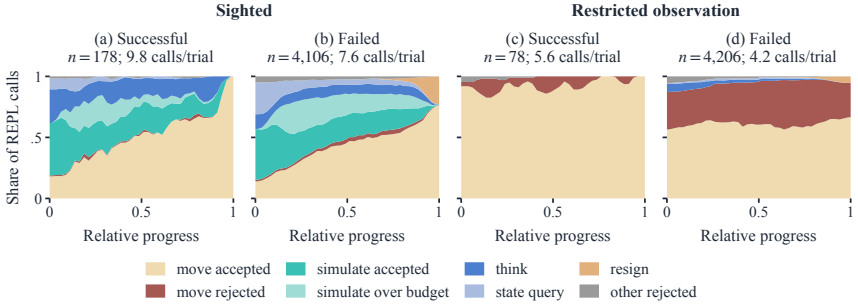


Figure 11: **REPL-call composition along trajectories.** Calls from all 12 LLMs are combined over normalized trajectory progress $[0, 1]$; each agent call belongs to exactly one category. *State query* counts accepted read-only perception calls of Table 1 (`view_board`, `get_legal_moves`, `get_history`; only `get_history` under Restricted observation); *other rejected* covers unknown or malformed commands, `think` calls without reasoning, and over-limit or disabled queries. Successful Sighted trials devote more calls to `think` and fewer to over-budget simulation than failed ones; failed Restricted trials contain a high share of rejected moves throughout.

Figure 11 tracks the composition of REPL calls across relative trial progress $[0, 1]$.

- **Deliberation versus over-budget simulation (Sighted):** Successful trials spend 14.9% of their calls on `think` (17.4% in the first quarter of the trajectory), against 7.6% in failed trials. Failed trials spend 14.2% of calls on simulation requests rejected for exceeding the per-turn budget, against 8.6% in successes; these requests do not advance the board. In the last quarter of failed trajectories, resignation rises to 9.3% of calls.
- **Persistent illegality (Restricted):** In failed Restricted trials, rejected moves make up 33.3% of all calls, against 6.7% in successful ones. The share is high from the start (29.1% in the first quarter, 33.1% in the last), indicating that state-tracking errors arise early rather than accumulating only as the position deteriorates.

Across the benchmark, the attacker being checkmated or stalemated remains the dominant failure mode (3,124 Sighted, 2,777 Restricted), while Restricted observation additionally produces 1,178 terminal invalid-action forfeits (Table 7).

The remaining figures use the construction of Fig. 11; environment-generated forfeit records are excluded. Figure 12 adds panels over all trials of each setting. Figure 13 splits the four Sighted models with the most successes by outcome. Within Gemini 3.1 Pro, successful trials spend 14.0% of calls on `think` and 8.7% on over-budget simulation, versus 2.4% and 19.0% in failed trials. GPT-5.5 differs little between outcomes and interleaves `think` and simulation until its final move (`think` 24.4% versus 21.2%; accepted simulation 32.2% versus 33.3%). Qwen 3.7 Max failures contain more other rejected calls (8.5% versus 2.3%). Figures 14 and 15 show all 12 LLMs in each setting. Under Restricted observation, rejected moves range from 5.2% of Gemini 3.1 Pro's calls and 5.6% of GPT-5.5's to 77.6% of MiniMax M2.5's and 92.9% of Mistral Large 3's; Qwen 3.7 Max is the only model whose other rejected calls (13.6%) outnumber its rejected moves. These are descriptive within-model contrasts with at most 79 successes per model.

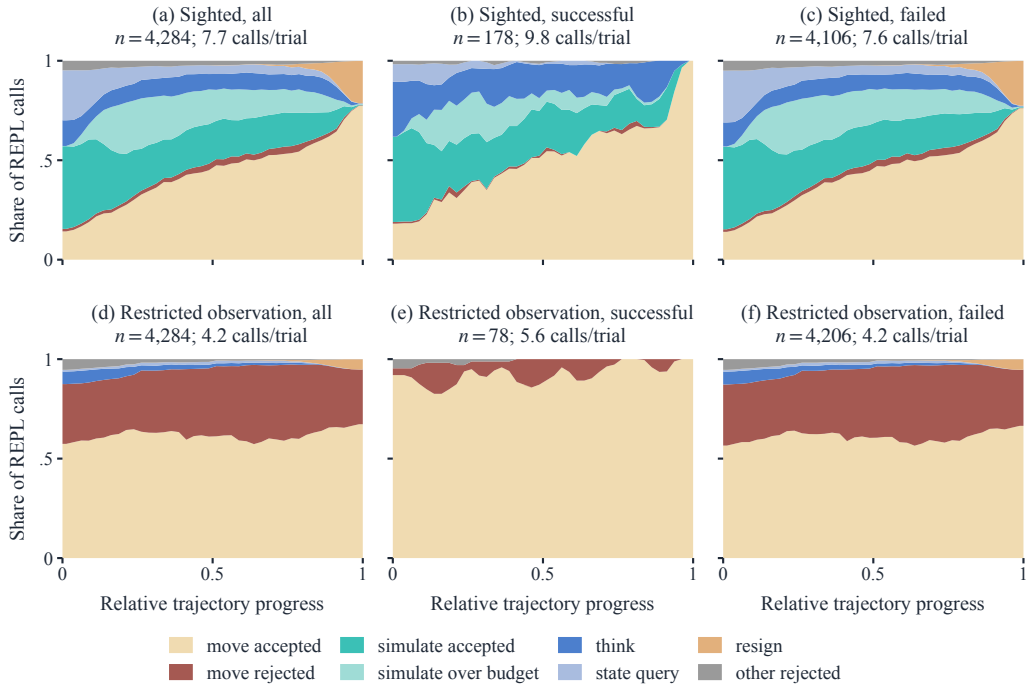


Figure 12: **REPL-call composition by setting and outcome, combining all 12 LLMs.** Columns show all, successful, and failed trials; construction and category definitions follow Fig. 11.

