MASTERMINDEVAL: A Simple But Scalable Reasoning Benchmark

Anonymous ACL submission

Abstract

Existing reasoning benchmarks are often static, hard to extend, and saturate quickly, which limits their use for tracking progress of LLM reasoning capabilities. We introduce MASTERMINDEVAL, a simple and scalable reasoning benchmark based on the board game Mastermind where a player needs to deduce the secret code from a solution state space by iteratively guessing and understanding combining the feedback across guesses. Since we can control the solution state space by two parameters, the code length and the number of symbols, game instances can be generated at arbitrary complexity and our benchmark stays representative as models improve. MASTERMINDEVAL supports three evaluation paradigms: *(i)* an *agentic* setting where the model plays the full game across multiple turns, *(ii)* a *deductive reasoning* setting where the model is given a pre-played game state that uniquely determines the code and must infer it in one step, and *(iii)* a *log-likelihood* setting that ranks candidate codes without test-time reasoning. We find that smaller models (<7B) achieve at most a 0.30 solve rate without thinking on the standard configuration (4-digit code, 6 colors), while larger models (26B–35B) drop to 0.21 on a harder setting (5-digit code, 7 colors).

1 Introduction

Large language models (LLMs) have demonstrated remarkable performance across various text generation tasks, spanning both text and vision modalities (Yang et al., 2025; Liu et al., 2026; Olmo et al., 2026). While LLMs become increasingly capable at complex reasoning tasks, recent work suggests reported gains may partly reflect benchmark saturation and data contamination rather than genuine deductive reasoning (Sinha et al., 2019; Geva et al., 2021; Dziri et al., 2023; Wu et al., 2025; Zhou et al., 2025; Bordt et al., 2025; Hasan et al., 2026).

Board-game-based benchmarks derived from Chess (Zhang et al., 2025), Tic-Tac-Toe or Breakthrough (Parmar et al., 2024; Duan et al., 2024; Yao et al., 2025) are appealing in this context, as gameplay requires forming strategies rather than recalling memorized answers, making them more robust to contamination (Schultz et al., 2025). However, existing benchmarks have two key limitations: First, most rely on two-player settings, where the model's reasoning is entangled with opponent modeling. Second, the underlying games typically do not allow difficulty to be scaled easily, limiting their usefulness for tracking progress across model generations.

To address these limitations, we propose MASTERMINDEVAL, a benchmark derived by the game of Mastermind, a logical deduction puzzle in which the player must infer a concealed symbolic code based on feedback from previous guesses. Our benchmark provides three distinct evaluation methodologies: *(i)* an *agentic evaluation*, where the LLM plays the game in full multi-turn interactions, *(ii)* a *deductive reasoning evaluation*, where the LLM must infer the only valid remaining code from the solution state space using feedback in a pre-played game, and *(iii)* a *multiple-choice evaluation*, in which the model ranks candidate codes by log-likelihood, thus performing deductive reasoning purely through its pretraining objective without relying on test-time compute (Snell et al., 2024; Muennighoff et al., 2025). We illustrate the first two variants in Figure 1.

We summarize our contributions as follows:

1. We introduce MASTERMINDEVAL, a reasoning benchmark based on the game of Mastermind and propose three distinct evaluation methodologies for assessing language model performance using our framework.

2. We conduct comprehensive evaluations, demonstrating that our benchmark is scalable and highlighting the current limitations of

