
Atomic Chess Reveals Compositional Reasoning Failures in LLMs

Ryan Co^{*1} Karthik Konuganti^{*1}

Abstract

Large language models are capable of playing chess at a notable level, but it is unclear whether this reflects reasoning or memorized priors. We study whether models can compose an explicit rule change using atomic chess, a variant that preserves the board and piece notation while altering the rules: every capture explodes adjacent pieces, turning many standard tactics into mistakes. We construct a diagnostic paired evaluation of 200 variant-divergent positions drawn from standard and atomic chess games. Every position is selected so that the best move under atomic rules differs from the best move under standard rules, and the standard-best move is a blunder under atomic rules. Across Claude Opus 4.6 and GPT-5.4, atomic Win% loss exceeds standard by up to $4.6\times$ on identical positions. Increasing reasoning compute reduces atomic mean Win% loss, but does not eliminate the gap. Qualitative analysis of reasoning traces reveals failure patterns including board representation and composition, most notably *unpropagated refutation*, where a model recognizes that a candidate move is bad under atomic rules but selects it anyway. These results suggest that reasoning LLMs can partially use explicit compositional rules, but sometimes fail to override familiar action priors.

1. Introduction

Chess has served as a benchmark for machine intelligence since the earliest days of AI, from Shannon’s and Turing’s chess programs through Deep Blue to AlphaZero, valued for its precisely defined rules and objectively measurable play. LLMs continue this tradition with notable results: searchless transformers reach Lichess blitz Elo 2895 (Ruoss et al., 2024), and gpt-3.5-turbo-instruct plays around Elo 1800 (Karvonen, 2024a). Yet this competence is brittle.

^{*}Equal contribution ¹Georgia Institute of Technology. Correspondence to: Ryan Co <rc03@gatech.edu>.

Accepted to the *2nd Workshop on Compositional Learning at ICML 2026*, Seoul, South Korea. Copyright 2026 by the author(s).

Frontier models often fail to complete games against random opponents (Kolasani et al., 2025), and simple board rotations inflate GPT-5.1’s centipawn error by over 600% (Song et al., 2025). It is therefore unclear whether LLM chess play reflects memorization, learned tactical priors, or reasoning.

We ask a question these benchmarks cannot answer directly: *can an LLM apply an explicitly stated rule change to a familiar domain when that change invalidates its learned priors?* Existing counterfactual tests perturb geometry such as swapping piece starting squares (Wu et al., 2024) or shuffled starting positions (Mészáros et al., 2025) but preserve the capture rule, the mechanic that determines move value.

Atomic chess as a probe. In atomic chess, every capture causes an explosion that removes the captured piece, the capturing piece, and all non-pawn pieces on the eight adjacent squares. Kings cannot capture, captures that would explode one’s own king are illegal, and exploding the opponent’s king wins (Lichess, n.d.a). Three properties make this a useful compositional probe:

1. **State invariance.** FEN, UCI, and PGN encodings are identical to standard chess, so gaps cannot be attributed to a new input language.
2. **Tactical inversion.** Captures that are strong in standard chess are often catastrophic in atomic, so pattern-matching to standard tactics produces systematic errors.
3. **Lower contamination risk.** There are far fewer documented Atomic chess games, reducing memorization risk.

Formal setup. We model a chess game as a tuple $\mathcal{C} = (\mathcal{S}, \mathcal{M}, \mathcal{T}, \mathcal{G})$: state space, legal-move generator, capture-resolved transition function, and terminal check, with induced optimal policy π^* and value function V^* . Atomic chess preserves \mathcal{S} and the notation but modifies the rest:

$$\mathcal{C}_S = (\mathcal{S}, \mathcal{M}_S, \mathcal{T}_S, \mathcal{G}_S), \quad \mathcal{C}_A = (\mathcal{S}, \mathcal{M}_A, \mathcal{T}_A, \mathcal{G}_A), \quad (1)$$

yielding π_S^*, π_A^* and V_S^*, V_A^* . Note that $\mathcal{M}_A \neq \mathcal{M}_S$: king captures and self-exploding captures are disallowed, and check constraints differ. Success requires *rule composition*: standard chess knowledge remains useful for representing pieces, attacks, and candidate moves, but move values must be recomputed under atomic semantics. The task is not to

discard standard knowledge but to determine which inherited concepts remain valid. Following Fodor and Pylyshyn (1988) and Lake and Baroni (2018), we distinguish *systematicity*, applying a stated rule to new inputs, from *productivity*, integrating the rule with existing knowledge into globally consistent behavior.