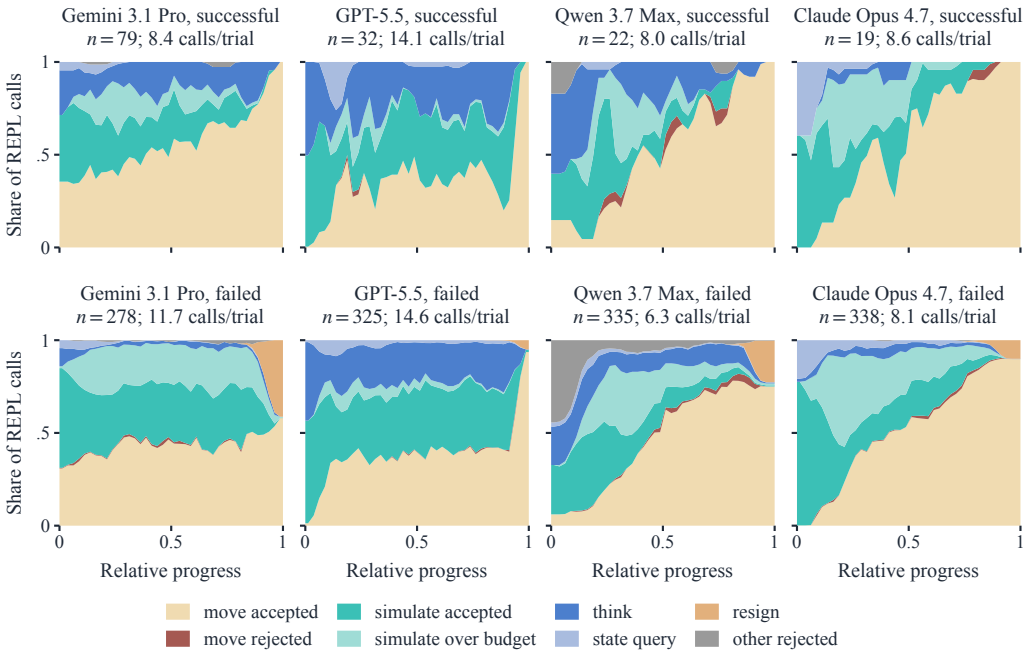


Figure 13: **Sighted REPL-call composition by outcome for the four models with the most successes.** Successful panels contain 19–79 trials and are noisier than failed panels.

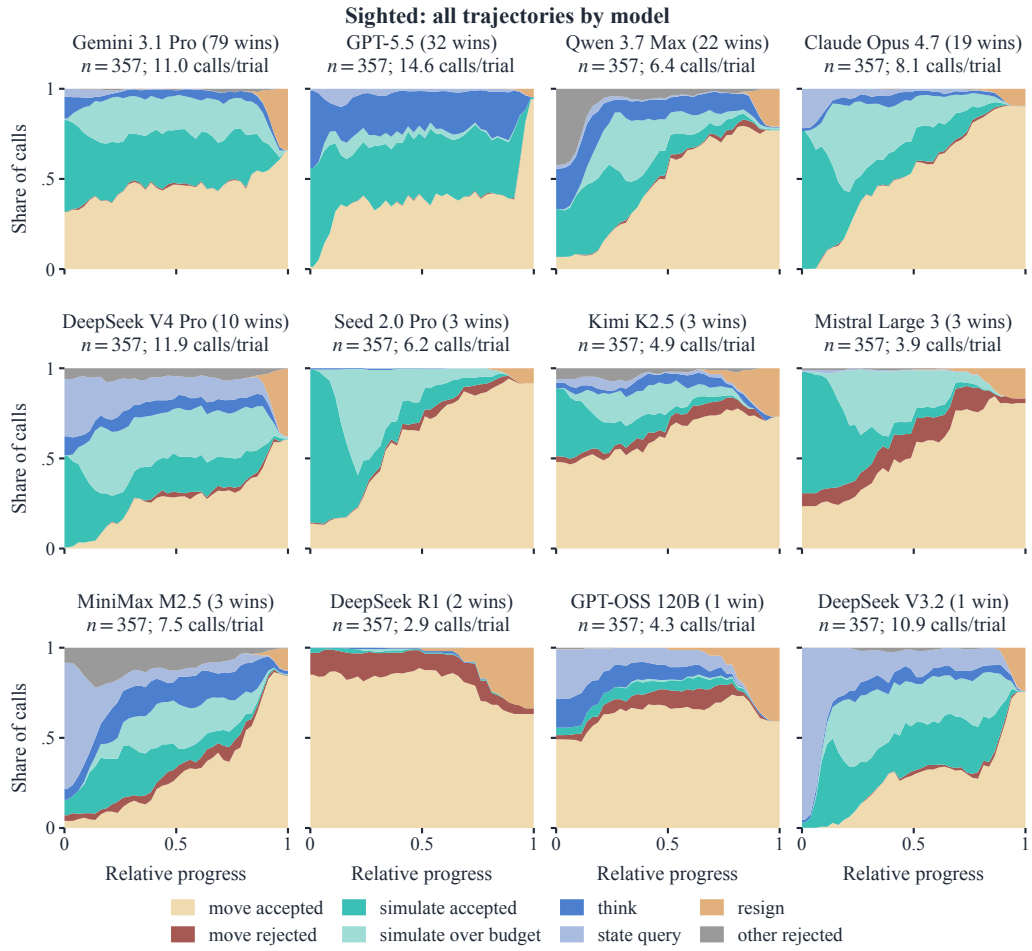


Figure 14: **Sighted REPL-call composition for each LLM over all 357 trials.**

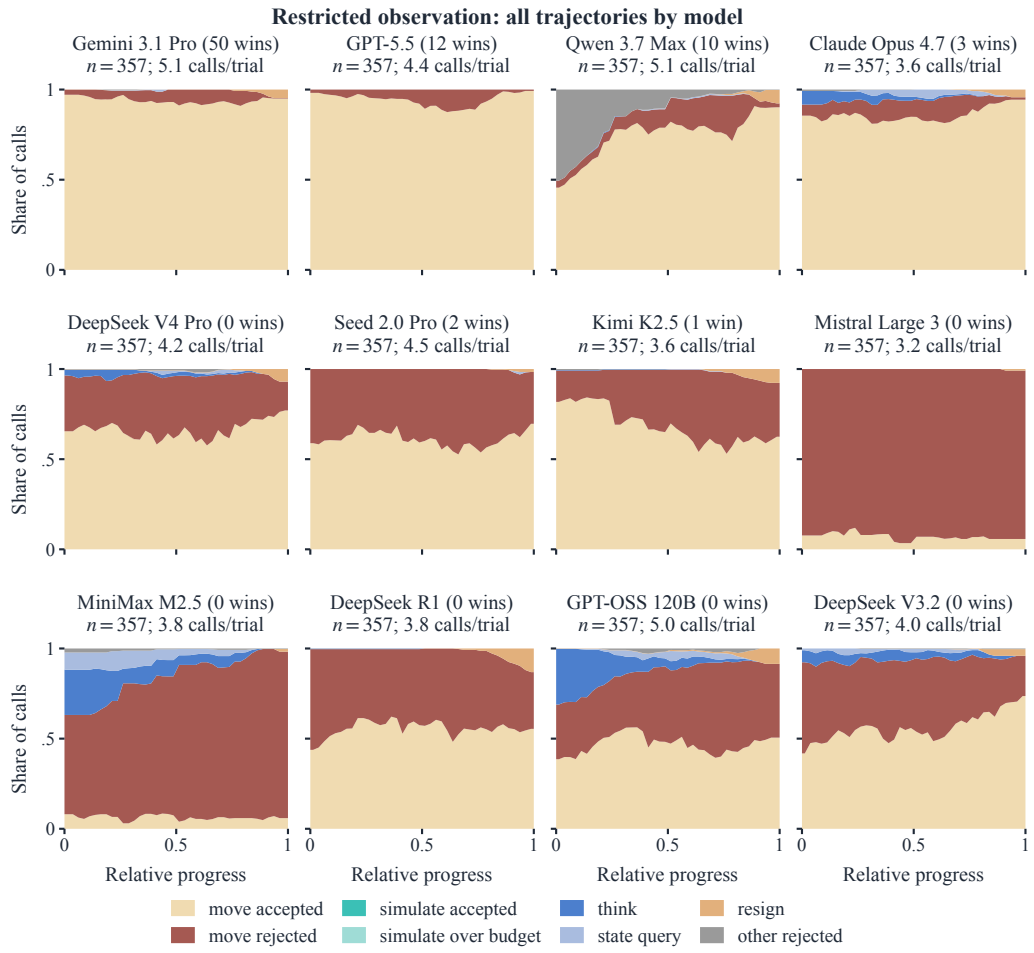


Figure 15: **Restricted-observation REPL-call composition for each LLM over all 357 trials.** Simulation, board views, and legal-move queries are unavailable in this setting, so its state queries are `get_history` calls.

