Disentangling generalization and memorization in large language models using chess

Leonard S. Pleiss
Technical University of Munich
leonard.pleiss@tum.de

Maximilian Schiffer
HEC Paris
schiffer@hec.fr

Robert K. von Weizsäcker
Technical University of Munich
weizsaecker@tum.de

Abstract

Large Language Models (LLMs) exhibit remarkable capabilities, yet it remains unclear to what extent these reflect sophisticated recall or genuine reasoning ability. We introduce chess as a controlled testbed aimed at disentangling these faculties. Leveraging the game's structure and scalable engine evaluations, we construct a taxonomy of positions varying in density of relevant priors—ranging from common states solvable by memorization to completely novel ones requiring generalization. Crucially, our approach achieves this distinction without requiring explicit knowledge of the models' training data. Applying this taxonomy, we combine a longitudinal analysis of the GPT lineage with a rigorous evaluation of contemporary models, including Claude Opus and Gemini. Our analysis reveals a steep gradient: performance consistently degrades as the density of relevant priors decreases. Notably, for tasks with few relevant priors, base model performance regresses to the random-play baseline, and all models propose illegal moves at sharply increasing rates, indicating that they fail to reliably apply the fixed rules of the game in unfamiliar positions. While newer models improve, progress slows significantly for tasks with sparse priors. Furthermore, while reasoning-augmented inference improves performance, its relative marginal benefit per token decreases in the absence of relevant priors. These results suggest limitations in systematic generalization, highlighting the need for mechanisms beyond scale to achieve robust performance when deprived of relevant priors.

1 Introduction

Large Language Models (LLMs) have achieved remarkable progress in recent years, setting new state-of-the-art results across a broad range of benchmarks and applications [Minaee et al., 2025]. This rapid advancement has led to widespread adoption in both research and industry [Raza et al., 2025, Naveed et al., 2025]. Yet, the origins of these performance gains are still under debate. It remains unclear whether they stem from enhanced retrieval across ever-expanding training corpora, or from improvements in genuine problem-solving and reasoning capability. This distinction is crucial: while both can yield impressive scores on specific benchmarks, only capabilities linked to genuine problem-solving ability enable an agent to systematically reason beyond its prior experience and generalize to novel problems [Chollet, 2019]. Understanding how much LLMs rely on memorization, and to which extent they are capable of generalizing beyond their training corpus therefore carries profound implications for their safety, interpretability and further trajectory.

Data and code are publicly available at `https://github.com/leonardpleiss/chess_llm_generalization`.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Empirically disentangling memorization and generalization poses a major challenge. It requires evaluating models on tasks that vary in availability of relevant priors: Performance on within-distribution (WD) tasks, i.e., tasks that were contained in the training data, is generally considered indicative of memorization capabilities, whereas out-of-distribution (OOD) performance is considered indicative of generalization [Chollet, 2019, Mészáros et al., 2026, Yang et al., 2025, Wu et al., 2025].

Yet, because modern LLMs are trained on massive and largely opaque datasets, quantifying the availability of relevant priors typically remains out of reach. As a result, existing studies into the origin of LLM capabilities frequently conflate both concepts, yielding fragmented evidence and sparking an ongoing debate (see Section 2).

We contribute to this debate by proposing a controlled, empirical framework that leverages chess as a structured environment for studying strategic reasoning in LLMs. As a domain combining clear rules, deep combinatorial structure, and computationally verifiable performance metrics via chess engines, chess provides a rigorous testbed for disentangling memorization and generalization: All chess games originate from a single initial state, and many early game positions are extensively studied. Yet, the number of reachable configurations grows combinatorially—approximately 10^{43} distinct legal board states exist [Shannon, 1950]—meaning that novel positions frequently arise after only a few moves. Therefore, chess engages both memorization and generalization: play on known, theoretical positions benefits from the ability to recall known positions, whereas succeeding in novel positions demands generalization beyond memorized game states and patterns (see Appendix A.2).

In this paper, we conduct a systematic evaluation of LLM performance in the structured domain of chess across tasks that require differing degrees of memorization and generalization. We do not seek to validate or dismiss LLM capabilities, but to provide rigorous, domain-grounded evidence that can inform the ongoing discourse on the origin of LLM capabilities and their developmental trajectory.

Our results reveal a consistent performance gradient across all tested models: performance declines as the density of relevant priors decreases. This observation corroborates recent literature [Qi et al., 2024]. We extend these foundations in four ways. First, without reasoning augmentation, model performance collapses to random levels when relevant priors are sparse. Second, the rate of illegal moves rises sharply as relevant priors become sparser, even though the rules that determine legality are identical in every position; we regard this as our most robust evidence of a generalization failure. Third, while newer model generations improve steadily, the gains from one generation to the next diminish, most strongly for tasks with few relevant priors. Fourth, reasoning-augmented inference provides measurable benefits, but its relative gain per reasoning token decreases when relevant priors are sparse. Collectively, these findings highlight persistent weaknesses in state-of-the-art LLMs when generalizing beyond the training data manifold.

2 Related work

With this investigation, we aim to contribute to an active debate on the extent to which LLMs exhibit memorization versus emergent reasoning and generalization. An increasing number of studies explore the degree to which current LLMs can effectively solve tasks beyond their training data distribution. However, the evidence remains inconclusive.

Some results point to genuine generalization—or even emergent capabilities [Wei et al., 2022, Ganguli et al., 2022, Srivastava et al., 2023, Brown et al., 2020, Si et al., 2024, Bubeck et al., 2023, Greatrix et al., 2024, Georgiev et al., 2025]. For instance, Si et al. [2024] found that LLM-generated research ideas were rated as more novel than those of human experts, while Greatrix et al. [2024] observed that LLMs succeed in spatial reasoning tasks with negligible data overlap. Conversely, other work highlights persistent limitations in reasoning or generalization [Chollet, 2019, Chollet et al., 2025, Yang et al., 2025, Duan et al., 2024]. The Abstraction and Reasoning Corpus ARC-AGI-3 [ARC Prize Foundation, 2026] remains largely unsolved. Moreover, Yang et al. [2025] demonstrate that while LLMs excel in low-level cognitive tasks, they struggle in higher-order reasoning settings. Such findings suggest that, despite impressive in-context learning and retrieval capabilities, LLMs may still lack the kind of flexible, compositional reasoning required to meaningfully generalize beyond the training data manifold.

One central reason for this ongoing debate and diverging findings is that existing benchmarks often lack explicit high-quality information on task contamination [Li and Flanigan, 2024], which makes it

difficult to disentangle memorization and generalization. Without knowledge of the training corpus, attributing performance to generalization remains inherently speculative. The lack of consensus within this debate therefore largely reflects the resulting challenge in reliable measurement and clean experimental design: A rigorous comparison between memorization and generalization in LLMs requires evaluation across tasks that differ systematically in their proximity to the model's training data distribution. Because the precise content of training corpora remains unknown, most existing studies rely on benchmarks that implicitly conflate recall and reasoning.

Various papers have recently tackled the challenge to disentangle memorization and generalization in language models, highlighting the subject's enormous practical relevance [Wu et al., 2025, Yang et al., 2025, Qi et al., 2024, Xie et al., 2025, Wang et al., 2025b, Mészáros et al., 2026]. Most approaches rely on small-scale transformer models to circumvent the issue of training corpus opacity. However, as scale is known to fundamentally alter the dynamics within language models [Wei et al., 2022], it is unclear how these findings generalize to large-scale LLMs.

We posit that the game of chess offers a unique opportunity to disentangle memorization and generalization in state-of-the-art LLMs. Chess has played a foundational role in the history of artificial intelligence [Heath and Allum, 1997, Schaeffer and Plaat, 1997]. Yet, the work at the intersection of language modeling and chess thus far primarily focused on general benchmarking or improving domain-specific performance through finetuning and supervised architectural specialization [Wang et al., 2025a, Ruoss et al., 2024, Hwang et al., 2025, Kuo et al., 2023, Kolasani et al., 2025]. Only a single work employs chess to investigate generalization versus memorization in transformer architectures [Mészáros et al., 2026]. It relies on a relatively small-scale model, trained via supervised learning, limiting its insights into how their findings generalize to state-of-the-art LLMs. It therefore remains open how general-purpose frontier LLMs perform zero-shot as positions become increasingly unfamiliar (see Appendix D for a more detailed literature review).

Contrary to this existing stream of research, we employ the unique characteristics of the chess domain to disentangle memorization and generalization in large-scale state-of-the-art language models. The game's combinatorial structure, its well-defined rules and extensive corpora of games enable us to estimate the density of relevant priors in opaque, proprietary training corpora. This unique characteristic facilitates a principled analysis of how contemporary general-purpose LLMs operate within and beyond their training distribution, and yields valuable insight into the intellectual nature of these systems.

3 Methodology

To systematically evaluate LLM capabilities as a function of the availability of relevant priors, we employ chess as a controlled testbed combining combinatorial complexity with scalable engine evaluation. This setup allows us to disentangle memorization and generalization by varying positional familiarity. We subsequently (i) formalize the link between performance under varying densities of relevant priors, memorization and generalization, (ii) instantiate this framework in chess, (iii) generate WD, near-distribution (ND), and OOD datasets, (iv) define performance metrics, and (v) detail the experimental protocol.

Conceptual framework. We study the origin of domain-specific ability within LLMs through the lens of *memorization* and *generalization*. Following the conceptual framework proposed by Chollet [2019], we align this distinction with machine learning terminology by formalizing these concepts in terms of performance under different data distributions.

Definition 3.0.1 (Memorization and generalization in LLMs)**.** Let $p_{\text{train}}(x)$ denote the (unknown) training data distribution of an LLM and let $p_{\text{test}}(x)$ be a test distribution over inputs $x \in \mathcal{X}$ within a given domain. We define:

Memorization as the model's ability to perform accurately under *WD* conditions, i.e., when $x \sim p_{\text{train}}(x)$. This reflects the model's ability to solve tasks when relevant priors are available, predominantly through retrieval and recall of previously observed solutions.