Contributions. (1) A paired benchmark of 200 *variant-divergent* positions from Lichess games (100 standard-sourced, 100 atomic-sourced), where the atomic-best move differs from the standard-best move and the standard-best move is harmful under atomic rules; each FEN is evaluated under both rulesets, yielding an atomic-rule gap on identical board states. This filter makes the benchmark diagnostic for memorization versus rule composition. (2) A within-family reasoning-effort ablation. (3) A no-rules-provided ablation. (4) An unfiltered atomic-source check showing the gap shrinks and becomes model-dependent without the filter, validating the contrastive design. (5) A qualitative analysis of high-reasoning traces, organized by the decomposition $(S, \mathcal{M}_A, \mathcal{T}_A, \mathcal{G}_A)$, identifying recurring failure patterns without treating trace categories as prevalence estimates.

2. Background

LLMs and chess. Small transformers trained on chess corpora learn legality, openings, and tactics (Noever et al., 2020; DeLeo and Guven, 2022; Stöckl, 2021; Toshniwal et al., 2022), with later models reaching strong play (Feng et al., 2023; Monroe and Chalmers, 2024; Zhang et al., 2025). Ruoss et al. (Ruoss et al., 2024) frame chess as a domain where memorization is futile, since evaluation positions diverge substantially from any plausible training set; Karvonen (Karvonen, 2024b) recovers board state via linear probes. For general LLMs, no model defeats Maia-1100 (Liu et al., 2025), structural and tactical weaknesses persist (Wen et al., 2025), and reasoning models still fail at agentic play (Kolasani et al., 2025).

Compositional generalization. The SCAN/COGS lineage (Lake and Baroni, 2018; Kim and Linzen, 2020) establishes that systematic compositional generalization is hard. Atomic chess is a productivity test: the rule package is small and stated, but applying it requires recombining it with existing chess knowledge. Dziri et al. (Dziri et al., 2023) argue transformer reasoning reduces to “linearized subgraph matching”, which is the failure mode atomic chess is designed to expose.

Memorization vs. reasoning. Carlini et al. (Carlini et al., 2021; 2023) formalize extractable memorization; Wu et al. (Wu et al., 2024) introduce counterfactual tasks; McCoy et al. (McCoy et al., 2024) predict low-frequency variants degrade. Reasoning-enhanced models (OpenAI, 2024;

DeepSeek-AI et al., 2025; Wei et al., 2022) motivate our within-family reasoning ablation.

Game benchmarks. GTBench (Duan et al., 2024), ZeroSumEval (Khan et al., 2025), and ChatArena (Wu et al., 2023) pose inter-model competition (Zhang et al., 2024). We instead hold the game and notation fixed and vary only the capture rule.

3. Experimental Design

Position pool. We randomly sample 200 board positions from Lichess games: 100 positions from standard chess and 100 from atomic chess. Every position is evaluated under both standard and atomic rules. We use state-of-the-art engines Stockfish 16 (standard) and Fairy Stockfish (atomic) at depth 24 to evaluate the best move in each position. We report move quality using centipawn loss, defined as the difference, in hundredths of a pawn, between the engine evaluation of the best move and the evaluation after the chosen move. Lower loss indicates stronger play. Each query is a one-move planning problem. Example positions are provided in Appendix D.

Variant-divergence filtering. The evaluation is a rule-composition test designed to evaluate whether models can selectively reuse familiar chess concepts while recomputing move value under the new atomic rules. Therefore, we retain only positions where the best moves under standard and atomic chess rules differ, and the standard-best is a mistake under atomic rules. We exclude forced checkmate positions, as short forcing lines are often easier to solve. The reported evaluation is therefore a rule-composition gap, not an estimate of average atomic-chess strength. Section 4.4 shows the signal vanishes without this filter, confirming the filter functions as a targeted detection method.

Models and prompts. We evaluate Claude Opus 4.6 and GPT-5.4, each at `reasoning_effort` \in {none, high}. The prompt names atomic chess and states the variant rules. Full prompt templates are provided in Appendix A.