G FEEDBACK-CONDITIONED BEHAVIOR AND EXPLORATORY GROUPING

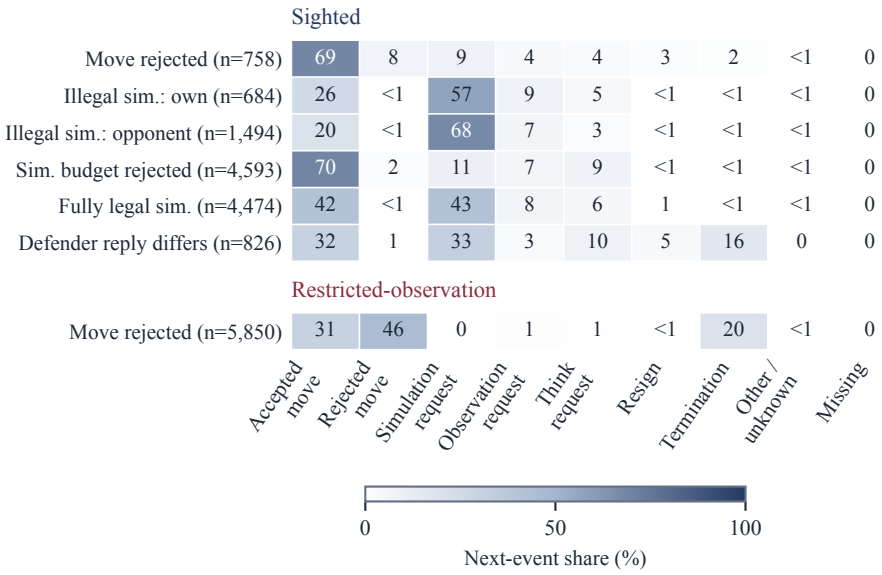


Figure 16: **Immediate behavior after selected feedback events.** Each row conditions on one feedback event and shows the next interaction in the same real decision turn; defender-reply mismatch instead begins at the next agent decision state after the real defense. Requests include rejected requests and do not imply successful execution. Counts retain termination, missing next events, and unknown actions; displayed percentages are rounded and may not sum to 100. Sighted trajectories contain no observed unknown or protocol-rejected simulation rows, and simulation is unavailable under Restricted observation.

We represent each non-synthetic interaction using observable events: state or history requests, candidate proposals, validation calls, feedback-conditioned candidate changes, real actions, and termination. The primary feature scope is the first agent decision turn; full-trajectory rates are used only as a sensitivity analysis because longer trajectories create more opportunities. For each rejected real move, recovery means that independent rules replay confirms a legal real move within the next one, two, or three recorded action interactions of the same turn and position. Read, simulation, and `think` interactions consume window positions; no-command responses are not represented as action interactions. Resignation and forced termination remain competing events in the denominator. A 1,000-resample case bootstrap retains all trials and overlapping error windows from each sampled case. The measure therefore combines correction behavior with the interface's retry and forfeit rules and does not score strategic move quality.

Among 758 Sighted real-move rejections, legal recovery occurs within one, two, and three subsequent action opportunities for 526 (69.4%), 630 (83.1%), and 666 (87.9%), respectively. The corresponding counts among 5,850 Restricted-observation rejections are 1,823 (31.2%), 2,260 (38.6%), and 2,273 (38.9%). At the three-opportunity window, case-bootstrap 95% intervals are [84.6, 90.9]% and [36.0, 42.1]%; each resample retains all rejection events from a sampled case. These are event-weighted totals, and models contribute unequal numbers of rejected moves: Mistral Large 3 and MiniMax M2.5 account for 1,052 and 1,049 of the Restricted-observation events and recover from 4.3% and 4.6% of them within three opportunities. Per model (Figure 17), recovery within three opportunities is 90.0% Sighted versus 95.8% Restricted for Gemini 3.1 Pro, 100% versus 93.3% for GPT-5.5, 100% versus 88.1% for Claude Opus 4.7, and 96.4% versus 91.0% for Qwen 3.7 Max, although these four models have only 11–28 Sighted events each. The setting-level difference is descriptive: a later legal action is recovery from an interface rejection, not a model-averaged capability, evidence that move quality improved, or a causal effect of the observation setting.