Generalization as the model's ability to perform accurately under *OOD* conditions, i.e., when $x \sim p_{\text{test}}(x)$ with $p_{\text{test}} \neq p_{\text{train}}$. It reflects the model's ability to solve tasks when relevant priors are sparse or entirely unavailable, indicating systematic and adaptive first-principle reasoning beyond previously memorized solutions to solve novel tasks within a familiar domain.

To operationalize this distinction, we quantify an LLM's performance ratio between WD and OOD conditions as a *generalization gap*,

$$R_{\text{gen}} = \frac{\mathbb{E}_{x \sim p_{\text{test}}}[L(f(x))]}{\mathbb{E}_{x \sim p_{\text{train}}}[L(f(x))]}, \tag{1}$$

where $L(f(x))$ is a loss or performance metric associated with the model's output $f(x)$ on input x. High WD accuracy reflects strong memorization. Accordingly, as $R_{\text{gen}} \to 1$, OOD performance approaches WD levels, signifying stronger generalization. Naturally, this estimate is only meaningful if WD performance exceeds random levels.

Quantifying this gap requires an isolated measure of WD and OOD performance in state-of-the-art LLMs, which is notoriously difficult because the underlying data distribution is both vast and ill-defined. In consequence, the density of relevant priors is generally difficult to estimate. However, we believe that the domain of chess offers a unique opportunity to approximate this density.

Approximating the density of relevant priors in chess. The state space of chess is combinatorially vast, with Shannon [1950] famously bounding the state-space complexity at roughly 10^{43}. Within this space, a small subset of configurations—predominantly opening positions—is extensively documented and frequently encountered. Given the extensive available data (literature, forum discussions, database appearances) on these frequently encountered positions, we assume they are likely covered by large-scale pre-training corpora, and denote them as WD positions: Models can likely achieve strong performance through pure memorization. Conversely, the overwhelming majority of valid chess positions are not well-studied, indicating that they may not be solved via direct recall. However, the fact that a given position is not well-known does not guarantee that there are few relevant priors: Many configurations maintain significant structural homology with familiar states, allowing models to succeed via interpolating known heuristics, which combines elements of memorization and generalization. We term these ND positions. To rigorously assess generalization and compel true first-principles reasoning, the evaluation distribution must employ positions that minimize structural similarity to the training manifold. We denote these as OOD positions.

Position categories. To evaluate WD, ND and OOD performance, we construct three subsets of the chess state space, $\mathcal{X}_{\text{WD}}, \mathcal{X}_{\text{ND}}, \mathcal{X}_{\text{OOD}}$, each corresponding to a different estimated density of relevant priors within the LLM's training distribution. We provide a detailed empirical basis for the condition-specific design principles and exact construction rules in Appendix A.

Within-distribution positions ($\mathcal{X}_{WD}$). These are well-known positions that occur frequently in human play and are likely present within the training data. We assign a position $x \in \mathcal{X}$ to $\mathcal{X}_{\text{WD}}$ if it occurs at least $n_{\min} = 1{,}000$ times in the Lichess Masters database [Lichess.org, 2025]. Such positions typically arise within extensively studied openings lines and therefore approximate WD conditions, $p_{\text{train}}(x_{\text{WD}}) \approx \max_{x \in \mathcal{X}} p_{\text{train}}(x)$.

Near-distribution positions ($\mathcal{X}_{ND}$). ND positions cover an intermediate regime between WD and OOD states. They represent configurations that are *absent* from the Lichess Masters database yet retain some structural characteristics commonly encountered in WD states (e.g., material balance, piece development, pawn structure). We generate these by performing k=10 random legal moves (20 plies) from the starting position and discarding any position that appears in the database. ND positions test associative recall, i.e., the recall of solutions from structurally similar positions, which combines elements of generalization and memorization.

Out-of-distribution positions ($\mathcal{X}_{OOD}$). OOD positions represent states with few relevant priors. To minimize the utility of associative recall, we aim to minimize the structural similarity to positions likely encountered during training. We construct OOD positions by uniformly sampling legal configurations through random placement of ten pieces per side onto the board, allowing overwrites and subject to position validity constraints (i.e., exactly one king per side, no impossible pawn placements, the side not to move cannot be in check). As intended, this creates theoretically feasible yet atypical board states that are extremely unlikely to arise during human play, forcing the agent to reason from first principles instead of relying on known heuristics.

Dataset summary. For each condition, we generate N=500 distinct positions using the `python-chess` library [Fiekas, 2014], yielding a balanced dataset of $1{,}500$ board configurations. We refer to Figure 1 and Appendix B for example positions.

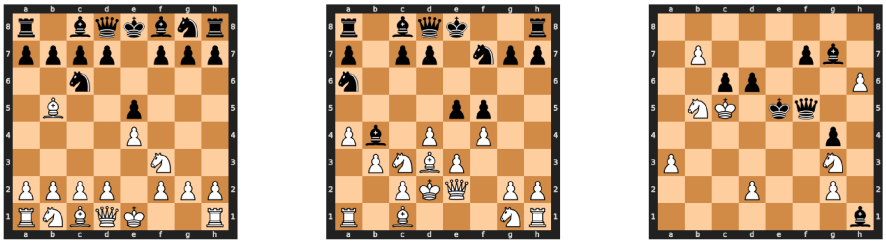


Figure 1: Example positions. Left to right: within-distribution, near-distribution, out-of-distribution. For a detailed discussion of our findings in these positions, we refer to Appendix B.

Move evaluation. To quantify the quality of model-recommended moves, we employ the Centipawn Loss (CPL), a standard measure of decision regret in the chess domain. Unlike other metrics like Win/Loss/Draw Probability, the CPL provides a meaningful signal on move quality, even in imbalanced positions. The CPL measures the positional value lost through a move. A centipawn corresponds to one hundredth of a pawn's nominal value, providing a more granular measure of move quality than other established metrics like win probability. Formally, for a position evaluated by a chess engine before and after a candidate move, we define $\ell_{\text{CPL}} = \min(\max(0,\, \text{eval}_\text{b} - \text{eval}_\text{a})\,, 1{,}000)$, where eval_b and eval_a denote the evaluations from the moving player's perspective before and after executing the selected move. Lower values of ℓ_{CPL} indicate stronger play, while higher values reflect increasing positional deterioration. By definition, the best move in each position has $\ell_{\text{CPL}} = 0$. As a broad empirical guideline, $\ell_{\text{CPL}} < 10$ signifies moves effectively on par with the engine's recommended move, $10 \leq \ell_{\text{CPL}} < 50$ corresponds to minor inaccuracies, $50 \leq \ell_{\text{CPL}} < 100$ to noticeable mistakes, and $\ell_{\text{CPL}} \gg 100$ to blunders that substantially reduce winning chances. We acknowledge that CPL is a heuristic metric; however, averaged over a large corpus, it serves as a robust proxy for systematic alignment with superhuman chess engines (see Appendix C). In addition to positional quality, we also track the frequency of *illegal moves*, i.e., moves that violate the rules of chess. The proportion of illegal moves serves as a direct indicator of reasoning capabilities—the ability to strictly adhere to formal constraints. According to the rules, playing an illegal move results in instant loss, hence it is scored equally to losing through checkmate, $\ell_{\text{CPL}} = 1{,}000$. We conducted all evaluations using Stockfish 17.1 at search depth 30 [Romstad et al., 2008], which provides a robust trade-off between computational cost and evaluation reliability.

Models. We analyze the evolution of WD, ND and OOD performance across LLMs. We focus primarily on OpenAI's GPT family to track performance across successive generations of one widely deployed lineage, and evaluate contemporary models from Anthropic and Google to assess whether our findings extend beyond it. We evaluate three GPT generations (Table 1): GPT-3.5, GPT-4o, and GPT-5, the most recent GPT generation at the time of data collection. Because GPT-5 allows configurable reasoning effort, we test it at *minimal* and *moderate* effort to assess the impact of reasoning augmentation; we refer to configurations without or with minimal reasoning as *base models*. For the cross-lineage comparison, we evaluate `gemini-3-flash-preview` and `claude-opus-4-6` at their lowest available reasoning settings against GPT-5 at minimal effort. As reasoning budgets differ across providers, this constitutes a common low-reasoning baseline that minimizes cross-provider differences in test-time computation, rather than strict parity.

Evaluation protocol. All GPT responses were collected between August and October 2025 via the OpenAI API; Claude and Gemini responses were obtained in March 2026. Each model received identical prompts requesting a single move recommendation in standard chess notation. As our

Table 1: Overview of model generations, API tags, and release dates

Model Name	API Tag	Release
GPT-3.5	gpt-3.5-turbo	Mar 2023
GPT-4o	gpt-4o	May 2024
GPT-5	gpt-5	Aug 2025
Gemini 3	gemini-3-flash-preview	Dec 2025
Claude Opus	claude-opus-4-6	Feb 2026

central quantities are within-model contrasts across conditions, the differing collection windows do not affect them, but they caution against reading small differences between lineages as rankings. Each query–response pair was logged with model metadata and evaluation metrics to ensure full reproducibility.

Baselines. To contextualize LLM performance, we introduce two reference points. First, we obtain a *random baseline* by uniformly sampling from the set of legal moves in each position. This represents a lower bound and mirrors play under knowledge of chess rules but without any strategic or tactical capability. Second, as an upper bound representative of domain-specialized neural network capability, we employ *Leela Chess Zero* (Lc0, Pascutto and Linscott [2021]), an open-source state-of-the-art neural network chess engine trained via self-play reinforcement learning and derived from the AlphaZero framework [Silver et al., 2017]. It employs Monte Carlo Tree Search (MCTS) guided by neural network evaluations, combining pattern recognition with lookahead search. To limit the influence of deep tree search and emphasize the network's evaluative capabilities, we enforce a strict compute budget of two seconds per position. Together, these baselines define the boundaries of a spectrum from rule-constrained random play to domain-optimized neural systems, enabling us to situate LLM performance across model generations and reasoning configurations within a well-calibrated performance landscape. For details on the configuration, we refer to Appendix A.5.