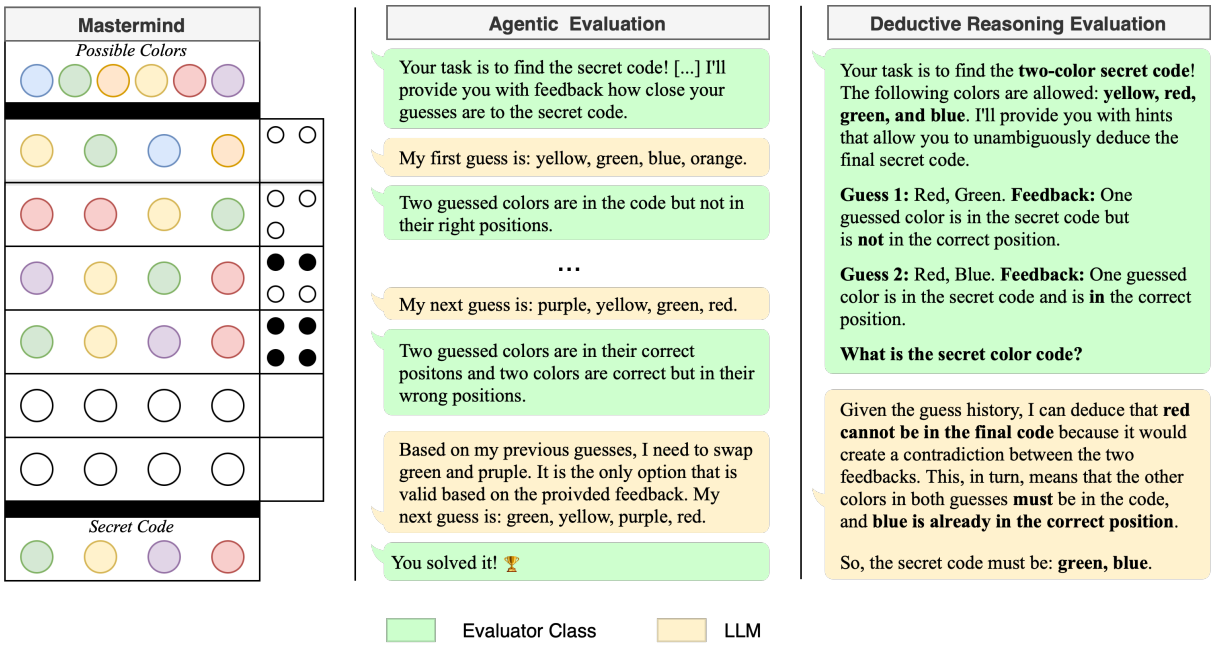

Figure 1: *(Left)* A Mastermind game instance with a secret code of length 4 and 6 possible colors, along with corresponding feedback for each guess (black and white pegs). *(Center)* Our agentic evaluation, where the LLM acts as the codebreaker in a multi-turn chat environment. *(Right)* Using Knuth's five-guess algorithm, we pre-play games until only one valid code remains, then prompt the model to deduce the final answer based on the provided hints.

models in combining multiple pieces of information.

3. We open-source our code and datasets, providing the research community with a scalable reasoning benchmark and potential training datasets to improve reasoning capabilities [1].

2 MastermindEval

2.1 Gameplay

In the game of Mastermind, the codebreaker attempts to deduce a hidden sequence of symbols by making successive guesses and receiving feedback. The game is defined by three key parameters: the code length c, the number of possible symbols n, and the allowed number of guesses g. A standard game configuration consists of a four-symbol code ($c = 4$) chosen from a set of six possible symbols ($n = 6$), with repetitions allowed, resulting in $6^4 = 1,296$ possible codes.

If the codebreaker's guess is (x_1, x_2, x_3, x_4) and the hidden code is (y_1, y_2, y_3, y_4), the player receives feedback indicating the accuracy of their guess based on two criteria:

1. The number of symbols that are in the correct position, i.e., the number of indices j such that $x_j = y_j$ (referred to as "black hits").

2. The number of symbols that are present in the code but in the wrong position, i.e., the number of indices j where $x_j \neq y_j$ but $x_j = y_k$ for some k, provided that y_k has not already been counted as a black hit (these are called "white hits").