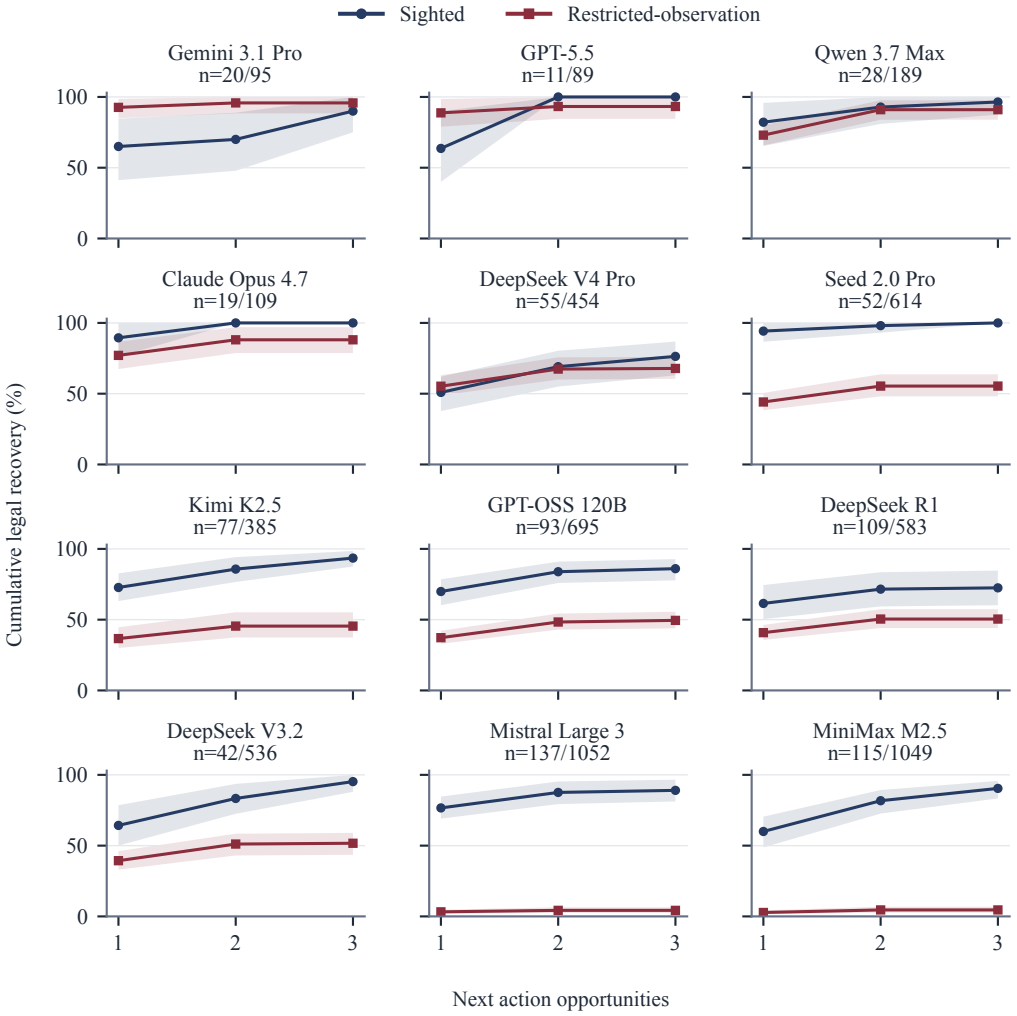


Figure 17: **Legal recovery after a rejected move, by model.** Each panel shows one model; curves give the cumulative fraction of its rejected-move events followed by at least one replay-confirmed legal real move within the next one, two, or three recorded action interactions of the same turn. Read and `think` interactions therefore consume window positions. Panel titles give n = Sighted/Restricted-observation rejected-move events (758 and 5,850 in total); forced termination and resignation remain competing events. Shading gives 95% intervals from 1,000 case-bootstrap resamples that retain all events from a sampled case. Overlapping error windows are not independent, and legal recovery does not imply strategic improvement.

Action Mix and First-Move Choices. Figure 5 in Section 6 reports each model's share of all actions by REPL command over complete trajectories. Simulation counts are broken down by model in Tables 4 and 5, and first-move choices by model and piece in Tables 8–11.

H Piece Composition and Action Selection

Presence groups describe overlapping initial material themes; combination groups are mutually exclusive. The figure summarizes outcomes, while the tables give case counts and first-move action composition.

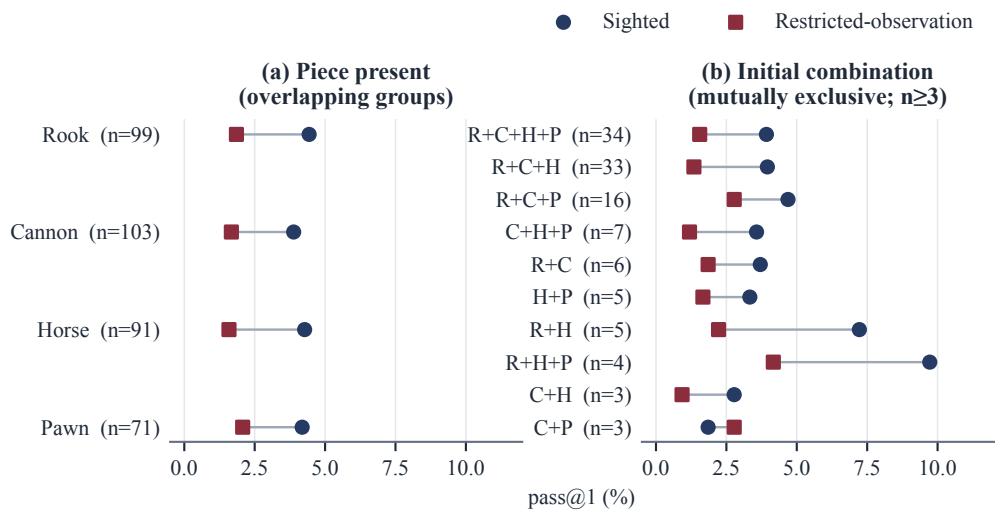


Figure 18: **Outcomes by initial attacking pieces.** Left: rook/cannon/horse/pawn presence groups, which overlap and cannot be summed; right: mutually exclusive combinations with at least three cases. Labels give case counts, and points pool 12 models and three trials per case under Sighted/Restricted-observation. Differences in position structure and small groups preclude causal piece-difficulty conclusions.

Table 12: Observed passes by initial red-piece presence. Rows overlap and must not be added. Position structure and difficulty confound comparisons; these are not causal piece effects.

Initial red piece	Cases	Sighted pass/traj.	Restricted-observation pass/traj.
Rook	99	158/3564	66/3564
Cannon	103	144/3708	62/3708
Horse	91	140/3276	52/3276
Pawn	71	107/2556	53/2556

Presence groups overlap, whereas combination groups are mutually exclusive. First-move counts are availability- and trajectory-weighted action composition, not preferences or piece-specific ability.

Table 13: Mutually exclusive initial red-piece presence combinations. Groups with fewer than five cases are retained for coverage, not model rankings.

Combination	Cases
rook+cannon+horse+pawn	34
rook+cannon+horse	33
rook+cannon+pawn	16
cannon+horse+pawn	7
rook+cannon	6
horse+pawn	5
rook+horse	5
rook+horse+pawn	4
cannon+horse	3
cannon+pawn	3
cannon	1
pawn	1
rook+pawn	1

Table 14: Piece type of the first accepted non-synthetic agent move, over 4,284 trajectories per setting. Other comprises general, advisor, and elephant; unknown denotes unparsed feedback. This is action composition, not position-independent preference.

First moved piece	Sighted	Restricted-observation
Rook	2032	1665
Cannon	705	588
Horse	851	608
Pawn	556	525
Other	31	88
Unknown	0	0
No Accepted First Move	109	810

I REPRESENTATIVE TRAJECTORIES

The overview trajectory. Figure 1 shows Sighted Gemini 3.1 Pro trial 3 on SQYQ-251; it is an illustrative trace, not a representative sample. One no-command response precedes the first explicit tool interaction. After the real prefix `h4g6 e7f7`, the second Red decision contains `think`, `simulate b8f8 h9f8 i9f9`, another `think`, and `move b8f8`. The figure reproduces the complete model response for that decision's pre-simulation `think` interaction, not tool output or all reasoning text across the trajectory. Its original `2.` and `2...` notation numbers full moves, whereas the figure's badges number individual plies 1–5. The model supplies both sides of the simulated continuation; all three plies are legal without changing the real board. The defender then actually plays `h9f8`; its move feedback and updated state precede the next `think` and `move i9f9`. Thus the simulated three-ply suffix matches the final three real plies. The archived result records a Red win labelled as stalemate; independent rules replay confirms checkmate. Unlike the Qwen trial 2 discussed below, which simulates all five plies at the opening decision and has three earlier no-command interactions, this Gemini trace demonstrates simulation after an observed defender reply. Neither case establishes a causal tool benefit or a strategy robust to untested replies.