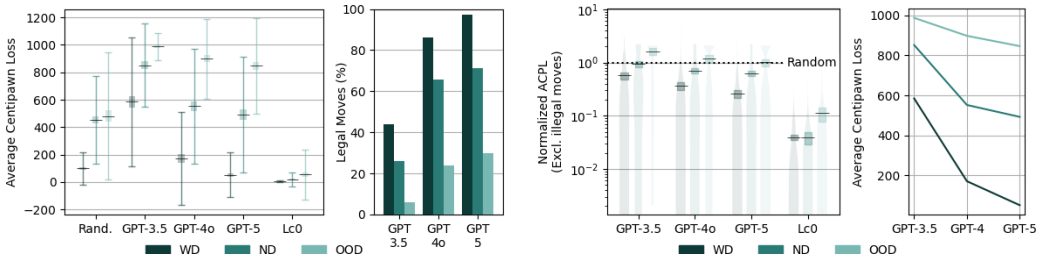
Prompting and parsing. Each chess position $x \in \mathcal{X}$ was encoded using Forsyth-Edwards Notation (FEN) and provided to each model via a standardized prompt requesting the strongest move in Standard Algebraic Notation (SAN) and Universal Chess Interface (UCI) notation, along with a short justification, in a standardized JSON format. The prompt template was held constant across all model configurations to eliminate stylistic bias and isolate systematic differences in capability. The template prompt is shown in Appendix A.1. All prompt-response pairs, along with metadata on model identity, token expenditure, and evaluation outcomes, were logged in structured JSON format to enable full reproducibility and downstream analysis.

The recommended moves in SAN notation were checked for syntactic validity. Across all models, syntactically invalid moves occurred in 2.59% of positions (0.26% for GPT-3.5, 0.02% for GPT-4o, 0.73% for GPT-5 with minimal reasoning, 0.07% for GPT-5 with moderate reasoning, 0.047% for Gemini, 14.4% for Claude-Opus). Syntactically valid moves that violate the rules of chess in the given position are counted as illegal moves. Unless stated otherwise, all results are based on the SAN move. To assess robustness to the output notation, we additionally evaluated the UCI-formatted move that each model returned within the same response; all findings replicate under this notation (Appendix G.3).

4 Results

In the following section, we report how model performance evolved across generations, and how it varies across reasoning conditions and model lineages. We begin by comparing the average centipawn loss and the frequency of illegal moves across GPT-3.5, GPT-4o, and GPT-5, highlighting consistent but decelerating improvements over time. We then normalize performance metrics to partially control for differences in condition difficulty and compare the observed rates of improvement across generations. Next, we investigate how increased reasoning effort affects GPT-5's chess performance, both in terms of accuracy and computational cost. Lastly, we move beyond the GPT family and compare performance across different state-of-the-art models.

Average centipawn loss. The left panel of Figure 2a depicts a consistent increase in average centipawn loss as relevant priors become sparser. Across all models, performance is best for WD positions and worst for OOD ones: the Average Centipawn Loss (ACPL) in ND positions is $4.75\times$ higher than in WD positions, and the ACPL in OOD positions is $7.79\times$ higher than in WD positions. A cross-condition comparison reveals steady but slowing progress: GPT-3.5 achieved an average ACPL of 807.97 ($SD = 368.58$), GPT-4o of 540.38 ($SD = 461.25$), and GPT-5 of 463.5 ($SD = 463.11$), corresponding to a 33.12% ACPL reduction from GPT-3.5 to GPT-4o and 14.23% from GPT-4o to GPT-5. The increasing variance across model versions primarily arises from the models' improved ability to avoid exclusively making poor or illegal moves, thereby broadening the distribution of outcomes. Despite this progress, GPT-3.5, 4o and 5 perform worse than random play across all conditions, with GPT-5 in WD positions being the sole exception. This indicates that outside of highly familiar contexts, the models struggle to consistently outperform a blind baseline that simply



(a) Raw Performance

(b) Normalized Performance

Figure 2: **(a)** Left: Average Centipawn Loss with Matescore = Illegal Move Score = 1,000. Right: Proportion of legal moves. **(b)** Left: Random-normalized Average Centipawn Loss under exclusion of illegal moves. Right: Observed rates of improvement across three generations. **Common parameters:** Horizontal lines indicate means, colored boxes around the mean display the 95% confidence interval, and vertical lines indicate standard deviations. WD = within-distribution, ND = near-distribution, OOD = out-of-distribution.

adheres to the rules. This effect is partly due to the high prevalence of illegal moves: The LLMs frequently incur ACPLs of 1,000 due to playing illegal moves, whereas the random policy exclusively samples from the set of legal moves, thereby avoiding the penalty.

Lc0, a domain-specialized engine trained via self-play without exposure to human games, achieves a mean ACPL of 25.10 ($SD = 110.12$), with ACPLs per condition ranging from 3.91 ($SD = 5.1$) to 53.72 ($SD = 179.61$), despite limited reliance on explicit search due to the two-second time limit.

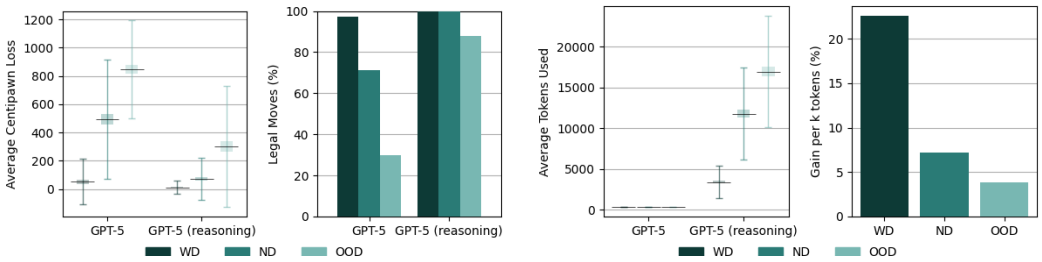
Illegal moves. The right panel of Figure 2a reveals that the frequency of illegal moves consistently increases with decrease in relevant priors. Across all model versions, illegal moves are least frequent in WD positions and most frequent in OOD positions. Specifically, the proportion of illegal moves is $1.46\times$ higher in ND positions and $4.72\times$ higher in OOD positions compared to WD ones. Lc0 does not play a single illegal move in any position.

Within the GPT lineage, illegal move rates decrease consistently across successive model versions, suggesting ongoing improvement. GPT-3.5 produced illegal moves in 74.7% of positions, GPT-4o in 41.6%, and GPT-5 in 33.8%. This corresponds to a decrease in illegal move frequency of 44.31% from GPT-3.5 to GPT-4o and 18.75% from GPT-4o to GPT-5, indicating a deceleration. The high rate of illegal moves in OOD positions ($> 30\%$ for GPT-5) underscores that the difficulty lies not merely in strategic planning, but in the fundamental maintenance of rule consistency when memorized patterns are unavailable.

Normalized average centipawn loss for legal moves. To isolate pure move quality, we reanalyzed ACPL under exclusion of illegal moves. We further normalized results by dividing each condition's ACPL by that of a random legal move policy. This calibrates each condition against the expected cost of uninformed legal play. In doing so, we obtain normalized ACPL scores independent of legal move frequency that are comparable across conditions, and partially control for differences in condition difficulty (Appendix C.6).

The left panel of Figure 2b shows that this random-normalized ACPL replicates the pattern observed previously: performance declines monotonically as relevant priors become sparser. Under exclusion of illegal moves, we observe decent performance in positions with relevant priors. For instance, in WD positions, GPT-5 achieved a centipawn loss of 26.19 ($SD = 49.53$), outperforming random move selection by a factor of approximately 4. Notably, even under the exclusion of illegal moves, performance within OOD positions does not exceed random performance for any of the base models. This suggests that, when relevant priors are sparse, base models exhibit no measurable strategic capability beyond random legal play.

Rates of improvement. The right panel of Figure 2b illustrates that, while models consistently improve, the rate of improvement decreases across generations. Specifically, from GPT-3.5 to GPT-4o, we observed rates of ACPL improvement of 70.7% for WD positions, 35.16% for ND positions, and 9.08% for OOD positions. From GPT-4o to GPT-5, the improvement rates dropped to 69.92%, 10.77% and 5.73%. Notably, progress slows with decrease in relevant priors in both proportional and



(a) Reasoning impact on performance

(b) Reasoning cost and gain per token

Figure 3: **(a)** Left: Average Centipawn Loss with minimal and moderate reasoning effort. Right: Proportion of legal moves with minimal and moderate reasoning effort. **(b)** Left: Reasoning tokens used by GPT-5 per position with minimal and moderate reasoning effort. Right: Percentage point improvement per token by condition. **Common parameters:** Means as horizontal black lines, 95% confidence interval as colored boxes, whiskers indicate standard deviations. WD = within-distribution, ND = near-distribution, OOD = out-of-distribution.

absolute numbers. These diminishing returns suggest that scaling current architectures yields progressively smaller generalization gains. While this decrease in return is negligible for WD positions (1.1% reduction), it is substantial in ND and OOD positions (69.36% and 36.89% reduction). These diminishing returns suggest that continuous, generation-over-generation refinements—encompassing data curation, post-training alignment, and architectural updates—yield progressively smaller generalization gains.

Impact of reasoning. We further examined the role of chain-of-thought reasoning. We compared GPT-5 with minimal reasoning effort against GPT-5 with moderate reasoning effort. Figure 3a illustrates that additional reasoning effort markedly improved performance across all conditions, reducing both average centipawn loss (left panel) and frequency of illegal moves (right panel). While reasoning yields the largest absolute performance gains in the OOD condition and the smallest in the WD condition, absolute differences are inherently confounded by the vastly different baseline performances across conditions. To ensure a calibrated comparison, we therefore focus our subsequent analysis on relative performance gains. Across conditions, the ACPL decreased by 71.74%. The relative gain was most pronounced in ND positions (85.3% decrease), followed by WD (77.25% decrease) and OOD (64.69% decrease) positions. Interestingly, reasoning widens the relative generalization gap: With minimal reasoning, the ACPL in the OOD condition was approximately 16 times higher than in the WD condition. With moderate reasoning, this gap widens to a factor of approximately 25.