Metrics. Centipawn loss is capped at ± 1000 following Kolasani et al. (2025), and BEST is the rate of selecting the top engine move. Following the Lichess accuracy convention (Lichess, n.d.b), we map centipawns c to win probability:

$$\text{Win}\%(c) = 50 + 50 \left(\frac{2}{1 + \exp(-0.00368208c)} - 1 \right). \quad (2)$$

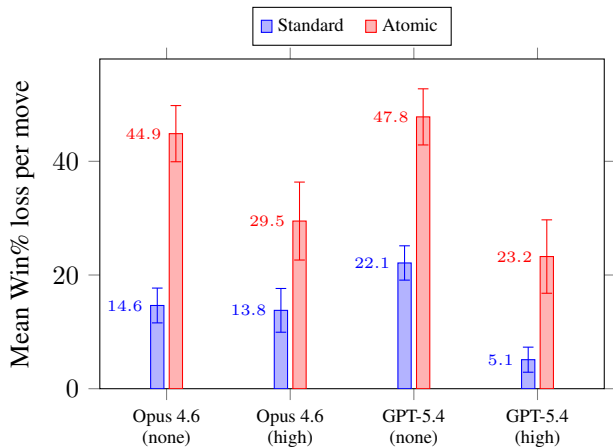


Figure 1. Mean Win% loss per move by ruleset. Numeric labels show mean loss; error bars show 95% bootstrap confidence intervals.

Letting $\Delta = \text{Win}\%_{\text{before}} - \text{Win}\%_{\text{after}}$:

$$\text{Move Quality} = \begin{cases} \text{BLUNDER} & \Delta \geq 30, \\ \text{MISTAKE} & \Delta \geq 20, \\ \text{INACCURACY} & \Delta \geq 10, \\ \text{GOOD} & \text{otherwise,} \end{cases} \quad (3)$$

applied in order from largest to smallest. For mean Win% loss, we report 95% confidence intervals computed by non-parametric bootstrap over positions. Because the evaluation is a diagnostic set of variant-divergent positions rather than a random sample from all chess positions, these intervals are descriptive uncertainty estimates rather than population-level estimates.

4. Results

We report mean Win% loss (ML) and best-move rate (BEST) on (i) the $n=200$ variant-divergent set, (ii) a within-family reasoning ablation, (iii) a no-rules ablation, and (iv) an unfiltered atomic-source check, followed by (v) a qualitative analysis of reasoning traces.

Table 1 reports categorical diagnostics. The increase in mean Win% loss is accompanied by substantially higher blunder rates under atomic rules, indicating that the rule-composition gap reflects severe move-selection failures rather than only small deviations from the engine-based oracle.

4.1. Rule-composition gap

On variant-divergent positions, atomic mean loss exceeds standard mean loss for all four configurations (Figure 1), increasing by $2.1\times$ for Opus high to $4.6\times$ for GPT-5.4. The gap is driven by the worst moves: switching from standard

Table 1. Move quality under standard and atomic chess rules on the same $n=200$ variant-divergent positions. Error categories use the Win% drop in Equation (3).

Rules	Model	BLUNDER	MISTAKE	INACC.	BEST
Standard	Opus 4.6 (none)	21.5%	9.0%	16.5%	19.5%
	GPT-5.4 (none)	36.5%	8.5%	12.0%	13.5%
	Opus 4.6 (high)	17.0%	10.0%	9.0%	29.0%
	GPT-5.4 (high)	4.0%	4.0%	5.0%	42.0%
Atomic	Opus 4.6 (none)	63.5%	5.5%	1.0%	13.5%
	GPT-5.4 (none)	68.5%	3.5%	1.5%	11.5%
	Opus 4.6 (high)	43.0%	3.0%	4.0%	35.0%
	GPT-5.4 (high)	30.0%	6.0%	5.0%	38.0%

to atomic rules on the same FENs raises BLUNDER rates from 17.0% \rightarrow 43.0% for Opus high and 4.0% \rightarrow 30.0% for GPT-5.4 high (Table 1). INACCURACY rates drop under atomic rules, so probability mass shifts toward lower quality moves.

4.2. Effect of reasoning effort

Increasing reasoning_effort from none to high reduces atomic mean loss by 34% for Opus 4.6 and 51% for GPT-5.4, and reduces atomic BLUNDER rates by roughly one-third for Opus 4.6 and more than half for GPT-5.4. Atomic BEST roughly triples. High reasoning reduces blunders while increasing top-engine moves rather than smaller errors.