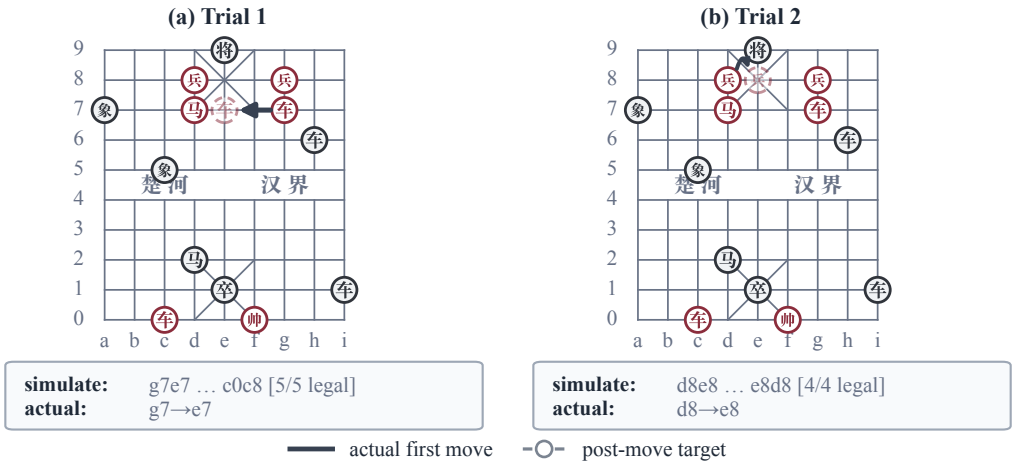


Figure 19: **Same model and task, different candidate sequences.** Two Sighted Gemini 3.1 Pro trials on *Shi Qing Ya Qu* case SQYQ-012 start from the displayed initial FEN. Trial 1 simulates a legal five-ply line, adopts rook `g7e7`, and passes; trial 2 simulates a legal four-ply line, adopts pawn `d8e8`, and fails by resignation. Solid arrows are actual first moves; pale dashed outlines mark their destinations, not separate simulated moves. The contrast is observational and does not identify a causal effect of either initial move.

On SQYQ-012, trial 1 submits the legal line `g7e7 c5e7 d8e8 e9e8 c0c8`, executes `g7e7`, and later records a win through `d8e8` and `d7b8`. Trial 2 submits the legal line `d8e8 e9e8 g8f8 e8d8`, executes `d8e8`, and eventually resigns. The example separates candidate legality, real execution, and terminal outcome; it does not show that the initial divergence caused the failure.

These examples are chosen by fixed, deterministic selection rules applied to the retained trajectories. They illustrate observable action differences without treating selected traces as estimates of a general model tendency.

Verified plan-to-execution example. In Sighted trial 2 of Qwen 3.7 Max on SQYQ-251, the model explicitly described a plan to force the Black general to f7 and then combine a rook sacrifice with the cannon; after three earlier no-command interactions, it stated that it would simulate the line before committing. The submitted line `h4g6 e7f7 b8f8 h9f8 i9f9` executed all five requested plies legally on the board copy. The first five alternating plies on the real board then matched this line exactly. An independent rules replay confirms that each Red move gives check, that `e7f7` and `h9f8` are the sole legal Black replies at their respective positions, and that the final position is checkmate, although the archived terminal label says stalemate. The model's written plan is evidence of an expressed plan, not direct access to an internal reasoning mechanism.

Table 15: **Same-case trajectory contrasts selected by fixed rules.** C1, largest same-case success spread between models; C2, same model and first move, different outcomes; C3, Restricted-observation invalid-action losses versus a peer; C4, explicit first-turn tool use versus none; C5, rejected moves recovered within the turn versus none rejected; C6, different first-moved piece, different outcomes. Trial k is the k-th of the three trials in a model–case–setting cell. Root move is the stored reference first move; Played lists up to four agent moves. B/L count successful/attempted explicit board and legal-move calls; S counts accepted/attempted simulation calls, not fully legal lines. Terminal reasons follow the rules replay of Appendix A.5. These purposefully selected examples do not estimate representative effects or internal reasoning.

Contrast; model; setting	Root move; agent moves	Outcome; actions
C1a JH-084 Gemini 3.1 Pro, trial 1 Sighted	Root move: g5g9. Played: g5g9, h5h9	Pass; checkmate. Invalid: 0. B/L/S: 0/0, 0/0, 0/0.
C1b JH-084 GPT-OSS 120B, trial 1 Sighted	Root move: g5g9. Played: g5d5, e0d0, d0e0, e0d0, ...	Fail; resignation. Invalid: 3. B/L/S: 1/1, 1/1, 0/0.
C2a SQYQ-066 Gemini 3.1 Pro, trial 2 Sighted	Root move: g8g9. Played: g8g9, d8e8, g9g8, i5g6, ...	Pass; checkmate. Invalid: 0. B/L/S: 0/0, 0/0, 4/7.
C2b SQYQ-066 Gemini 3.1 Pro, trial 1 Sighted	Root move: g8g9. Played: g8g9, d8d9, g9g8, i5g6, ...	Fail; attacker checkmated. Invalid: 0. B/L/S: 0/0, 0/0, 8/18.
C3a SQYQ-062 DeepSeek R1, trial 2 Restricted observation	Root move: f8e8. Played: a5e5, e5e8, b8c6	Fail; invalid-action limit. Invalid: 5. B/L/S: 0/0, 0/0, 0/0.
C3b SQYQ-062 Gemini 3.1 Pro, trial 1 Restricted observation	Root move: f8e8. Played: c8c9, a5e5, e2g0	Fail; attacker checkmated. Invalid: 1. B/L/S: 0/0, 0/0, 0/0.
C4a JH-327 DeepSeek V4 Pro, trial 3 Sighted	Root move: h2h9. Played: h2h9, i2d2, d2d4, d4d5	Pass; checkmate. Invalid: 0. B/L/S: 1/1, 1/1, 2/3.
C4b JH-327 Kimi K2.5, trial 2 Sighted	Root move: h2h9. Played: h2h9, i2i9	Fail; attacker checkmated. Invalid: 0. B/L/S: 0/0, 0/0, 0/0.
C5a SQYQ-098 Qwen 3.7 Max, trial 2 Restricted observation	Root move: b2b9. Played: b2b9, g7g8, d0d1, d1d2, ...	Pass; defender perpetual check. Invalid: 5. B/L/S: 0/0, 0/0, 0/0.
C5b SQYQ-098 Gemini 3.1 Pro, trial 2 Restricted observation	Root move: b2b9. Played: b2b9, h6h9, g7g8, h9f9, ...	Pass; checkmate. Invalid: 0. B/L/S: 0/0, 0/0, 0/0.
C6a SQYQ-012 Gemini 3.1 Pro, trial 1 Sighted	Root move: g7e7. Played: g7e7, d8e8, d7b8	Pass; checkmate. Invalid: 0. B/L/S: 0/0, 0/0, 1/1.
C6b SQYQ-012 Gemini 3.1 Pro, trial 2 Sighted	Root move: g7e7. Played: d8e8, d7c9, g8f8, f8e8, ...	Fail; resignation. Invalid: 1. B/L/S: 0/0, 0/0, 6/10.