As stated in Bonnet and Viennot (2016), this evaluation process can be formalized as a grading function, which is a symmetric function taking two codes as input and returning a pair of integers (b, w) as output. Given an instance (c, n) of the Mastermind game, a secret code s, and a guess g, the number of black and white hits is computed as:

$$b = \sum_{i=1}^{c} \delta(s_i, g_i),$$

$$w = \max_{\tilde{g} \in \text{Perm}(g)} \left(\sum_{i=1}^{c} \delta(s_i, \tilde{g}_i) \right) - b.$$

where δ denotes the Kronecker delta function, which returns 1 if $s_i = g_i$ and 0 otherwise. $\text{Perm}(g)$ denotes the set of all permutations of g.

[1] Removed for double-blinded review.

The objective of the codebreaker is to achieve c black hits by iteratively refining guesses based on the feedback. This iterative process allows the player to eliminate incorrect possibilities and progressively narrow down the possible solution state space. In Figure 1 (deductive reasoning view, right side), an example illustrates how the feedback can reduce the set of possible codes until only one valid solution remains. The challenge is to identify this solution within the allotted number of guesses g.

2.2 Evaluation Paradigms

2.2.1 Agentic Evaluation

In this setup, the LLM acts as the codebreaker and must deduce the secret code through multi-turn reasoning, continuing until it identifies the code or the maximum number of allowed guesses is reached. This paradigm assesses two capabilities: *(i)* deducing the code from prior feedback, and *(ii)* selecting guesses that progress toward the solution efficiently. For example, classical algorithms for Mastermind (Knuth, 1977; Kooi, 2005) maintain the set of candidate codes consistent with prior feedback and choose guesses that maximize worst-case information gain over the resulting partitions. Therefore, strong performance reflects a model's ability to eliminate inconsistent hypotheses, adapt to feedback, and progressively narrow down the solution space.

2.2.2 Deductive Reasoning Evaluation

This setting is a single-turn interaction in which the LLM must deduce the secret code from prior feedback where the prior feedback narrows down the solution space to exactly one valid remaining code. To construct this dataset, we use Knuth's algorithm (Knuth, 1977) across different game configurations and record guesses and feedback until the secret code can be uniquely determined. Since Knuth's algorithm maximizes information gain at every step, the model must integrate evidence across all guesses rather than exploit a single near-solution guess at the end.

Knuth's Five-Guess Algorithm. Knuth's algorithm (Knuth, 1977) solves the Mastermind game optimally in at most five guesses using the minimax strategy. Given a secret code s from a set $\mathcal{S}$, the algorithm iteratively selects guesses $g \in \mathcal{S}$ to minimize the worst-case number of remaining possibilities:

$$g^* = \arg\min_{g \in \mathcal{S}} \max_{\substack{s' \in \mathcal{S} \\ f(g,s')=f(g,s)}} |\mathcal{S}'|$$

where $f(g, s)$ denotes the generalized feedback function from Section 2.1, and $\mathcal{S}'$ represents the reduced solution space remaining after incorporating feedback $f(g, s)$. The process begins with an optimal first guess (e.g., $(1, 1, 2, 2)$) and continues until only a single valid code remains ($|\mathcal{S}| = 1$), ensuring a solution within at most five steps.

2.2.3 Multiple-Choice Evaluation

Finally, we derive a multiple-choice variant of the deductive reasoning benchmark, in which we rank candidate codes by log-likelihood using the pre-training objective. Unlike the previous settings, this setting does not allow models to leverage test-time compute, isolating their internalized reasoning capabilities. We construct wrong answer options that differ from the correct code by a single symbol and exclude any that appear among previous guesses, since these can be trivially ruled out.

Log-Likelihood Ranking. Given a prompt t describing the partially observed game and a candidate code $\mathbf{y}_i \in [n]^c$, we score each candidate by its conditional log-likelihood

$$\log \hat{p}(\mathbf{y}_i \mid t) = \log \hat{p}(\mathbf{y}_i, t) - \log \hat{p}(t).$$

Since t is shared across all candidates, ranking reduces to ranking by $\log \hat{p}(\mathbf{y}_i, t)$, which causal language models compute as the sum of token-wise log-likelihoods.