The gap nevertheless persists. At high reasoning, atomic mean loss per move still exceeds standard by 15.7 (Opus) and 18.1 (GPT-5.4) Win% points on the same FENs, and atomic BLUNDER remains $2.5\text{--}7.5\times$ the standard rate. High-reasoning runs spend $1.8\text{--}2.3\times$ more output tokens on atomic than on standard prompts (Appendix B), so the residual gap is not attributable to an insufficient thinking budget.

4.3. No-rules ablation

Removing the rules of atomic chess (Table 2) collapses BEST and raises BLUNDER rates for low and high reasoning models. Naming the variant alone is insufficient when standard-chess priors are misleading; explicit rule text drives most of the partial success in Section 4.1. Notably, the cross-family gap closes under no-rules prompting (Opus high vs. GPT-5.4 high: 1.23% in mean loss, vs. 6.24% with rules supplied), suggesting the family separation depends on rule-conditioning rather than baseline atomic skill.

4.4. Unfiltered position check

Without the variant-divergent filter (Table 3), the atomic-standard difference becomes model-dependent. GPT-5.4 high retains a $2.9\times$ gap in mean loss per move, while Opus high shows no atomic degradation (mean loss 11.2 atomic vs. 14.0 standard). The unfiltered positions include many cases where the same move is strong in both variants, so stan-

Table 2. No-rules-provided ablation on the same $n=200$ source-balanced variant-divergent positions. The prompt names atomic chess but omits the rule statement. ML = mean Win% loss.

Model	BLUNDER	BEST	ML
Opus 4.6 (none)	76.0%	6.0%	52.6
GPT-5.4 (none)	76.0%	6.0%	52.2
Opus 4.6 (high)	46.0%	12.0%	32.0
GPT-5.4 (high)	42.0%	16.0%	30.8

Table 3. Unfiltered evaluation on $n=100$ positions sampled from atomic Lichess games without the variant-divergent filter or short forcing-sequence exclusion. Each position is evaluated under both atomic and standard rules. ML = mean Win% loss.

Model	Atomic rules			Standard rules		
	BLUNDER	BEST	ML	BLUNDER	BEST	ML
Opus 4.6 (high)	20.0%	24.0%	11.2	20.0%	26.0%	14.0
GPT-5.4 (high)	14.0%	36.0%	11.2	4.0%	58.0%	3.8

standard priors dominate. This shows that the variant-divergent set is a targeted test of rule composition under conflict, not an estimate of average atomic play.

4.5. Reasoning Trace Analysis

We inspect visible high-reasoning atomic-rules traces from Opus and GPT-5.4 to identify recurring behavioral patterns that help interpret the quantitative atomic performance gap. This analysis is qualitative and illustrative: visible traces are model outputs rather than guaranteed-faithful internal computations, so we use them to characterize plausible behavioral patterns rather than to estimate mechanism prevalence.

We organize the observations by which component of $\mathcal{C}_A = (\mathcal{S}, \mathcal{M}_A, \mathcal{T}_A, \mathcal{G}_A)$ appears implicated. O1–O2 are representation-level observations; O3–O5 are composition-level observations, where the trace appears to encode enough local information for atomic reasoning but fails to integrate that information into final action selection.

O1: Rule instability ($\mathcal{T}_A, \mathcal{M}_A$). Some traces state inconsistent versions of the atomic rule. For example, GPT-5.4 claims that “*pawns aren’t immune either!*” to explosion, even though they are immune under atomic chess rules. Such traces suggest that the model’s representation of the rule transition is unstable.

O2: Board state drift (\mathcal{S}). Some traces reason from an incorrect state. Opus claims “*I have a queen on e7, a knight on a3,*” when the queen is actually on h4 and the knights on c3 and f3. The atomic rule may be applied coherently, but to the wrong state.

O3: Unpropagated refutation (local \mathcal{T}_A vs. π). In some traces, the model identifies a candidate move as bad but selects it anyway. In one example, Opus chooses Bb4+ and calls it winning, but also notices a key line is problematic: “*the black bishop on b4 would be removed! That’s bad for black,*” then later concludes “*the best approach is Bb4+*” and plays it. The local refutation does not propagate to action selection.