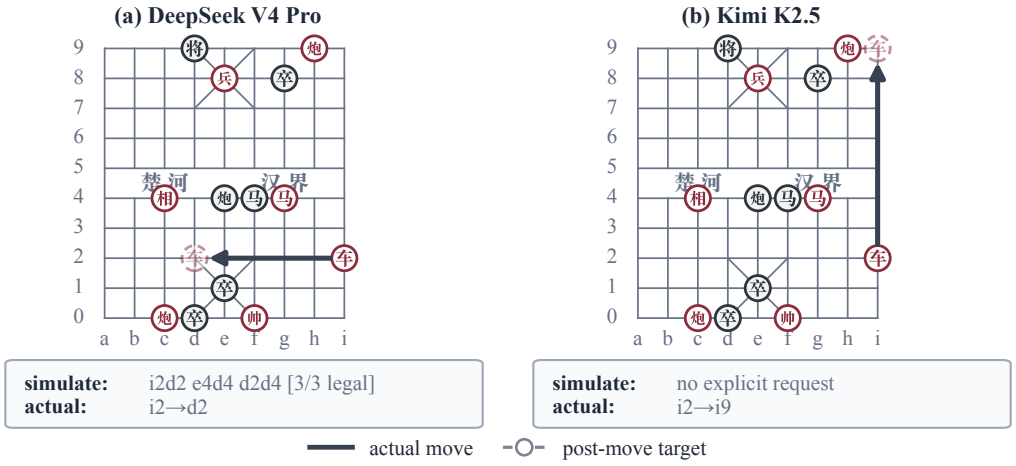


Figure 20: **Cross-model action contrast on Jianghu case JH-327.** After the shared Sighted real-game prefix `h2h9, g9g8`, both panels show the same second Red decision state. DeepSeek V4 Pro trial 3 submits a 3/3-legal simulation and plays `i2d2` (pass); Kimi K2.5 trial 2 makes no explicit simulation and plays `i2i9` (fail). Solid arrows are the actual moves; pale dashed piece outlines mark each move's post-move target and are not current pieces or additional simulation arrows. Simulation details appear in the panel data lines. Saved outcomes are observational and do not identify a tool effect.

Tactical sequence and same-case contrast. After `h4g6`, the Red horse on g6 controls e7 and protects f8, so the Black general on f7 cannot capture a rook on f8. The checking sacrifice `b8f8` forces the sole reply `h9f8`: the Black horse captures the rook, opens the cannon's path from i9 to f9, and itself becomes the screen for `i9f9` checkmate. In a Sighted DeepSeek V4 Pro trial on the same case, the real prefix also begins `h4g6 e7f7`, but the next move is `b8e8`. That move is legal but does not give check, despite the model's textual claim that the pieces were on the same rank and the move checked; after Black plays `h7h0`, Red is checkmated. This comparison identifies an observable trajectory divergence and a rule-checkable textual error. It does not establish that `b8e8` is an engine-determined loss point, that every move in the failed trajectory is erroneous, or that simulation caused the successful outcome.

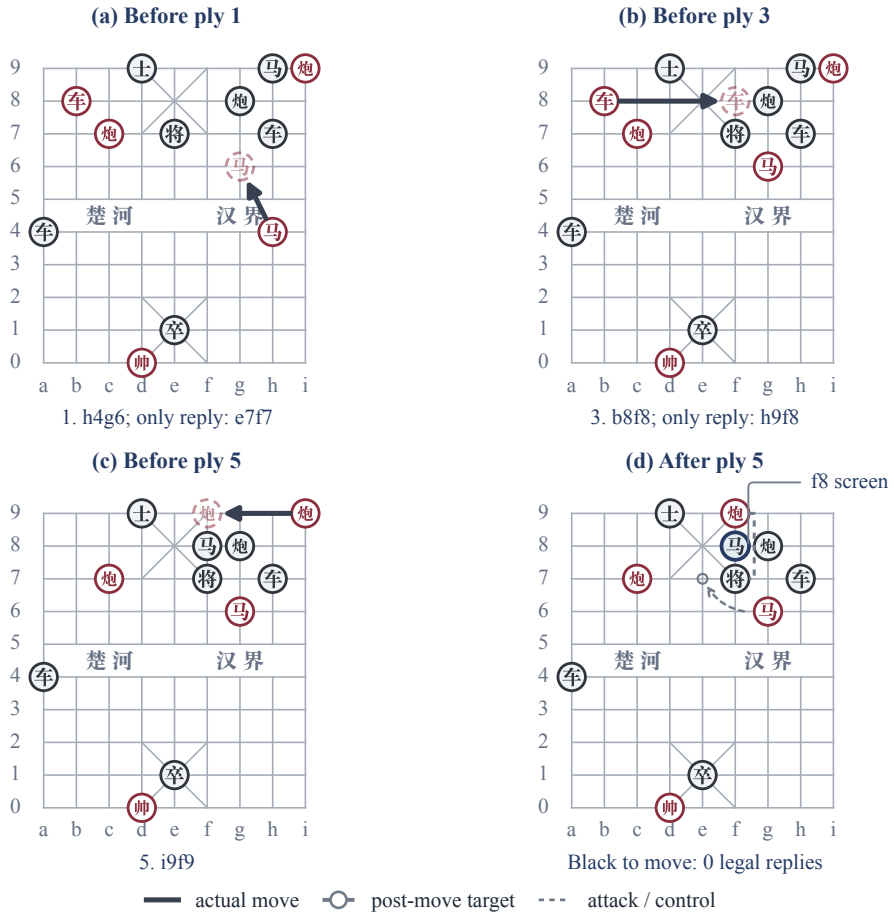


Figure 21: **Verified tactical sequence on SQYQ-251.** Panels A–C show the real position immediately before plies 1, 3, and 5 (Red's three decisions); each solid dark arrow is the actual move, and the pale dashed target marks that move's destination rather than another piece. Independent rules replay finds that `e7f7` is Black's sole legal reply after `h4g6`, and `h9f8` the sole reply after `b8f8`. Panel D shows the real position after five plies: the cannon on f9 checks the Black general on f7 through the Black horse on f8, while the Red horse on g6 controls e7; Black has no legal move. Thin gray dashed marks in D denote these attack or control relations. The f8 emphasis is explanatory, not an interface signal. The archived feedback reported the mating move as non-checking and labelled the terminal as stalemate; the independent replay identifies checkmate.

J Model Coverage and API Configuration

The evaluation contains 12 LLMs with three scored trials per case in both observation settings (Table 16). Model names follow those reported by API providers; underlying deployment versions and service identity were not independently verified. Three additional models were attempted but lack complete coverage: Kimi K2.6 had incomplete collection after credential failure, GLM 5.1 repeatedly exceeded the request timeout, and Seed 2.1 Pro produced no scored record before timeout. These observations concern collection infrastructure and are not game losses. They also illustrate why the main estimates are conditional on a complete recorded matrix.