We once again observe that performance follows a clear gradient mirroring the density of relevant priors. Specifically, for WD positions, reasoning led to an ACPL of 11.72 ($SD = 47.44$), reflecting a caliber of play characteristic of advanced human practitioners. Yet, in OOD positions the model incurs an ACPL of 298.81 ($SD = 427.61$), still effectively creating a losing disadvantage on most moves. This observation suggests that even with reasoning enabled, the chain-of-thought process struggles to construct valid strategies in the absence of recognizable pattern anchors.

The left panel of Figure 3b shows the automatically allocated number of reasoning tokens to increase as relevant priors become sparser: GPT-5 with moderate reasoning effort allocated an average of $3{,}426.41$ ($SD = 1{,}967.16$) tokens (≈ 10.28 pages, assuming 0.75 words per token and 250 words per page, OpenAI [2025]) for WD positions, $11{,}800.1$ ($SD = 5{,}606.32$) tokens (≈ 35.43 pages) for ND positions, and $16{,}952.54$ ($SD = 6{,}831.54$) tokens (≈ 50.9 pages) for OOD ones. This paradoxical increase in computational cost for OOD tasks—where the model allocates the most resources but achieves the least relative gain—points to an inefficient search process when the solution manifold is unknown.

The right panel of Figure 3b presents a further analysis revealing that relative performance gains per token diminish as the density of relevant priors decreases: In WD positions, $1{,}000$ reasoning tokens yield a relative marginal performance gain of 22.55%, dropping by 67.98% to 7.23% in ND positions, and by 83.10% to 3.81% in OOD ones. This diminishing relative marginal utility indicates that while reasoning compensates for some deficits, it does not resolve the fundamental challenge of generalizing beyond the training data distribution. Critically, it implies that current

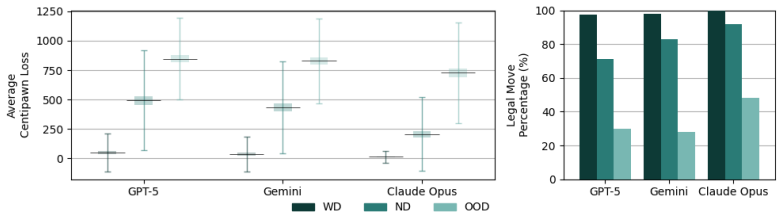


Figure 4: Performance across model lineages, all at minimal reasoning effort. Left: Average Centipawn Loss with Matescore = Illegal Move Score = 1,000. Right: Proportion of legal moves. Horizontal lines indicate means, boxes around the mean display the 95% confidence interval, vertical lines indicate standard deviations. WD = within-distribution, ND = near-distribution, OOD = out-of-distribution.

chain-of-thought mechanisms act primarily as amplifiers of memorization rather than as independent engines of generalization.

Performance across state-of-the-art models. As displayed in Figure 4, Claude and Gemini replicate the same qualitative performance gradient as the GPT lineage. In the WD condition, Claude Opus achieves an ACPL of 14.0 ($SD = 49.8$) with an illegal move rate of 0.2%, while Gemini posts an ACPL of 34.8 ($SD = 146.8$) with a 2.2% illegal move rate. For the ND condition, Claude Opus records an ACPL of 207.2 ($SD = 314.0$) alongside an 8.2% illegal move rate, and Gemini's ACPL shifts to 434.1 ($SD = 390.0$) with 17.0% illegal moves. Finally, in the OOD condition, Claude Opus yields an ACPL of 728.6 ($SD = 427.0$) with 51.6% illegal moves, and Gemini reaches an ACPL of 829.7 ($SD = 359.7$) with an illegal move rate of 72.0%. Even these most recent models thus propose illegal moves in more than half of the OOD positions, compared with at most 2.2% in WD positions.

5 Discussion

This study examined memorization and generalization, their development across model generations, and reasoning dynamics in LLMs using the structured domain of chess, and finds a consistent structure in LLM performance. First, *memorization remains a dominant strength, yet generalization seems brittle, indicating a generalization gap*: When positions resemble those likely encountered during training, models exhibit decent play. However, when relevant priors are sparse, base model performance collapses to random move selection, and the illegal move rate spikes. The steep performance gradient across both illegal moves and move quality suggest that current models have not acquired a robust, generalizable representation of the chess world, but rely on patterns tied to familiar configurations that become brittle once the structural regularities of standard play are removed. We note that the two metrics underlying this finding differ in evidential strength. The rise in illegal moves is the more robust indicator of a generalization failure: legality is governed by rules that are identical in every position, and verifying move legality is a fully specified, bounded procedure. An agent that has internalized these rules should not become less rule-compliant in unfamiliar positions, as the available evidence on human players suggests (Appendix C.7). The decline in move quality to a random baseline is consistent with this trend, but is not conclusively isolated from a potential relationship between novelty and difficulty. Second, *successive model generations show diminishing returns*: although we observe measurable improvements from generation to generation, their magnitude appears to diminish over time. Third, *reasoning helps, but gains diminish when priors are sparse*: Reasoning substantially lowers both illegal move rates and centipawn loss. However, its relative gain per token diminishes as relevant priors become sparser, and the relative generalization gap widens under reasoning. This is consistent with the hypothesis that chain-of-thought mechanisms in their current form function primarily as retrieval optimizers rather than as engines of first-principles derivation. Overall, our results point towards substantial weaknesses in the OOD performance of LLMs in chess, and suggest that neither generation-over-generation refinements nor reasoning augmentation, in their current form, reliably extend model competence beyond the training data manifold.

Limitations. Our work comes with multiple limitations. First, our design is observational: Without access to proprietary training corpora, prior density cannot be varied independently of positional structure, so conditions differ jointly in estimated prior density and in properties such as material, board topology, and tactical character (Appendix C.6). Thus, our findings across model generations and reasoning settings are strictly descriptive, capturing empirical trends rather than establishing universal scaling laws. Second, we cannot guarantee the separation of WD and OOD positions within the training corpus. While we hypothesize that leakage would attenuate the observed gap, we cannot verify this empirically (Appendix C.3). Third, despite random-policy normalization and an Lc0 baseline, some confounding between distribution proximity and tactical complexity is unavoidable, narrowing the scope of evaluated generalization primarily toward tactical calculation and rule application rather than broader strategic reasoning (Appendix C.2). Fourth, in absence of a human baseline, our move-quality results should be interpreted with caution (Appendix C.7). Fifth, our findings replicate across output notations (Appendix G.3), but sensitivity to the input encoding remains untested. Finally, engine-based centipawn loss is imperfect since chess is unsolved, but robust when aggregated over many positions (Appendix C).

Implications beyond chess. Chess constitutes a clean, fully observable and analytically rich testbed for studying the interaction between memorization and generalization. The game's fixed rules, combinatorial structure, and reliable evaluations using superhuman chess engines enable a systematic assessment of generalization without the confounding factors that typically accompany natural language tasks. The cognitive demands required to excel at chess—logically sound reasoning, long-horizon planning, abstract pattern recognition, and exact adherence to formal rules—closely mirror the core competencies necessary to succeed in other complex domains. Consequently, the motifs observed in this study—robust memorization, brittle generalization, and diminishing returns across successive model generations—may not be idiosyncratic to the chess domain. Rather, our findings possibly represent a manifestation of a more general phenomenon in LLMs that defies measurement in other domains: the systemic difficulty of inferring structural rules to solve novel tasks without close priors.

The diminishing gains observed across successive model generations further suggest that the refinements driving recent progress may not suffice to enable robust generalization. If this pattern extends beyond the present domain, it would imply that model improvements primarily enhance recall and pattern completion within data-rich regions of the training distribution, while leaving structurally novel or low-density regions largely unresolved. Hence, scaling appears to deepen memorized knowledge rather than to broaden fluid adaptability. A similar picture emerges for reasoning. Although chain-of-thought reasoning improves performance, its relative gain per token diminishes as relevant priors become sparser, which challenges the view that reasoning alone can compensate for limited generalization. One interpretation consistent with our data is that extended reasoning chains primarily correct local inconsistencies and surface-level errors rather than inducing new conceptual representations. Under this interpretation, reasoning in current architectures functions mainly as a mechanism for exploiting existing knowledge rather than serving as a reliable pathway toward genuine conceptual synthesis.

Finally, while we anticipate that our findings extend beyond the game of chess, it is crucial to recognize that our empirical findings are limited to a single domain that is characterized by perfect information, and governed by strict combinatorial logic and immense state-space complexity. It remains an open question to what extent the observed limitations transfer to other domains, specifically to highly semantic, probabilistic, or open-ended linguistic ones, where LLMs frequently demonstrate robust associative flexibility, or to domains receiving extensive targeted post-training.

Future work. Promising extensions include (i) a constrained setting in which models choose from the complete set of legal moves, separating move generation from strategic evaluation; (ii) rare but reachable positions matched on material and legal-move count; (iii) alternative input encodings; (iv) a human baseline on randomized positions; and (v) analyses of reasoning traces.

Acknowledgments and Disclosure of Funding

L.S.P. was supported by a Roland Berger doctoral fellowship. Roland Berger had no role in the design, execution, analysis, or reporting of this research. The authors declare no competing interests.

References

ARC Prize Foundation. ARC-AGI-3: A New Challenge for Frontier Agentic Intelligence. *arXiv*, 2026.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *International Conference on Neural Information Processing Systems*, 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *arXiv*, 2023.

William G. Chase and Herbert A. Simon. Perception in chess. *Cognitive Psychology*, 1973.

Francois Chollet, Mike Knoop, Gregory Kamradt, Bryan Landers, and Henry Pinkard. ARC-AGI-2: A New Challenge for Frontier AI Reasoning Systems. *arXiv*, 2025.

François Chollet. On the measure of intelligence. *arXiv*, 2019.

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 limitations of LLMs via game-theoretic evaluations. In *International Conference on Neural Information Processing Systems*, 2024.

Niklas Fiekas. python-chess, 2014. URL `https://python-chess.readthedocs.io/`.