2.3 Technical Implementation

We implement a dedicated evaluator that encapsulates the Mastermind logic, parameterized by code length c, number of symbols n, and guess budget g. Each game starts by sampling a secret code uniformly at random from the n^c possible codes.

While the agentic setting can draw instances from the full code space, the deductive reasoning and multiple-choice variants require pre-generating games with Knuth's algorithm. We provide roughly 50k games for configurations up to $(c{=}4, n{=}6)$, and 5k (deductive reasoning) / 2k (multiple-choice) for larger ones, since Knuth's algorithm becomes substantially slower at higher difficulty (see Table 4). We discard games where the feedback does not uniquely determine the secret code, since the

Model	Game Config $(c = 2, n = 4)$	$(c = 3, n = 5)$	$(c = 4, n = 6)$	$(c = 5, n = 7)$	Avg.
Llama-3.1-8B	0.79±0.08	0.33±0.10	0.11±0.06	0.01±0.01	0.31
OLMo-3-7B	0.63±0.09	0.26±0.08	0.05±0.05	0.00	0.23
Qwen2.5-7B	0.73±0.09	0.18±0.08	0.04±0.04	0.00	0.24
Phi-4-14B	0.85±0.07	0.52±0.10	0.30±0.09	0.07±0.05	0.43
Qwen3-8B	0.95±0.04	0.65±0.09	0.25±0.09	0.07±0.05	0.48
w/ thinking	**1.00**±0.00	**1.00**±0.00	0.52±0.10	0.14±0.07	0.67
Qwen3.5-9B	0.98±0.03	0.68±0.09	0.21±0.08	0.06±0.04	0.48
w/ thinking	**1.00**±0.00	0.97±0.03	0.71±0.09	**0.27**±0.08	**0.74**
Gemma-4-E4B	**1.00**±0.00	0.59±0.09	0.25±0.09	0.05±0.05	0.47
w/ thinking	**1.00**±0.00	0.80±0.08	0.30±0.09	0.07±0.05	0.54
Gemma-4-26B	**1.00**±0.00	0.96±0.03	**0.79**±0.08	0.21±0.08	**0.74**
Qwen3.5-35B	**1.00**±0.00	0.79±0.08	0.48±0.10	0.21±0.08	0.62

Table 1: *(Agentic Evaluation)* Win rate of LLMs playing full multi-turn game of Mastermind across game configurations ($\pm$ 95 % CI via bootstrapping with $N = 10{,}000$). *w/ thinking* denotes the usage of model's dedicated reasoning mode. We highlight the best score per configuration in bold.

task requires the answer to be logically deducible from the prompt alone.

To ensure reliable parsing, we prompt the model to emit its guess in a structured format (see Figure 4), extract the guess via a regular expression, and validate it as a sequence of allowed symbols. We then score the guess against the secret code following the feedback rules of Section 2.1. In the agentic setting, we append each guess-feedback pair to the model's context, making all prior reasoning available for the next guess. The following sections detail the two evaluation paradigms we use to assess LLMs with MASTERMINDEVAL.

3 Experimental Setup

Evaluation Protocol and Metrics. Throughout our experiments, we denote game configurations as (c, n), where c is the code length and n the number of colors. For the agentic and deductive evaluations, each model plays 100 game instances per configuration, and we report the solve rate defined as the fraction of games in which the model identifies the concealed code within the allowed number of guesses. To ensure comparability across models, we draw all game instances from the same random seed. We discard any game in which a model solves the code on the first guess, since the probability of a lucky hit can be non-trivial for easy configurations (e.g., roughly 6.3% at $(c{=}2, n{=}4)$), and continue drawing from the seed until 100 valid games have been played. The deductive evaluation follows the same protocol but caps the model at a single attempt. For the multiple-choice evaluation, we instead assess performance across the full set of generated instances per configuration.