O4: Pattern Reliance (π_S prior dominates π_A). Some traces appear to prioritize familiar standard-chess tactics over atomic calculation. GPT-5.4 calls “*Using Nxh7 to explode h7*” “*a well-known tactic—textbook!*” despite it being a large atomic blunder.

O5: Forcing Moves (V_S^* used as V_A^*). Some traces underweight quiet atomic threats and prefer forcing captures. GPT-5.4 dismisses the best move $f6e4$ because “*e4 is empty in the position. So it’s just a move,*” then plays $b4c3$ which is a capture but also a blunder.

Interpretation. These observations suggest that the atomic-rule gap is not just a failure to recall or state the rules. In several cases, the model appears to represent the relevant atomic consequence locally, but fails to make that consequence control the final move choice. The clearest example is unpropagated refutation, where the trace identifies why a candidate move is bad under atomic rules, yet the selected action still follows that refuted line. We therefore interpret the qualitative evidence as pointing to an integration failure between local rule application and global action selection. This is a behavioral interpretation of visible traces, not a claim about the model’s internal mechanism.

5. Discussion

Systematicity vs. productivity. Following the classical systematicity/productivity distinction (Fodor and Pylyshyn, 1988; Lake and Baroni, 2018), models show partial systematicity as explicit rules and higher reasoning improve atomic move quality on variant-divergent positions. Because the evaluation is selected so that the atomic-best move differs from the standard-best move, the gap is best interpreted as a rule-composition gap under prior conflict: models struggle when a familiar action prior conflicts with an explicit rule update. The unfiltered atomic-source check clarifies the role of the filter: when prior-conflict is not enforced, the gap is diluted and model-dependent, with GPT-5.4 still showing a strong atomic degradation but Opus roughly tied by mean loss.

Why reasoning helps but does not close the gap. Reasoning likely improves local move evaluation but does not replace the candidate-generation prior π_S learned from stan-

standard chess. Models still appear to generate standard-chess candidates first and only partially correct them via the stated rules, a pattern illustrated by the qualitative observations. The aggregate gap and elevated blunder rates under no-rules prompting are consistent with this (Kolasani et al., 2025; OpenAI, 2024; DeepSeek-AI et al., 2025).

Implications for agentic and safety-relevant settings.

The most striking qualitative observation is unpropagated refutation: a model is given an explicit constraint, correctly identifies that a candidate action violates it, and takes the action anyway. Atomic chess is not a deployed tool-use setting, but it offers a closed-world analogue of this failure pattern in agentic contexts for an agent following a stated policy, an updated API, or changed preconditions. Because high reasoning effort reduces but does not eliminate the atomic-rule gap, test-time compute alone may be insufficient to integrate stated constraints with a strong prior policy. Benchmarks for constrained tool use should therefore include cases where the constraint directly contradicts a familiar default.

Limitations. Because the samples are filtered to variant-divergent positions, absolute performance is not representative of average standard or atomic chess play. The main pool contains $n=200$ positions, so estimates remain moderately noisy. We did not evaluate more positions, additional model families, or repeated samples because budget and compute costs limited the feasible evaluation scale. We include a smaller unfiltered comparison, but it uses only $n = 100$ positions and high-reasoning settings. Its model-dependent results should be interpreted as supporting the need for variant-divergent filtering rather than replacing the main benchmark. Raw centipawn losses from Stockfish and Fairy-Stockfish may not be directly comparable, so categorical labels use the Win% drop in Equations (2) and (3). The no-rules ablation shows explicit rule text matters but does not distinguish missing knowledge from failure to retrieve it. Lower public prevalence of atomic chess reduces but does not eliminate contamination risk.

6. Conclusion

We construct a paired evaluation of 200 variant-divergent positions from standard and atomic Lichess games, where the atomic-best move differs from the standard-best move. This tests memorization versus reasoning, as a model cannot solve these cases by simply selecting the move favored by standard chess priors. On these positions, models exhibit a substantial performance gap, performing worse under atomic rules than standard rules. Higher reasoning effort reduces this gap, and explicit rule statements help substantially relative to a no-rules prompt, but neither fully eliminates reliance on standard-chess move priors. Qualitative observations suggest that representation errors persist but

can be accompanied by composition errors, most diagnostically unpropagated refutation, where the model locally identifies a move as bad and selects it anyway. These results suggest that compositional rule substitution remains incomplete when a familiar action prior conflicts with a stated rule variant.