Deep Ganguli, Danny Hernandez, Liane Lovitt, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova Dassarma, Dawn Drain, Nelson Elhage, Sheer El Showk, Stanislav Fort, Zac Hatfield-Dodds, Tom Henighan, Scott Johnston, Andy Jones, Nicholas Joseph, Jackson Kernian, Shauna Kravec, Ben Mann, Neel Nanda, Kamal Ndousse, Catherine Olsson, Daniela Amodei, Tom Brown, Jared Kaplan, Sam McCandlish, Christopher Olah, Dario Amodei, and Jack Clark. Predictability and surprise in large generative models. In *Conference on Fairness Accountability and Transparency*, 2022.

Bogdan Georgiev, Javier Gómez-Serrano, Terence Tao, and Adam Zsolt Wagner. Mathematical exploration and discovery at scale. *arXiv*, 2025.

Thomas Greatrix, Roger Whitaker, Liam Turner, and Walter Colombo. Can large language models create new knowledge for spatial reasoning tasks? *arXiv*, 2024.

David Heath and Derek Allum. The historical development of computer chess and its impact on artificial intelligence. In *AAAI Conference of Artificial Intelligence*, 1997.

Dongyoon Hwang, Hojoon Lee, Jaegul Choo, Dongmin Park, and Jongho Park. Can Large Language Models Develop Strategic Reasoning? Post-training Insights from Learning Chess. *arXiv*, 2025.

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*, 2025.

Mu-Tien Kuo, Chih-Chung Hsueh, and Richard Tzong-Han Tsai. Large Language Models on the Chessboard: A Study on ChatGPT's Formal Language Comprehension and Complex Reasoning Skills. *arXiv*, 2023.

Changmao Li and Jeffrey Flanigan. Task contamination: language models may not be few-shot anymore. In *AAAI Conference on Artificial Intelligence*, 2024.

Lichess.org. Lichess game database. https://database.lichess.org/, 2025. Public export of rated games, puzzles, and evaluations.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv*, 2025.

Anna Mészáros, Patrik Reizinger, and Ferenc Huszár. Out-of-distribution tests reveal compositionality in chess transformers. In *International Conference on Machine Learning*, 2026.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, August 2025.

OpenAI. What are tokens and how to count them?, 2025. URL `https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them`.

Gian-Carlo Pascutto and Gary Linscott. Leela Chess Zero, 2021. URL `http://lczero.org/`.

Zhenting Qi, Hongyin Luo, Xuliang Huang, Zhuokai Zhao, Yibo Jiang, Xiangjun Fan, Himabindu Lakkaraju, and James Glass. Quantifying generalization complexity for large language models. *arXiv*, 2024.

Mubashar Raza, Zarmina Jahangir, Muhammad Bilal Riaz, Muhammad Jasim Saeed, and Muhammad Awais Sattar. Industrial applications of large language models. *Scientific Reports*, 2025.

Eyal Reingold and Heather Sheridan. Chess expertise reflects domain-specific perceptual processing: Evidence from eye movements. *Journal of Expertise*, 2023.

Tord Romstad, Marco Costalba, Joona Kiiski, et al. Stockfish: A strong open source chess engine (version 17.1). `https://stockfishchess.org/`, 2008.

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 *International Conference on Neural Information Processing Systems*, 2024.

Jonathan Schaeffer and Aske Plaat. Kasparov versus Deep Blue: The Rematch. *ICGA Journal*, 1997.

Claude E. Shannon. Programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1950.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs Generate Novel Research Ideas? A Large-Scale Human Study with 100+ NLP Researchers. In *International Conference on Learning Representations*, 2024.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv*, 2017.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

Shu Wang, Lei Ji, Renxi Wang, Wenxiao Zhao, Haokun Liu, Yifan Hou, and Yingnian Wu. Explore the Reasoning Capability of LLMs in the Chess Testbed. *arXiv*, 2025a.

Xinyi Wang, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. Generalization v.s. Memorization: Tracing Language Models' Capabilities Back to Pretraining Data. In *International Conference on Machine Learning*, 2025b.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *arXiv*, 2022.

Junjie Wu, Mo Yu, Lemao Liu, Dit-Yan Yeung, and Jie Zhou. Understanding LLMs' fluid intelligence deficiency: An analysis of the ARC task. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2025.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. On memorization of large language models in logical reasoning. In *International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, 2025.

Yue Yang, MingKang Chen, Qihua Liu, Mengkang Hu, Qiguang Chen, Gengrui Zhang, Shuyue Hu, Guangtao Zhai, Yu Qiao, Yu Wang, Wenqi Shao, and Ping Luo. Truly assessing fluid intelligence of large language models through dynamic reasoning evaluation. *arXiv*, 2025.

V. B. Zakharov, M. G. Mal'kovskii, and A. I. Mostyaev. On Solving the Problem of 7-Piece Chess Endgames. *Programming and Computer Software*, 2019.

A Methodological details

A.1 Prompt

Positions were passed to the model in Forsyth-Edwards Notation (FEN). The following prompt was utilized to obtain model recommendations with `temperature = 0`, with `{fen}` being filled in with the position-specific FEN-string.

INPUT:

Consider the following chess position, denoted as a FEN string: `{fen}`

TASK:

Recommend the best move to play in the provided position and state your reasoning in the required format.

OUTPUT REQUIREMENTS:

You must return JSON only (no extra text, no code fences, no comments). Return a single JSON object matching this schema:

```
{
    move_uci: <Your move recommendation in UCI format, e.g. e2e4>,
    move_san: <Your move recommendation in SAN-Format, e.g. e4>,
    reasoning: <Provide a concise rationale why you decided for
        this move; no more than 3 sentences>
}
```

IMPORTANT INFORMATION:

- Your performance is scored on format correctness and recommendation quality.
- Not providing a legal chess move in the given position is considered a failed attempt.
- Only return JSON. Not precisely following the output requirements is considered a failed attempt.
- You are strictly forbidden from accessing external resources, like conducting a web-search or using other tools. This, too, will be considered a failed attempt.

A.2 Coverage study

To empirically assess and challenge our theoretical assumptions about the relationship between game progress and database frequency, we simulated 50 chess games. Each game was generated by iteratively selecting one of the three highest-rated engine moves for both Black and White using Stockfish [Romstad et al., 2008]. For every resulting position, we queried the Lichess masters database to determine whether it had previously occurred [Lichess.org, 2025]. The Lichess masters database contains more than 2.7 million games played since 1952 of players with FIDE ratings of 2,200 or more. In addition, we tracked the number of pieces remaining on the board, since modern tablebases are known to include all possible positions with seven or fewer pieces [Zakharov et al., 2019]. We therefore consider a position as covered—that is, potentially solvable via memorization—if it occurs in the Lichess Masters database or contains at most seven pieces. We then computed database coverage rates as a function of move number.

Our results, displayed in Figure 5, show that coverage is initially high, drops rapidly as the game progresses, and then rises again in later stages—beginning around move 49—as the number of pieces begins falling below eight in some games. We thus conclude that the training data coverage likelihood is high in very early and very late game stages, gradually decreases towards the mid-game and reaches its minimum in the late mid-game, around move thirty, when there are approximately 20 pieces on the board.

Derivation of condition design principles. Beyond validating our theoretical assumptions, the coverage study also aimed to inform the derivation of design principles for constructing positions with varying levels of familiarity. Our findings were threefold:

1. **WD positions.** High coverage rates can be found in early and certain late game positions. Yet, coverage in late-game positions solely relies on tablebases, where each position occurs once. As such, even if tablebases are part of the training corpora, late-game positions were likely seen few times. In contrast, early-game positions occur thousands of times within the Lichess database, and likely even in other sources such as opening books. Thus, early-game positions exhibit the highest likelihood of frequent and consistent coverage. We therefore obtained WD samples by performing a structured breadth-first search (BFS) starting from the initial chess position. We retained only positions that appeared in at least $1{,}000$[1] games, thereby maximizing both early occurrence and empirical frequency.
2. **ND positions.** The empirical finding of novel games beyond move nine in our case study suggested that generating positions by playing ten random moves per side from the starting position yields positions that are not memorized explicitly but remain structurally related to early-game configurations. Importantly, in some modern main lines, moves are known more than thirty moves deep. To ensure that the ND position is not part of one of those well-known main lines, we excluded positions which were found within the Lichess database. ND positions preserve key strategic features—such as central control, typical pawn structures, and early piece activation—allowing for the application of known heuristics.
3. **OOD positions.** To minimize structural similarity to WD configurations, we generated positions via random piece placement rather than random play. Based on our analysis, positions with no database coverage contained on average 20.25 pieces. We therefore placed ten pieces per side when constructing OOD positions.

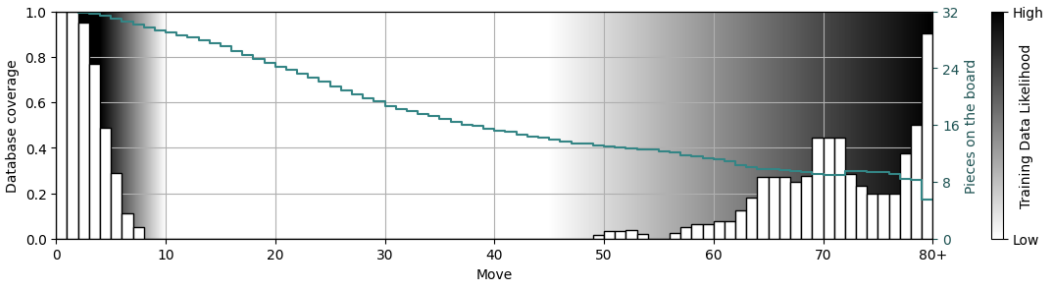


Figure 5: Lichess Masters database coverage by move number.

A.3 Within-distribution dataset generation

To construct the within-distribution dataset, we systematically extract 500 established opening configurations from the Lichess Masters database via a bounded breadth-first search (BFS) rooted at the standard initial chess position. State representations are tracked using Forsyth-Edwards Notation (FEN) and a hash set to resolve transpositions. For each dequeued state, we compute its total historical frequency. If a state appears at least $1{,}000$ times, it is appended to the dataset, and its legal continuations are enqueued for further evaluation. Crucially, if a state fails to meet this threshold, it is discarded and its branch is strictly pruned, preventing the exploration of low-frequency variations. The search terminates immediately upon reaching 500 distinct positions, ensuring the final dataset comprises logically contiguous, highly familiar theoretical lines.