Table 16: **Evaluated models.** Model IDs are the names under which trajectories are stored, and *Weights* indicates whether model weights are publicly released. Each model contributes 357 scored trajectories per observation setting.

Model	Developer	Weights	Model ID
Gemini 3.1 Pro	Google	Closed	`gemini-3.1-pro`
GPT-5.5	OpenAI	Closed	`gpt-5.5`
Qwen 3.7 Max	Alibaba	Closed	`qwen3.7-max`
Claude Opus 4.7	Anthropic	Closed	`claude-opus-4.7`
DeepSeek V4 Pro	DeepSeek	Open	`deepseek-v4-pro`
Seed 2.0 Pro	ByteDance	Closed	`seed-2.0-pro`
Kimi K2.5	Moonshot AI	Open	`kimi-k2.5`
Mistral Large 3	Mistral AI	Open	`mistral-large-3`
MiniMax M2.5	MiniMax	Open	`minimax-m2.5`
DeepSeek R1	DeepSeek	Open	`deepseek-r1`
GPT-OSS 120B	OpenAI	Open	`gpt-oss-120b`
DeepSeek V3.2	DeepSeek	Open	`deepseek-v3.2`

K Complete Results

The tables report pass@1, pass@3, and pass^3 for both settings, followed by the full finite-sample pass@k and pass^k values for $k = 1, 2, 3$.

Table 17: **Three-trial success and consistency (%).** Pass@3 is the fraction of cases with at least one success; pass^3 is the fraction for which all three trials succeed. S/R denote Sighted/Restricted observation.

	Pass@1		Pass@3		pass^3	
Model	S	R	S	R	S	R
Gemini 3.1 Pro	22.1	14.0	38.7	32.8	5.9	1.7
GPT-5.5	9.0	3.4	17.6	6.7	2.5	0.8
Qwen 3.7 Max	6.2	2.8	13.4	5.9	0.8	0.8
Claude Opus 4.7	5.3	0.8	10.9	1.7	0.8	0.0
DeepSeek V4 Pro	2.8	0.0	5.0	0.0	1.7	0.0
Seed 2.0 Pro	0.8	0.6	2.5	1.7	0.0	0.0
Kimi K2.5	0.8	0.3	1.7	0.8	0.0	0.0
Mistral Large 3	0.8	0.0	0.8	0.0	0.8	0.0
MiniMax M2.5	0.8	0.0	0.8	0.0	0.8	0.0
DeepSeek R1	0.6	0.0	1.7	0.0	0.0	0.0
GPT-OSS 120B	0.3	0.0	0.8	0.0	0.0	0.0
DeepSeek V3.2	0.3	0.0	0.8	0.0	0.0	0.0

Table 18: **Full finite-sample consistency metrics (%).** For each case with $n = 3$ trials and c successes, Pass@$k = 1 - \binom{3-c}{k}/\binom{3}{k}$ and pass^k $= \binom{c}{k}/\binom{3}{k}$; entries average the case values. No metric with $k > 3$ is estimated.

		Pass@k			pass^k		
Model	Prot.	$k=1$	$k=2$	$k=3$	$k=1$	$k=2$	$k=3$
Gemini 3.1 Pro	S	22.1	33.1	38.7	22.1	11.2	5.9
Gemini 3.1 Pro	R	14.0	24.4	32.8	14.0	3.6	1.7
GPT-5.5	S	9.0	14.0	17.6	9.0	3.9	2.5
GPT-5.5	R	3.4	5.3	6.7	3.4	1.4	0.8
Qwen 3.7 Max	S	6.2	10.4	13.4	6.2	2.0	0.8
Qwen 3.7 Max	R	2.8	4.5	5.9	2.8	1.1	0.8
Claude Opus 4.7	S	5.3	8.7	10.9	5.3	2.0	0.8
Claude Opus 4.7	R	0.8	1.4	1.7	0.8	0.3	0.0
DeepSeek V4 Pro	S	2.8	3.9	5.0	2.8	1.7	1.7
DeepSeek V4 Pro	R	0.0	0.0	0.0	0.0	0.0	0.0
Seed 2.0 Pro	S	0.8	1.7	2.5	0.8	0.0	0.0
Seed 2.0 Pro	R	0.6	1.1	1.7	0.6	0.0	0.0
Kimi K2.5	S	0.8	1.4	1.7	0.8	0.3	0.0
Kimi K2.5	R	0.3	0.6	0.8	0.3	0.0	0.0
Mistral Large 3	S	0.8	0.8	0.8	0.8	0.8	0.8
Mistral Large 3	R	0.0	0.0	0.0	0.0	0.0	0.0
MiniMax M2.5	S	0.8	0.8	0.8	0.8	0.8	0.8
MiniMax M2.5	R	0.0	0.0	0.0	0.0	0.0	0.0
DeepSeek R1	S	0.6	1.1	1.7	0.6	0.0	0.0
DeepSeek R1	R	0.0	0.0	0.0	0.0	0.0	0.0
GPT-OSS 120B	S	0.3	0.6	0.8	0.3	0.0	0.0
GPT-OSS 120B	R	0.0	0.0	0.0	0.0	0.0	0.0
DeepSeek V3.2	S	0.3	0.6	0.8	0.3	0.0	0.0
DeepSeek V3.2	R	0.0	0.0	0.0	0.0	0.0	0.0

L Additional Failure Statistics

The invalid/no-command category includes only those explicit terminal reasons; its complement spans multiple outcomes and is not a strategic-loss label. Qwen 3.7 Max combines more Sighted wins than most models with a comparatively high Sighted invalid/no-command loss rate (25.5%), while several models have few such forfeits but also few or no wins.

Table 19: Invalid/no-command loss rate (%) by observation setting. Check feedback is affected as described in Appendix A.5. Records classified as aborted or unfinished are excluded before selecting three scored records per cell; each denominator is 357 scored trials.