Impact Statement

This paper presents a diagnostic probe for compositional rule-following in language models, focused on rule-conditioned action selection for settings where a stated rule changes the consequences of actions in a familiar domain. This abstraction resembles simplified aspects of agents in the real world. The most novel finding, that high-reasoning models can locally identify an action as bad under stated rules and then select it anyway, is a closed-world analogue of a broader constrained-action failure pattern. Better evaluation of these failure modes may help identify brittle behavior before such models are deployed in domains where accuracy and alignment are critical.

References

- A. Ruoss, G. Delétang, S. Medapati, J. Grau-Moya, L. K. Wenliang, E. Catt, J. Reid, C. A. Lewis, J. Veness, and T. Genewein. Amortized planning with large-scale transformers: A case study on chess. *NeurIPS*, 2024.
- A. Karvonen. Chess-GPT’s internal world model. Technical report (blog post), 2024.
- A. Karvonen. Emergent world models and latent variable estimation in chess-playing language models. *arXiv:2403.15498*, 2024.
- Y. Zhang, X. Han, H. Li, K. Chen, and S. Lin. Complete chess games enable LLM become a chess master. *NAACL*, 2025.
- S. Kolasani, M. Saplin, N. Crispino, K. Montgomery, J. Q. Davis, M. Zaharia, Chi Wang, and Chenguang Wang. LLM CHESS: Benchmarking reasoning and instruction-following in LLMs through chess. *arXiv:2512.01992*, 2025.
- J. Liu, S. He, J. Wu, X. Wang, Y. Chen, Z. Kuang, S. Bao, and Y. Yao. ChessArena: A chess testbed for evaluating strategic reasoning capabilities of large language models. *arXiv:2509.24239*, 2025.
- X. Song, W. Wang, R. Cao, and Q. Hu. Beyond accuracy: A geometric stability analysis of large language models in chess evaluation. *arXiv:2512.15033*, 2025.

- Q. Wen, Z. Tang, and A. Anderson. ChessQA: Evaluating large language models for chess understanding. *arXiv:2510.23948*, 2025.
- Z. Wu, L. Qiu, A. Ross, et al. Reasoning or reciting? Exploring the capabilities and limitations of language models through counterfactual tasks. *NAACL*, 2024.
- A. Mészáros, P. Reizinger, and F. Huszár. Out-of-distribution tests reveal compositionality in chess transformers. *arXiv:2510.20783*, 2025.
- N. Carlini et al. Extracting training data from large language models. *USENIX Security*, 2021.
- N. Carlini et al. Quantifying memorization across neural language models. *ICLR*, 2023.
- R. T. McCoy, S. Yao, D. Friedman, M. D. Hardy, and T. L. Griffiths. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *PNAS*, 2024.
- N. Dziri et al. Faith and fate: Limits of transformers on compositionality. *NeurIPS*, 2023.
- J. A. Fodor and Z. W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1–2):3–71, 1988.
- B. Lake and M. Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. *ICML*, 2018.
- N. Kim and T. Linzen. COGS: A compositional generalization challenge based on semantic interpretation. *EMNLP*, 2020.
- D. Noever, M. Ciolino, and J. Kalin. The chess transformer: Mastering play using generative language models. *arXiv:2008.04057*, 2020.
- M. DeLeo and E. Guven. Learning chess with language models and transformers. *Data Science and Machine Learning (DSML)*, 2022.
- A. Stöckl. Watching a language model learning chess. *RANLP*, 2021.
- S. Toshniwal, S. Wiseman, K. Livescu, and K. Gimpel. Chess as a testbed for language model state tracking. *arXiv:2102.13249*, 2022.
- X. Feng, Y. Luo, Z. Wang, H. Tang, M. Yang, K. Shao, D. Mguni, Y. Du, and J. Wang. ChessGPT: Bridging policy learning and language modeling. *arXiv:2306.09200*, 2023.
- D. Monroe and P. A. Chalmers. Mastering chess with a transformer model. *arXiv:2409.12272*, 2024.
- OpenAI. Learning to reason with LLMs. Technical report, 2024.
- DeepSeek-AI, D. Guo, D. Yang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv:2501.12948*, 2025.
- J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 2022.
- J. Duan, R. Zhang, J. Diffenderfer, B. Kailkhura, L. Sun, E. Stengel-Eskin, M. Bansal, T. Chen, and K. Xu. GT-Bench: Uncovering the strategic reasoning capabilities of LLMs via game-theoretic evaluations. *NeurIPS*, 2024.
- H. Khan, H. A. Alyahya, Y. Alnumay, M. S. Bari, and B. Yener. ZeroSumEval: Scaling LLM evaluation with inter-model competition. *arXiv:2504.12562*, 2025.
- Y. Wu, Z. Jiang, A. Khan, Y. Fu, L. Ruis, E. Grefenstette, and T. Rocktäschel. ChatArena: Multi-agent language game environments for large language models. GitHub, 2023.
- Y. Zhang, S. Mao, T. Ge, X. Wang, A. de Wynter, Y. Xia, W. Wu, T. Song, M. Lan, and F. Wei. LLM as a mastermind: A survey of strategic reasoning with large language models. *arXiv:2404.01230*, 2024.
- Lichess. Atomic chess rules. Available at: <https://lichess.org/variant/atomic>.
- Lichess. Accuracy metric. Available at: <https://lichess.org/page/accuracy>.