A.4 Out-of-distribution dataset generation

To generate the OOD ($\mathcal{X}_{\text{OOD}}$) dataset, we employ a randomized piece placement algorithm utilizing the `python-chess` library. The objective is to construct board states that are syntactically valid under

[1]We selected a threshold of $1{,}000$ based on an empirical sensitivity analysis. Lower thresholds (e.g., 100) introduced significant noise from under-studied positions, while more restrictive thresholds (e.g., 10,000) limited the sample size.

the formal rules of chess but structurally devoid of typical strategic patterns, making their presence in the models' training corpora highly unlikely.

The generation process iteratively attempts to build positions satisfying several strict constraints. First, regarding king placement, the White and Black kings are placed on two distinct, uniformly sampled squares. Following this, we address piece distribution by placing exactly 10 additional pieces per side, allowing overwrites. The piece types are sampled with weights reflecting standard starting frequencies while discounting pawn probability to account for frequent early-game pawn trades and prevent closed structures, ensuring piece mobility (Pawns: 4, Knights: 2, Bishops: 2, Rooks: 2, Queens: 1). We strictly enforce maximum piece counts corresponding to a standard chess set, ensuring there are no more than 8 pawns or 1 queen per side.

To guarantee positional validity, the board state is verified using `python-chess` internal validation routines, ensuring fundamental legality such as confirming the side not to move is not currently in check. We immediately discard any position that is already in a terminal state, such as checkmate or stalemate. Finally, for database filtration, the generated FEN string is queried against the Lichess database. The position is solely accepted into the final dataset if it has exactly zero historical occurrences.

A.5 Leela Chess Zero Configuration

We utilized a locally deployed variant of the Lc0 network `t3-512x15x16h-distill-swa-2767500`. It represents a medium-sized configuration designed to balance performance and efficiency across both GPU and CPU settings. It employs 512 convolutional filters and 15 residual blocks, yielding a memory footprint of approximately 1.8 GB and a compact file size of around 150-155 MB. As a distilled model with stochastic weight averaging (SWA), it captures much of the strength of larger networks while remaining more accessible for resource-constrained environments. This positioning makes it a practical choice for experiments requiring strong playing strength without the computational overhead of very large models [Pascutto and Linscott, 2021].

B Example positions

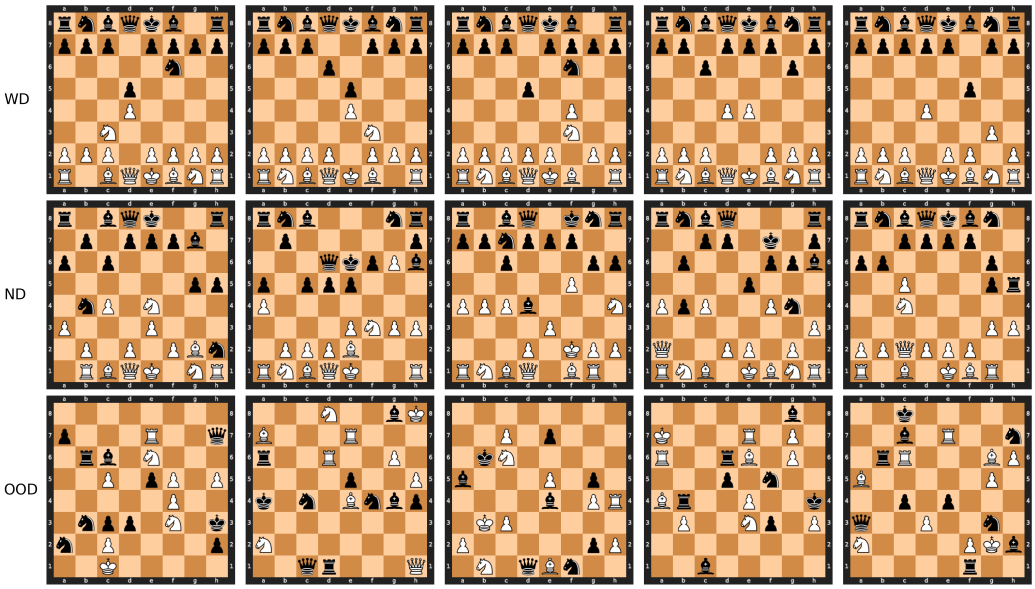


Figure 6: Example positions by condition. WD = within-distribution, ND = near-distribution, OOD = out-of-distribution.

We subsequently discuss the example positions displayed in Figure 1 to contextualize our quantitative results.

WD This is the Ruy López Opening, with black to play. Stockfish recommends ...Nf6, indicating a slight advantage for white (+0.28). GPT-3.5 tries to play ...Qc8, which is illegal. Lc0, GPT-4o and GPT-5 (moderate reasoning) all recommend ...a6, a well-known theoretical continuation known as the Morphy Defense, incurring a negligible CPL of 13. GPT-5 with minimal reasoning recommends Nd4, also following a known theoretical line called the Bird's Defense. It is not considered optimal by modern standards, incurring a CPL of 43. The random agent selects ...Nf6, matching Stockfish's choice.

ND White is to play. While there are many decent moves in the position, Stockfish recommends Bb2, developing the bishop. After Bb2, white is better (+3.38). Lc0 recommends Bxa6, taking the knight, incurring a CPL of 1. GPT-3.5 recommends to play e4, incurring a CPL of 713, effectively losing the game: After ...exd4, white loses a piece. GPT-4o recommends fxe5, incurring a CPL of 31. GPT-5 (minimal reasoning) tries to play cxd5, which is illegal. GPT-5 (moderate reasoning), like Lc0, recommends Bxa6, also incurring a CPL of 1. The random recommendation is Ba3, incurring a CPL of 88.

OOD White is to play. Stockfish recommends Kb6 to get out of check. After Kb6, white is better (+3.24), as black cannot stop white from promoting the b7 pawn. Lc0 finds this move, incurring a CPL of 0. GPT-3.5 recommends illegally promoting the pawn via b8=Q, ignoring that white is currently in check. GPT-4o recommends b7g7+. This is non-standard algebraic notation, incurring an illegal move. GPT-5 (minimal reasoning) recommends playing Nd4, also ignoring that white is in check, incurring an illegal move. GPT-5 (moderate reasoning) recommends Nxd6, recognizing the check. While legal, it effectively loses the game to ...Qf2+, as white will no longer be able to promote its pawn, incurring a CPL of 992. The random recommendation is also Nxd6, incurring a CPL of 992.

We provide further example positions in Figure 6. A study of the generated positional categories reveals that many positions, especially within the ND and OOD setting, are unlikely to occur in games between strong players. This systematic difference to practical positions is intentional. We explicitly aim to break known patterns, especially in OOD positions. In doing so, we aim to minimize the utility of basic pattern recognition and associative recall. Thereby, we decrease the likelihood of obtaining strong solutions using memorized game knowledge like opening theory, databases, or standard mid- or end-game strategy, aiming to isolate genuine reasoning capability.

C Framework validity and scope

Our experimental framework rests on several foundational characteristics: the reference world defined by chess engines, the construction of position types, the approximation of training-distribution proximity, the choice of evaluation metrics, the input encoding, the observational nature of the design, and the absence of a human baseline. Each introduces specific constraints and abstractions. We explicitly delineate the scope of our claims below to address potential validity concerns regarding the use of chess as a proxy for general reasoning.

C.1 The reference world: abstracted engine chess

It is critical to note that the reference world for this study is not the sport of chess as played by humans, but rather a reduced, abstracted world defined by chess engines. Stockfish and similar engines (including the neural-network-based Lc0) are not intelligent in the human sense. They are highly optimized, heuristic search algorithms rooted in massive calculation [Romstad et al., 2008, Pascutto and Linscott, 2021]. They do not understand chess strategy as a narrative; they solve it as a logic puzzle. Therefore, when we measure LLM performance against an engine, we are measuring the model's ability to align with a computational optimum, not its ability to exhibit human-like intuition or creativity.

While engines are not ground truth solvers of chess (which remains unsolved), they exceeded human capability decades ago [Schaeffer and Plaat, 1997]. For the purpose of evaluating LLM errors—which often involve fundamental misunderstandings of board state or rules—the engine serves as a sufficiently high-resolution ruler. A high CPL objectively represents a failure to optimize within the game's logical constraints, regardless of the human quality of the move. Consequently, while engine evaluations may not capture the full semantic nuance of human strategy, they provide a necessary

and rigorous baseline for assessing the tactical soundness of the models' reasoning. Thus, within the scope of this study, we accept engine evaluations as a robust heuristic for distinguishing between valid reasoning and retrieval failure, without claiming they represent a ground truth.

C.2 Nature of OOD positions

Our OOD positions are generated via random placement to maximize distance from the training manifold. This generation process introduces specific structural biases that warrant clarification: proficient human play relies on a synergy of *strategy* (often termed *positional play* in domain literature, focusing on long-term planning and static evaluation) and *tactics* (immediate calculation of concrete variations). Due to their randomized nature, our OOD positions often assess tactical ability rather than strategical prowess. They frequently lack standard pawn skeletons, or King safety norms typical of human games.

While unnatural relative to human play, this tactical focus is a direct consequence of isolating generalization rather than an arbitrary design choice. In chess, familiar patterns are carried predominantly by pawn structures, which form the skeleton of a position and anchor most strategic knowledge. Strategic positions therefore almost always structurally resemble known games, making them highly susceptible to memorization. Breaking familiar patterns requires departing from typical pawn structures—in our generator, by down-weighting pawns (Appendix A.4)—which in turn renders the resulting positions predominantly tactical. By requiring the model to solve positions that break standard heuristic patterns, we test its ability to apply the rules of the game and derive a solution from the concrete position. In a formal system like chess, when pattern matching is precluded, calculation becomes the primary proxy for fluid reasoning. Accordingly, our claims about generalization primarily concern tactical calculation and rule application rather than long-term strategic planning, and a performance collapse on these positions indicates a limitation in exactly this capacity.