Models and Hyperparameters. We evaluate a diverse set of open-weight models on our benchmark, including Qwen2.5 (Qwen et al., 2025), Qwen3 (Yang et al., 2025), Qwen3.5 (Qwen Team, 2026), Llama-3.1 (Grattafiori et al., 2024), Ministral (Jiang et al., 2023), OLMo-3 (Groeneveld et al., 2024), Gemma-4 (Gemma Team, 2024), and Phi-4 (Abdin et al., 2024). These open-weight models span a parameter range from 4B to 35B. For Qwen3, Qwen3.5, and Gemma-4 we additionally enable the model's extended *thinking* mode, allowing us to isolate the effect of test-time reasoning from the underlying instruction-tuned model. We reuse all suggested hyperparameters as defined either on the Hugging Face Hub (Wolf et al., 2020) or in the respective technical report, and run all open-weight experiments on four Nvidia H100 GPUs with 80 GB of memory using vLLM (Kwon et al., 2023).

Generation Budget. For the generative evaluations (agentic and deductive reasoning), we cap the model's output at 2,048 tokens per response for standard models and 16,384 tokens per response for thinking-enabled models, since the latter need additional headroom for their internal reasoning traces. We set the vLLM context window to 40,960 to-

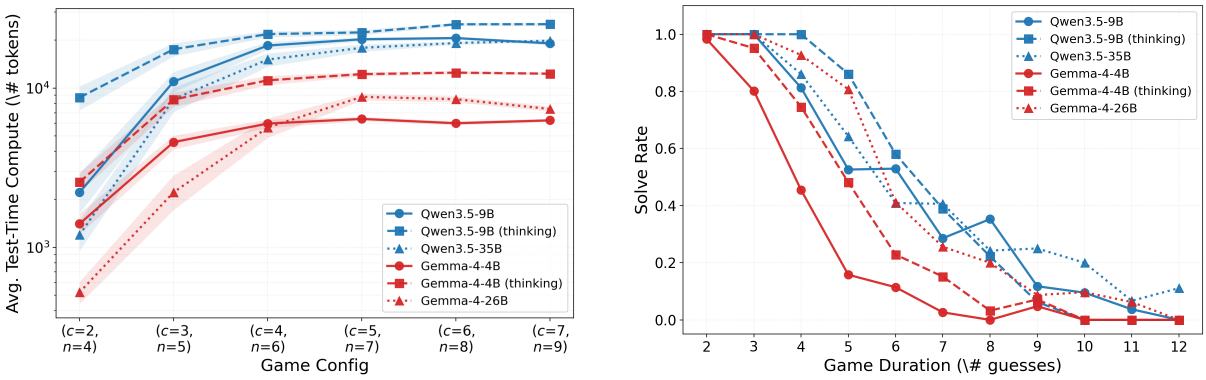

Figure 2: *(left)* Average test-time compute (generated tokens) grows monotonically with game complexity for every model family, confirming that MASTERMINDEVAL scales smoothly with the secret-code length c and the symbol count n. *(right)* Solve rate as a function of how many guesses a game takes: accuracy collapses as games run longer, indicating that once a model makes an incorrect deduction it rarely recovers.

kens for the agentic full game, which must hold the full sequence of guesses, feedback, and responses across all turns, and to 20,480 tokens for single-turn deduction. The multiple-choice evaluation performs no generation: we score each candidate code by its cumulative log-probability under the model, so neither a generation budget nor reasoning modes apply.

4 Results

4.1 Agentic Evaluation

Performance degrades rapidly with task complexity. Table 1 reports solve rates across game configurations. Qwen3.5-9B with thinking and Gemma-4-26B lead overall (avg. 0.74 each), and both solve nearly every game at (c=2, n=4) and (c=3, n=5). Their advantage diminishes quickly with increasing game difficulty, e.g., at (c=4, n=6) Qwen3.5-9B (thinking) solves 0.71 and Gemma-4-26B solves 0.79, but at (c=5, n=7) both drop to 0.27 and 0.21 respectively. This pattern holds universally. Every model collapses toward zero at (c=5, n=7), where the best result reaches only 0.27. The benchmark thus remains sharply discriminative even within a small range of c and n, and easy-looking instances are far from saturated.