A. Prompt Templates

This appendix gives the exact prompt templates used in the explicit atomic-rules condition, no-rules atomic condition, and standard-chess condition. The placeholders `<fen>`, `<color>`, and `<legal_moves>` are filled separately for each queried position.

Atomic chess prompt with rules provided

You are choosing the best move in Atomic chess.

Atomic chess rules:

1. When a capture occurs, an explosion happens on the destination square.
2. The captured piece, the capturing piece, and all non-pawn pieces on adjacent squares are removed.
3. Pawns adjacent to the explosion square are not removed.
4. Kings cannot make captures.
5. A move is illegal if the explosion removes the moving side’s own king.
6. A move is legal and immediately winning if the explosion from a capture removes the opponent’s king. This happens by capturing a piece adjacent to the king, not by capturing the king itself because kings cannot be captured.

Here is the position:

FEN: `<fen>`

Submit the best move in UCI notation for `<color>` in Atomic chess using the `submit_move` tool.

Legal moves in UCI notation:

`<legal_moves>`

Atomic chess prompt without rules provided

You are choosing the best move in Atomic chess.

Here is the position:

FEN: `<fen>`

Submit the best legal move in UCI notation for `<color>` in Atomic chess using the `submit_move` tool.

Legal moves in UCI notation:

`<legal_moves>`

Standard chess prompt

You are choosing the best move in chess.

Here is the position:

FEN: `<fen>`

Submit the best legal move in UCI notation for `<color>` in chess using the `submit_move` tool.

Legal moves in UCI notation:

`<legal_moves>`

B. Model Compute Statistics

All model calls used temperature 0, one sample per position, and submit-move tool output constrained to the supplied legal UCI move list. The evaluated model IDs were `claude-opus-4-6` and `gpt-5.4`. Table 4 reports mean latency per move for each model and reasoning-effort setting. Table 5 reports mean output tokens per move for high-reasoning runs, separated by ruleset.

Table 4. Average latency per move.

Model	Reasoning effort	Avg. latency (s)
Opus 4.6	none	1.91
GPT-5.4	none	1.34
Opus 4.6	high	138.02
GPT-5.4	high	393.05

Table 5. Average output tokens per move for high-reasoning runs.

Model	Ruleset	Output tokens / move
Opus 4.6	atomic	6,374
Opus 4.6	standard	2,720
GPT-5.4	atomic	2,484
GPT-5.4	standard	1,382

C. Qualitative Observation Excerpts

This appendix provides the observation excerpts underlying the qualitative categories in Section 4.5. Each boxed excerpt is a visible high-reasoning trace. Bold text marks the portions quoted or directly referenced in the main paper.

O1: Rule Instability

GPT-5.4

I need to choose the best move in Atomic chess from the given list, so I've got to call `submit_move` right away in commentary. I need to consider the rules carefully, like how captures result in explosions that destroy both the capturing and captured pieces, along with surrounding pieces, except for pawns, actually, **pawns aren't immune either!** Kings can't capture because they'd explode if they're adjacent to another piece. I'll analyze the position using the provided FEN to make the best legal move.