This limitation stands in stark contrast to human capabilities. While human players also rely heavily on structural chunking and heuristic intuition—which are largely inapplicable in OOD positions—they retain the ability to consistently identify legal moves and calculate local tactics in *any* position. The LLMs' failure in OOD settings, highlighted not just by a very high ACPL but by a catastrophic spike in illegal moves, demonstrates an inability to fall back on fundamental rule adherence. This underscores a critical deficit in the foundational components of formal fluid reasoning.

C.3 Approximation of Training Data Proximity

The taxonomy employed (WD, ND, OOD) relies on an approximation of the unobserved training distribution. We utilized the Lichess Masters database to approximate the training distribution. We acknowledge that unlike curated professional databases (e.g., ChessBase), public internet databases often contain theoretically irrelevant positions. To mitigate this noise, we enforced a strict threshold of $n_{min} = 1{,}000$ occurrences for the WD set. While this specific cutoff is heuristic, it serves as a robust filter against idiosyncratic games, ensuring that WD positions are not merely present in the database, but are statistically established standard positions likely to be over-represented in the LLM's massive training corpora.

We cannot formally guarantee that all positions categorized as WD were present in the LLM training corpus, nor strictly preclude the appearance of OOD positions. However, given the combinatorial explosion of the state space, the probability of a specific, randomly generated position with high piece count appearing in the training data is negligible. Even if the categorization is imperfect, this uncertainty biases our results conservatively: Misclassification would act as a form of label noise between conditions. OOD positions that were in fact seen during training would, under the plausible assumption that exposure does not impair performance, raise OOD performance toward WD levels. Conversely, WD positions that were in fact absent would lower WD performance toward OOD levels. Both effects attenuate the measured difference between conditions. We therefore hypothesize that the reported results likely represent an upper bound on the model's true fluid reasoning capability, which may be even more limited than observed. We emphasize that this is a reasoned hypothesis rather than an established fact: it cannot be verified empirically without access to proprietary training corpora, and it relies on the assumption that exposure to a position does not systematically harm performance.

C.4 Choice of evaluation metrics

We employ CPL—a measure of decision regret relative to an engine's optimal move—as our primary metric. Critically, we argue that outcome-based metrics are insufficient for measuring performance in this context, as they fail to distinguish between a reasoning process that is logically sound but imperfect, and one that is effectively random. However, as engines do not exhibit perfect play, we explicitly acknowledge that CPL is not a flawless descriptor of overall playing strength, particularly in strategic contexts where sub-optimal moves may result in long-term positional disadvantages that may exceed the evaluation depth of contemporary engines. Our choice is nonetheless motivated by the fact that alternative methods like win/draw/loss probability do not sufficiently differentiate move quality in positions that are objectively decided. This saturation masks qualitative differences in play. Conversely, CPL provides a continuous measure of gradual error magnitude. It distinguishes between a sub-optimal move that retains the current balance (e.g., CPL 15) and a blunder (e.g., CPL 300), even in imbalanced positions. This granularity is essential for quantifying the subtle gradients of reasoning capability across model generations that would otherwise remain invisible under binary evaluation criteria.

We recognize that CPL is a non-linear metric and its magnitude is sensitive to the absolute evaluation of a given position. Because our OOD positions are generated randomly, they may feature material imbalances from the outset, which can inflate or deflate CPL penalties for sub-optimal moves. However, given the sample size within each condition, we expect these localized positional fluctuations to average out across the dataset, yielding a representative aggregate measure. Nevertheless, to further mitigate the risk of over-interpreting CPL variance, we consider the ACPL in conjunction with the rate of illegal moves. The concurrent, severe degradation of both metrics in OOD contexts confirms a fundamental breakdown in positional reasoning, independent of CPL non-linearity.

C.5 The interplay of tokenization and generalization

The observed performance degradation in OOD scenarios may intersect with the input-encoding mechanisms of current LLM architectures, specifically Byte-Pair Encoding (BPE) tokenizers. Board states in our study were provided via FEN strings. Because modern tokenizers optimize for subword frequencies, standard WD FENs may be compressed into familiar, semantically rich tokens due to their high prevalence in the pre-training data. Conversely, the highly atypical FENs generated for our OOD dataset could force the tokenizer to fracture the input into unlearned, character-level fragments.

While this modality mismatch could be viewed as a perceptual bottleneck rather than a purely strategic failure, it fundamentally reinforces our core thesis regarding the limits of generalization in LLMs. The reliance on familiar subword tokenization to parse a strictly logical, rules-based environment is itself a manifestation of over-reliance on crystallized patterns. A system possessing robust fluid ability should be capable of falling back on strict syntactic and character-level deduction when familiar heuristic chunks are unavailable. The fact that the base models instead suffer a catastrophic breakdown—evidenced by the spike in illegal moves—demonstrates that their reasoning capabilities are inextricably tethered to the recognition of memorized input distributions. Future work could isolate the exact magnitude of this tokenization effect by evaluating models using direct spatial encodings, though our findings strongly suggest that escaping this crystallized dependency will require architectural mechanisms beyond current autoregressive scaling.

C.6 Observational design and the limits of causal attribution

We emphasize that our study is observational and does not constitute a randomized controlled trial: prior density is not manipulated independently of the structural properties of the board. Assignment is instead determined by the position-generating process, with the consequence that our conditions differ jointly in estimated prior density, but also in material configuration, pawn structure, board topology, game phase, branching factor, and tactical character. We stress that this is not a control we omitted, but one that the domain itself precludes. Our coverage study (Appendix A.2) demonstrates that structural properties and database coverage are themselves correlated. We argue that constructing positions that are simultaneously low in prior density and structurally indistinguishable from high-prior-density positions is not possible, because *structural atypicality is constitutive of what it means for a chess position to fall outside the human game corpus*. A genuine randomized manipulation would require intervening on the pre-training corpus itself—precisely the object that is inaccessible

for proprietary frontier models, and the inaccessibility that motivates our proxy-based design in the first place. Controlling for structural confounds is only feasible employing small-scale, self-trained architectures, at the cost of forfeiting any analysis of frontier-scale behavior. We scope our claims to reflect this limitation: we document a robust association between estimated prior density and performance, but we do not claim to identify a causal effect of prior density net of all co-varying positional properties.

Three features of our design nonetheless constrain the space of alternative explanations. First, the ND condition is absent from the database yet structurally continuous with early-game play, and already exhibits substantial, graded degradation relative to WD, which is difficult to reconcile with an account in which only extreme structural unnaturalness drives the effect. Second, the rules that determine legality are identical in every position. Legality determination is a bounded procedure whose difficulty does not increase in our OOD positions. Third, the degradation in legality persists across output notations (Appendix G.3) and under a roughly fivefold increase in reasoning tokens. Jointly, these observations render a purely structural account of our findings difficult to sustain, even though they do not constitute causal identification in the strict experimental sense. In particular, we interpret the legality results as evidence that the models fail to robustly apply the rules of chess in unfamiliar positions, rather than as proof that memorization and generalization have been fully separated.

C.7 Legality, move quality, and the absence of a human baseline

A natural objection to our interpretation is that humans, too, may perform worse on unfamiliar positions. Human chess expertise relies heavily on chunking, i.e., the recognition of familiar configurations of pieces [Chase and Simon, 1973], and such chunks are largely unavailable in our OOD positions. A human facing a randomized board would therefore plausibly require more time and might select weaker moves. We argue that this objection applies with very different force to our two metrics, and we scope our claims accordingly.

Move quality. Finding a strong move in an unfamiliar position plausibly requires more deliberation from any agent. Without a human baseline on our specific positions, we cannot rule out that time-constrained humans would also show some degradation in move quality on OOD positions. We therefore regard the ACPL results as *consistent with*, but not conclusively isolated from, a generic difficulty-under-novelty effect.

Rule adherence. Legality differs categorically from move quality. The rules that determine whether a move is legal are fixed and identical across all positions. Applying them requires reconstructing the board state, but this procedure is fully specified, bounded, and requires no strategic judgment or pattern knowledge. An agent that has internalized the rules should therefore become slower, but not less rule-compliant, on unfamiliar positions. Evidence from a simplified setting is consistent with this expectation for humans: Reingold and Sheridan [2023] tested expert and novice human players on a double-check detection task, rendered either in the familiar symbol notation or in an unfamiliar letter notation. Error rates remained below 1% for both groups in both notations. The unfamiliar notation slowed responses but did not reduce accuracy, indicating that human players retain their ability to adhere to a fixed rule set, even under aggravated circumstances.

By contrast, the LLMs in our study propose illegal moves at sharply increasing rates as positions become less familiar, and additional reasoning does not eliminate this failure: GPT-5 with moderate reasoning still proposes illegal moves in more than 10% of OOD positions despite using roughly five times as many reasoning tokens as in WD positions. Because verifying move legality is a bounded procedure, additional deliberation should suffice to resolve it, as additional time would for a human player. We therefore regard the illegal-move results as the more robust evidence that current LLMs fail to apply the rules of chess systematically outside familiar configurations.

A dedicated human study on our position set would provide a direct test of both claims and constitutes a valuable avenue for future work.

D Extended related work

This appendix contrasts our approach with closely related work at the intersection of language models and chess.

Ruoss et al. [2024] introduce ChessBench, a large-scale dataset for chess, and train specialized transformers using supervised learning on Stockfish evaluations. While they do not reach superhuman performance, they demonstrate that playing strength scales with model and dataset size. Our work differs from theirs in several critical ways. Ruoss et al. [2024] train a specialized transformer via supervised learning on Stockfish-generated labels to maximize overall chess performance. In contrast, we evaluate general-purpose frontier LLMs zero-shot to disentangle memorization from reasoning. Further, their evaluation relies on data derived from human play, meaning the models are tested on naturally occurring board states. In this regime, pattern memorization remains an extremely effective heuristic. By limiting their evaluation to the region of the chess state space frequently encountered in human play, their setup differs from ours. Given their findings, it would be a highly interesting avenue for future research to evaluate their specialized models on structurally novel OOD positions to observe if extensive chess-centric training helps mitigates the generalization gap we observed in frontier LLMs.