Among the smaller models ($\leq$ 14B), Qwen3-8B and Qwen3.5-9B with thinking perform strongest (avg. 0.67 and 0.74), followed by Gemma-4-E4B with thinking (avg. 0.54) and Phi-4-14B (avg. 0.43). The older instruction-tuned models (Llama-3.1-8B, OLMo-3-7B, Qwen2.5-7B) trail substantially (avg. 0.23 to 0.31) and effectively fail beyond (c=3, n=5). Newer model generations therefore deliver clear gains in agentic play that parameter count alone cannot explain, as Qwen3.5-9B (thinking) matches the much larger Gemma-4-26B.

Thinking helps in the agentic setting. Enabling extended reasoning consistently improves multi-turn performance. Qwen3-8B improves from 0.48 to 0.67 on average, Qwen3.5-9B from 0.48 to 0.74, and Gemma-4-E4B from 0.47 to 0.54. The gains peak at intermediate difficulty: at (c=4, n=6), Qwen3.5-9B jumps from 0.21 to 0.71 once thinking is enabled. This shows that even with multi-turn feedback, longer per-turn reasoning traces meaningfully sharpen guess selection.

Test-time compute scales with task complexity. Figure 2 (left) plots the average number of generated tokens for the Qwen3.5 and Gemma-4 families across game configurations of increasing complexity. For every model, test-time compute grows monotonically as c and n increase. The thinking variants and the larger 35B/26B models operate at a markedly higher token budget throughout, but the relative growth is steepest for the non-thinking instruction-tuned models. These models start nearly two orders of magnitude lower at (c=2, n=4) and progressively expand their reasoning to close part of the gap on harder instances, before plateauing once context limits bind.

MASTERMINDEVAL thus offers smooth difficulty scaling with c and n, elicits proportional compute spend from models, and avoids a trivial ceiling at the largest configurations. Procedural instance generation further protects it from data contamination.

Long games rarely recover. Figure 2 (right) plots solve rate against the number of guesses a

Model	Game Config $(c=2, n=4)$	$(c=3, n=5)$	$(c=4, n=6)$	$(c=5, n=7)$	Avg.
Llama-3.1-8B	0.10±0.06	0.01±0.01	0.01±0.01	0.00	0.03
OLMo-3-7B	0.66±0.09	0.16±0.07	0.03±0.04	0.01±0.01	0.21
Qwen2.5-7B	0.20±0.03	0.04±0.01	0.01±0.01	0.01±0.01	0.08
Phi-4-14B	0.50±0.10	0.12±0.07	0.03±0.04	0.02±0.03	0.17
Qwen3-8B	0.64±0.07	0.14±0.05	0.04±0.03	0.02±0.03	0.23
w/ thinking	**1.00**±0.00	**0.80**±0.08	**0.56**±0.10	0.29±0.09	0.66
Qwen3.5-9B	0.77±0.08	0.23±0.09	0.01±0.01	0.00	0.25
w/ thinking	0.98±0.03	0.79±0.08	0.37±0.09	0.01±0.01	0.54
Gemma-4-E4B	0.77±0.08	0.26±0.08	0.01±0.01	0.02±0.03	0.27
w/ thinking	0.97±0.03	0.79±0.08	0.53±0.10	**0.32**±0.09	0.65
Qwen3.5-35B	0.78±0.08	0.30±0.09	0.04±0.04	0.04±0.04	0.29
Gemma-4-26B	**1.00**±0.00	0.79±0.08	0.47±0.10	0.26±0.08	0.63