O2: Board State Drift

Opus 4.6

I think Qe7+ is more forcing. Let me go with that.
 Actually wait. Let me reconsider the whole position. **I have a queen on e7, a knight on a3**, and I'm trying to checkmate or win material. The black king is on e8.
 Qe7+ forces Kf8 (only legal move). Then what? I could play Qf7#? Is that checkmate? f7 is adjacent to e8, so the queen on f7 gives check. The king on f8 can move to... g8, g7, e8. So not checkmate.

O3: Unpropagated Refutation

Opus 4.6

But actually, Bb4+ looks like it might be **the strongest move here**. Let me work through the main variations: if White blocks with c3, Black has flexibility with moves like Qd6 or Ba3. If Kf1 instead, then Qb6 threatens Qxf2# and forces White into a defensive bind, though White can try Be3 to block the threat.
 The critical line is when White plays Nd2 to block the check—Black simply captures with Bxd2, and since d2 is adjacent to e1 where the king sits, the explosion from the capture is checkmate. The same fate awaits if White tries Bd2 instead. So Bb4+ appears to be winning in all variations. After Bb4+, if White plays c3 to block, I need to figure out Black's best continuation—maybe d4 to threaten the pawn, though I'm still working through the atomic chess capture rules and what happens to adjacent pieces. A capture happens, the capturing piece and captured piece are both removed along with adjacent non-pawn pieces. So dxc3: d4 pawn captures c3 pawn, both pawns removed, adjacent non-pawns removed. Adjacent to c3: b2(pawn-safe), b3(empty), **b4(bishop!)**, c2(empty), c4(empty), d2(empty), d3(empty), d4(was the capturing pawn, already removed). So **the black bishop on b4 would be removed! That's bad for black.**
 I think **the best approach is Bb4+** and after the forced responses, black gets a strong position.

O4: Pattern Reliance

GPT-5.4

In atomic chess, the exchange of pieces isn't as significant since material changes from explosions can be massive. **Using Nxh7 to explode h7 is a well-known tactic** because it causes the adjacent rook on h8 and the knight on g8 to disappear. There's a parallel idea with Nxf7, where blowing up the f7 square could potentially affect the king on e8, but since f7 is empty, that wouldn't apply here. So, Nxh7 really is **textbook!**

O5: Forcing Moves

GPT-5.4

Wait, let me re-examine. f6e4 — the knight goes from f6 to e4. **e4 is empty in the position. So it's just a move.**
 What about f6d5? That square's also empty.
 Now I'm reconsidering b4c3 — the bishop captures the knight on c3, which looks extremely strong since it removes white's knight along with both bishops. Let me calculate the material: black loses the bishop on b4, but white loses the knight on c3, the bishop on d3, and the bishop on d2. That's a net gain of two pieces for black, which is decisive.

D. Evaluation Set Examples

The main contrastive evaluation set contains 200 FENs: 100 sourced from atomic-chess Lichess games and 100 sourced from standard-chess Lichess games. The variant-divergence filter is applied so that each retained position has a different atomic-best and standard-best move, requiring the model to distinguish standard-rule move quality from atomic-rule move quality. Table 6 lists representative examples.

Table 6. Representative examples from the contrastive evaluation set.

Source	FEN
Atomic	3k1b1r/1p3ppp/3pp3/8/4P3/5P2/PPPP1nPP/R1B1K1NR w KQ - 0 11
Atomic	rn1qkbnr/p3p2p/1ppp1pp1/8/2N4Q/4P3/PPPP2PP/R1B1KB1R b KQkq - 1 8
Atomic	r1bqkbnr/1pp4p/4ppp1/3p4/3P3Q/2N1PN2/PPP2PPP/R1B1K2R w KQkq - 0 8
Atomic	r2q1knr/p6p/bpppppp1/8/7Q/3P4/PP2PPPP/R1B1KB1R b KQ - 1 12
Standard	r1b2Bk1/p4p1p/1qp3p1/2b1Pp2/8/2P4Q/PP4PP/RN2R2K b - - 0 18
Standard	r4rk1/1pq3pp/p1n2p2/3p4/Q2P2b1/2P2N2/PP1N2PP/2R1R1K1 w - - 0 17
Standard	r4rk1/pp4b1/2p1q2p/n4pp1/3PB3/2P2N2/PPKN4/R2Q3R w - - 0 24
Standard	6k1/5p1p/4p1p1/1P6/P7/r5P1/1R4KP/8 w - - 2 36