Wang et al. [2025a] introduce MATE, a dataset of approximately one million positions from chess games and puzzles in which candidate moves are annotated with explanations of long-term strategy and short-term tactics. They fine-tune LLaMA-3-8B on MATE and show that the fine-tuned model outperforms commercial LLMs, including GPT, Claude, and Gemini models, and that language explanations improve this ability. Their setting differs from ours in that models are fine-tuned on chess data, the task is a choice between provided candidate moves (which removes the need to generate legal moves), and positions stem from human play and puzzles. We instead evaluate unconstrained, zero-shot move generation across conditions that differ in the density of relevant priors.

Hwang et al. [2025] investigate whether LLMs can develop strategic reasoning through reinforcement learning on chess, using a chess-pretrained action-value network to provide dense rewards on move quality, a form of knowledge distillation. All models plateau far below expert level, which the authors attribute to a deficit in the pretrained models' internal understanding of chess that reinforcement learning alone may not overcome. This conclusion is complementary to ours: we characterize the zero-shot competence of frontier models directly and find that it is concentrated in regions of the state space that are densely covered by likely training data.

Kuo et al. [2023] study ChatGPT's chess abilities using metrics for both the legality and the quality of its moves, and report limitations in its comprehension of the formal board representation, identifying rule adherence as a central weakness. We extend this line of work by showing that rule adherence degrades systematically as positions become less familiar, and that this pattern persists across three GPT generations, two further model lineages, and reasoning-augmented inference.

Kolasani et al. [2025] introduce LLM CHESS, an agentic benchmark in which more than 50 models play complete games against a random opponent and can retrieve the board state and the list of legal moves through tool calls. Elo estimates are derived from games against a chess engine for a subset of models. The authors report that state-of-the-art models struggle to complete games or to win consistently, and reasoning models separate clearly from non-reasoning models. In contrast to our work, their benchmark measures end-to-end agentic play rather than performance as a function of positional familiarity.

E Societal impact

This research highlights a generalization gap in LLMs within the domain of chess, suggesting that their impressive performance may stem more from extensive pattern recall than from adaptive, first-principles reasoning. By demonstrating that model reliability can decline when encountering novel configurations, these findings encourage a cautious approach to deploying LLMs in specialized or rapidly evolving fields where historical data may not be a perfect guide.

F Hardware specification

All experiments were performed on a 2024 MacBook Air with an Apple M3 processor.

G Additional analyses

G.1 Centipawn loss distributions

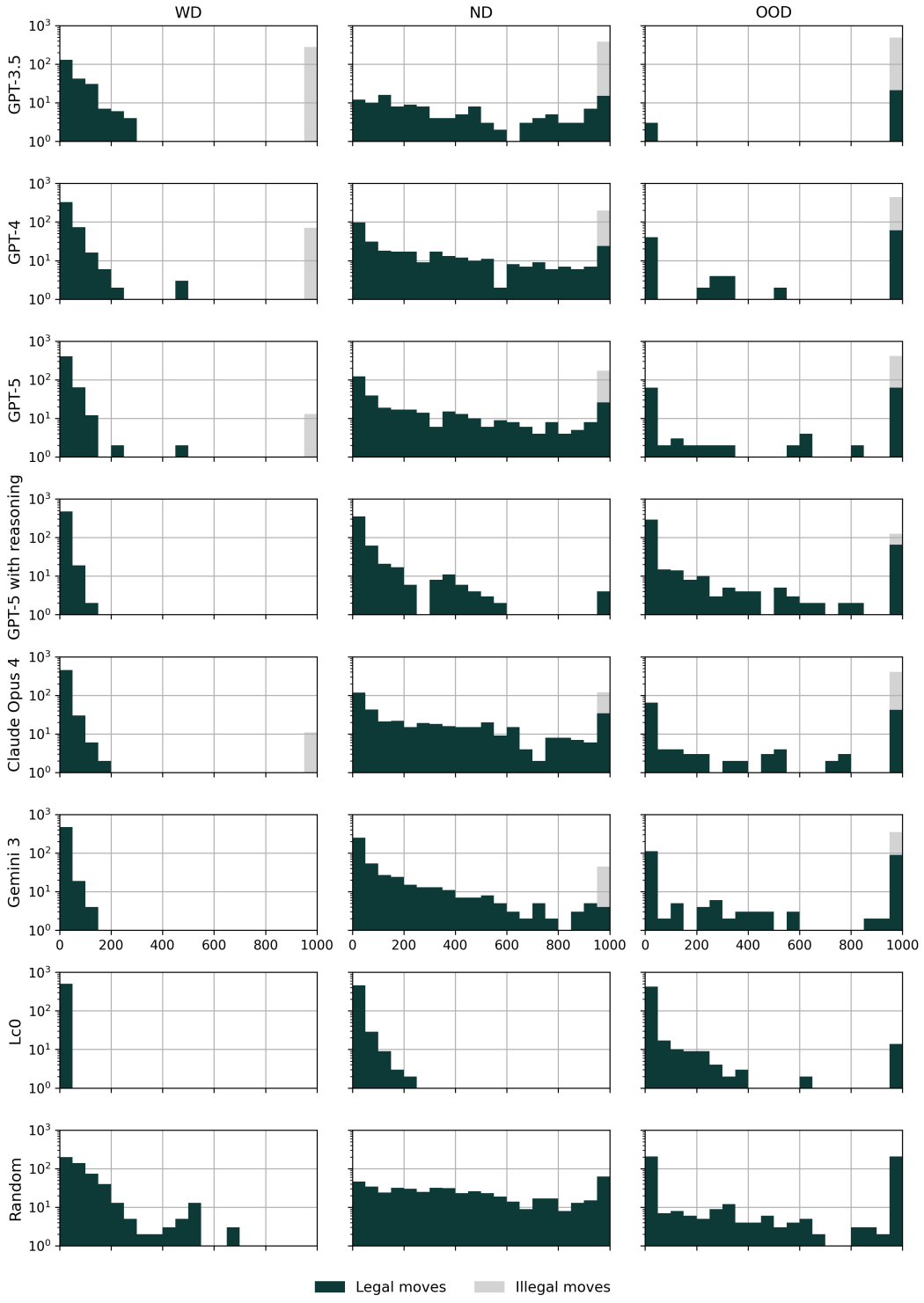


Figure 7: Centipawn loss distributions by model and condition. WD = within-distribution, ND = near-distribution, OOD = out-of-distribution.

G.2 Split diagnostics

Table 2 characterizes the three conditions. WD and ND positions retain near-complete and balanced material, whereas OOD positions contain substantially less material and exhibit a much larger variance in the number of legal moves.

Table 2: Split diagnostics (mean $\pm$ SD over 500 positions per condition). Material in pawn units (P = 1, N = B = 3, R = 5, Q = 9; kings excluded). Legal moves are counted for the side to move.

Condition	Legal moves	White material	Black material
WD	26.27 ± 4.42	38.98 ± 0.13	38.99 ± 0.11
ND	31.21 ± 7.68	37.76 ± 2.20	37.84 ± 2.30
OOD	19.75 ± 18.85	24.36 ± 4.69	24.99 ± 4.74

G.3 Robustness to the output notation

Each model returned its recommendation in both SAN and UCI notation within the same response (Appendix A.1). Table 3 reports results when moves are evaluated in UCI instead of SAN. All qualitative findings replicate: for every model and reasoning setting, the rate of legal moves is lowest and the ACPL highest in OOD positions, both including and excluding illegal moves; successive GPT generations improve, with diminishing gains in the OOD condition; and moderate reasoning improves performance in all conditions, while GPT-5 with moderate reasoning still proposes illegal moves in 12.4% of OOD positions.

Table 3: Results under UCI output notation. ACPL including illegal moves assigns a loss of $1{,}000$ to illegal moves; ACPL excluding illegal moves is computed over legal moves only.

Model	Condition	Legal (%)	ACPL (incl. illegal)	ACPL (excl. illegal)
GPT-3.5	WD	37.2	648.17	54.22
	ND	17.6	893.35	394.05
	OOD	4.2	989.46	749.05
GPT-4o	WD	85.6	174.52	35.65
	ND	72.0	514.16	325.22
	OOD	32.4	849.37	535.08
GPT-5 (minimal)	WD	97.0	55.53	26.32
	ND	79.0	440.76	292.10
	OOD	35.0	838.03	537.23
GPT-5 (moderate)	WD	100.0	9.29	9.29
	ND	100.0	69.54	69.54
	OOD	87.6	286.80	185.84

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All main claims made (performance gradient from crystallized to fluid intelligence; improvement across model generations; slowing progress, especially for fluid intelligence tasks; decreased marginal gain per token for fluid intelligence) are supported by at least one Figure in the main body.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We acknowledge the limitations of our work in the Discussion section, and thoroughly discuss them in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper exhaustively explains the construction rules for all conditions, the prompt, and all model specifications, enabling an exact reproduction of the experimental setup. Our code and the dataset will be open-sourced.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Data and code are open-source; we will add the repository link to the camera-ready version.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We thoroughly describe our experimental setup in the Methodology section. We herein and within the Appendix disclose all details required to understand the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report 95% confidence intervals or standard deviations alongside all reported means.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: As disclosed in Appendix F, all numerical experiments were performed on a 2024 MacBook Air with an Apple M3 processor. As the experiments rely on proprietary APIs to access state-of-the-art LLMs, we are not able to provide exact information on the required amount of compute.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research does not conflict with the NeurIPS Code of Ethics in any way.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the societal impacts of our work in Appendix F.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper proposes a chess-based evaluation framework, which bears no risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All owners of utilized assets are credited and all terms of use are respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our code and the evaluation dataset are open-sourced under a CC-BY 4.0 licence. We will add the repository link to the camera-ready version.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The usage of LLMs to obtain best-move suggestions across a variety of chess positions is extensively discussed in the main body. Beyond this use case, the core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.