Table 2: *(Deductive Reasoning Evaluation)* Accuracy of LLMs deducing the secret code in pre-played games that uniquely determine it, given a single attempt ($\pm$ 95 % CI via bootstrapping, $N = 10{,}000$). We highlight the best score per configuration in bold.

game ultimately takes. For all models, accuracy drops sharply with game duration. Instances solved within 2 to 3 guesses are won almost universally, but solve rates fall below 0.5 once games extend past 5 to 6 turns, and approach zero beyond 9 turns. This indicates that once a model makes an incorrect deduction early on, the multi-turn feedback signal rarely suffices for recovery, and errors compound rather than self-correct. The pattern holds even for the strongest thinking models, suggesting that the bottleneck lies not in per-turn reasoning capacity but in the ability to revise hypotheses as contradictory evidence accumulates.

4.2 Deductive Reasoning Evaluation

Table 2 reports the solve rate in the single-step deductive setting, where the provided feedback uniquely determines the code and the model has exactly one attempt. We find that the reasoning mode is the dominant factor: with thinking enabled, Qwen3-8B improves from an average of 0.23 to 0.66, Qwen3.5-9B from 0.25 to 0.54, and Gemma-4-E4B from 0.27 to 0.65. We observe that the thinking variant of Qwen3-8B leads almost everywhere and still solves 0.56 of the standard (c=4, n=6) games, while every non-thinking model collapses to at most 0.04 at the same configuration. Without thinking, we find that even the strongest instruction-tuned models (Qwen3.5-9B and Gemma-4-E4B at 0.77, Qwen3.5-35B at 0.78 on the easiest configuration) effectively fail once $c \geq 4$, and that the older Llama-3.1-8B and Qwen2.5-7B never exceed 0.20 even at (c=2, n=4). Among non-thinking models, we observe Gemma-4-26B as the clear outlier, reaching 0.63 on average and matching the thinking variants of smaller models.

These results mirror the agentic setting. We observe that thinking adds little when the model can iterate against live feedback, but find it decisive when the model must combine all constraints in a single pass. The contrast also reveals a masking effect: we find that Qwen3.5-9B and Gemma-4-E4B solve 0.59 to 0.77 of (c=3, n=5) games agentically (Table 1), yet only 0.23 to 0.26 when the same information arrives at once. We conclude that strong multi-turn play does not imply strong deduction, and that models instead rely on iterative, heuristic refinement rather than integrating the feedback into a single consistent inference.

4.3 Multiple-Choice Evaluation

In Table 3, we report the results for the multiple-choice setting. We keep the evaluation here limited to the best performing models from previous settings.

Models cluster tightly without any reasoning. We find that Qwen3.5-35B achieves best performance overall (avg. 0.61), with the remaining models clustering between 0.54 and 0.57. The spread across model families is much smaller than in the generative settings, and we observe that even the smallest model (Gemma-4-E4B at 0.54) lies within

Model	Game Config $(c=2, n=4)$	$(c=3, n=5)$	$(c=4, n=6)$	$(c=5, n=7)$	$(c=6, n=8)$	Avg.
Qwen3-8B	$0.29_{\pm 0.02}$	$0.46_{\pm 0.02}$	$0.57_{\pm 0.02}$	$0.66_{\pm 0.12}$	$0.84_{\pm 0.09}$	0.57
Qwen3.5-9B	$0.34_{\pm 0.02}$	$0.49_{\pm 0.02}$	$0.54_{\pm 0.02}$	$0.63_{\pm 0.12}$	$0.83_{\pm 0.09}$	0.57
Qwen3.5-35B	$0.38_{\pm 0.02}$	$0.49_{\pm 0.02}$	$0.56_{\pm 0.02}$	$\mathbf{0.73}_{\pm 0.11}$	$\mathbf{0.88}_{\pm 0.08}$	0.61
Gemma-4-E4B	$0.37_{\pm 0.02}$	$0.49_{\pm 0.02}$	$0.54_{\pm 0.02}$	$0.61_{\pm 0.12}$	$0.67_{\pm 0.12}$	0.54
Gemma-4-26B	$0.39_{\pm 0.02}$	$0.49_{\pm 0.02}$	$0.58_{\pm 0.02}$	$0.63_{\pm 0.12}$	$0.77_{\pm 0.10}$	0.57