Model	Sighted	Restricted-observation
Gemini 3.1 Pro	2.2	0.3
GPT-5.5	0.0	0.6
Qwen 3.7 Max	25.5	3.4
Claude Opus 4.7	0.0	1.1
DeepSeek V4 Pro	0.0	13.2
Seed 2.0 Pro	0.0	25.2
Kimi K2.5	0.0	19.6
Mistral Large 3	1.1	93.8
MiniMax M2.5	0.8	93.8
DeepSeek R1	2.2	25.2
GPT-OSS 120B	0.8	31.9
DeepSeek V3.2	0.0	23.5

M Repeated-Trial Profiles

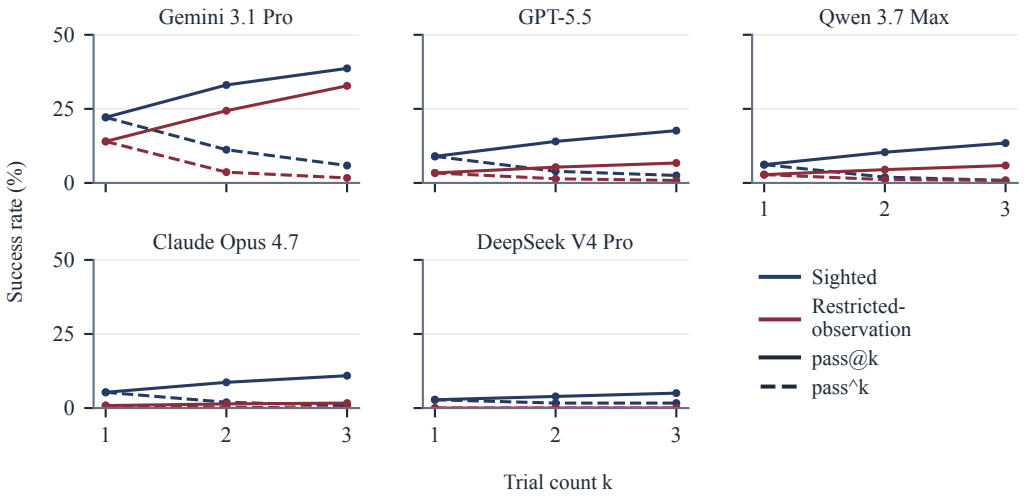


Figure 22: **At-least-once and all-trials success diverge as k grows.** Panels show the five models with the highest observed Sighted pass@1. Colors identify Sighted and Restricted observation; solid and dashed lines show pass@k and pass^k for $k = 1, 2, 3$. Values are finite-sample case-level estimates from the same three saved trials. Their monotonicity follows from the definitions and is not evidence of a compute-scaling law.

For Gemini 3.1 Pro in Sighted, pass@k rises from 22.1% at $k = 1$ to 33.1% at $k = 2$ and 38.7% at $k = 3$, whereas pass^3 is 5.9%. These three archived attempts do not support extrapolation to larger trial budgets.

N Evaluation Set Size and Estimate Stability

We vary the number of evaluated cases while retaining the same three trials per case for all 12 models. Figure 23 uses nested subsets of the fixed 119-case corpus, so changes reflect case composition rather than training, additional trials per case, or changes in model ability. A separate analysis uses 1,000 common random permutations to quantify sensitivity to subset order; the resulting quantile ranges describe variation within this finite corpus, not confidence intervals for future performance.

For Gemini 3.1 Pro in Sighted, the 2.5th–97.5th percentile range of pass@1 across these permutations narrows from 3.3–43.3% at $m = 10$ to 19.7–24.3% at $m = 100$, compared with 22.1% on all 119 cases. Rare all-three successes are particularly sensitive to case selection: at $m = 10$, pass^3 is zero in 50.6% of the Gemini Sighted subsets and 75.5% of the GPT-5.5 Sighted subsets. These observations reflect sparse successful cases, not an improvement in ability as the evaluation set grows. At $m = 119$, all permutations contain the same corpus and reproduce the main table exactly; collapse of subset variation at that point does not establish certainty about new cases.

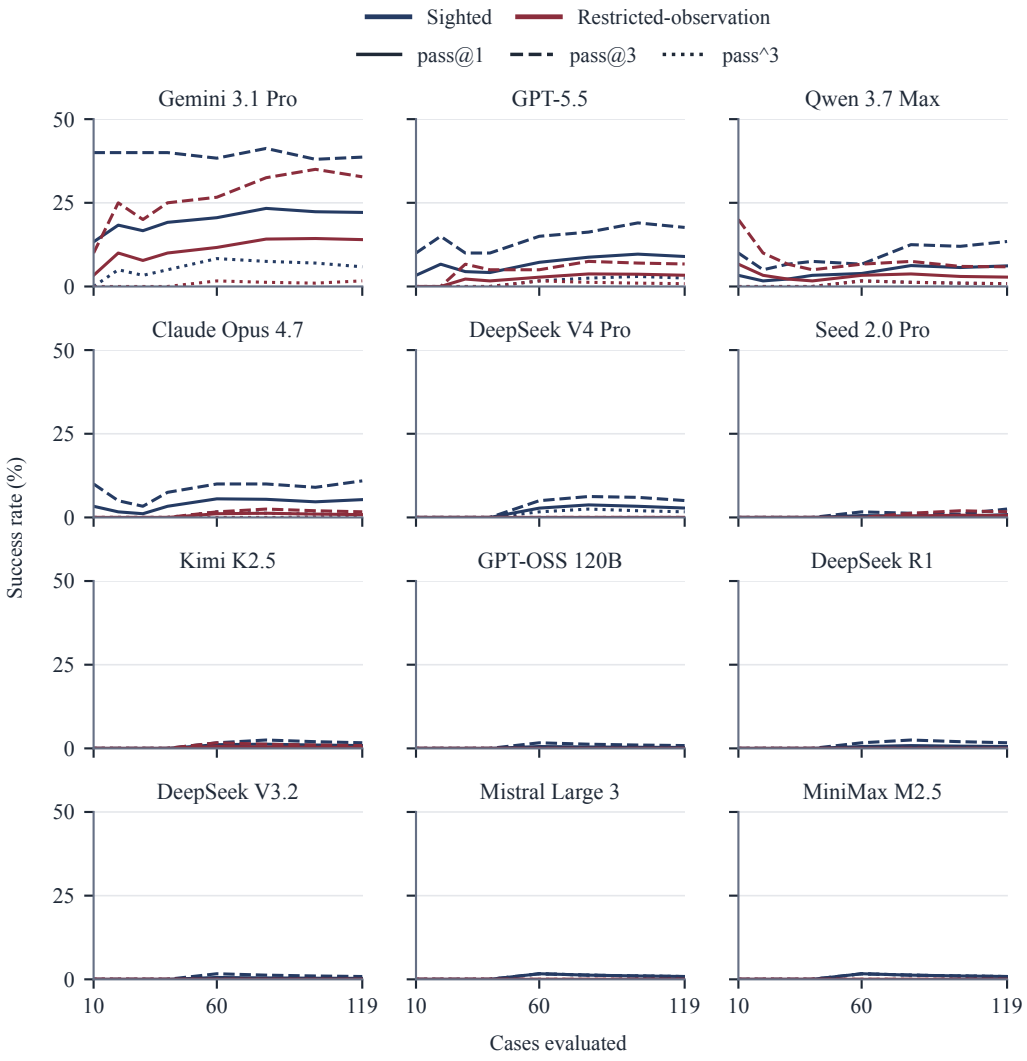


Figure 23: **Estimate variation across nested evaluation subsets.** Each panel shows one model on common nested case subsets of sizes $m = 10, 20, 30, 40, 60, 80, 100, 119$, with three fixed trials per case. All models and both settings share one fixed random case ordering. Colors indicate Sighted/Restricted-observation; solid, dashed, and dotted lines show pass@1, pass@3, and pass^3. This is not a training curve or an increase in trial count k; scores need not be monotonic. Full-set scores reproduce the main table. The figure shows one ordering, not the quantile ranges from the separate 1,000-permutation analysis.