Table 3: *(Multiple-Choice Evaluation)* Log-likelihood ranking accuracy across game configurations (accuracy $\pm$ 95 % CI, bootstrapped $N = 100k$).

a few points of the larger ones. We interpret this as evidence that the elimination signal exploited here is broadly present in pretrained parameters and not specific to the most capable models.

Accuracy rises with game complexity. Counter-intuitively, we find that multiple-choice accuracy *increases* with c and n. Qwen3.5-35B improves from 0.38 at $(c{=}2, n{=}4)$ to 0.88 at $(c{=}6, n{=}8)$, and we observe similar monotone trends for every model. Longer codes give the model more positions over which the log-probability of the correct sequence can separate from the distractors, and the highly constrained prompt lets that signal accumulate.

Figure 3 resolves this effect position by position. The top-1 rank of the correct option (left) stays at or below chance for all but the final code position, where it climbs sharply, and the climb is steeper for longer codes. The log-probability margin between the correct option and the best wrong option (right) tells the same story: margins remain negative throughout the prefix and only turn positive at the final position. We conclude that the gain concentrates entirely at the end of the code rather than reflecting a length-driven prior, which is consistent with the model integrating the deductive content of the prompt rather than exploiting a surface artifact. Crucially, ranking pre-given options is a much weaker probe than the generative settings: strong log-likelihood discrimination does not translate into the ability to actively deduce the code or play the game.

5 Related Work

Reasoning has long been considered a fundamental capability of language models (Liu et al., 2020; Cobbe et al., 2021; Suzgun et al., 2023; Srivastava et al., 2023). As model size increases, reasoning abilities naturally emerge, making it a core attribute of LLMs, as demonstrated in prior work (OpenAI et al., 2024). To better utilize this capability, techniques such as chain-of-thought prompting (Wei et al., 2023) and specialized training through reinforcement learning (Lambert et al., 2025) have been widely adopted. Most recently, many works have demonstrated the positive effect of increased test-time compute (generating more tokens at test time) to be better at reasoning tasks (Snell et al., 2024; Muennighoff et al., 2025). Another direction is multi-step reasoning that breaks down complex decision-making and planning tasks into smaller solvable tasks, which are frequently encountered in LLM-based agents (Guo et al., 2024; Xi et al., 2023).

Many benchmarks exist to assess various aspects of reasoning such as inductive reasoning (Sinha et al., 2019), temporal reasoning (Fatemi et al., 2024; Wang and Zhao, 2024), spatial reasoning (Mirzaee et al., 2021), counterfactual reasoning (Tandon et al., 2019), mathematical reasoning (Glazer et al., 2024) or logical reasoning (Han et al., 2024). (Gui et al., 2024). Additionally, knowledge-based reasoning, particularly commonsense reasoning (Talmor et al., 2019), assesses whether models can apply commonsense knowledge to make sound inferences. More advanced reasoning tasks, such as theory-of-mind reasoning, evaluate whether models can infer human thoughts and beliefs (Wagner et al., 2024).

Reasoning can also be evaluated through board games such as Chess (Campbell et al., 2002) or Go (Silver et al., 2016), which require highly complex strategic planning. While specialized models are typically trained for such games, reasoning capabilities in language models are assessed using benchmarks such as LogicGame (Gui et al., 2024), LogicBench (Parmar et al., 2024), BoardgameQA (Kazemi et al., 2023), TruthQuest (Mondorf and Plank, 2024), and GAMEBoT (Lin et al., 2024). These benchmarks introduce simplified yet chal-

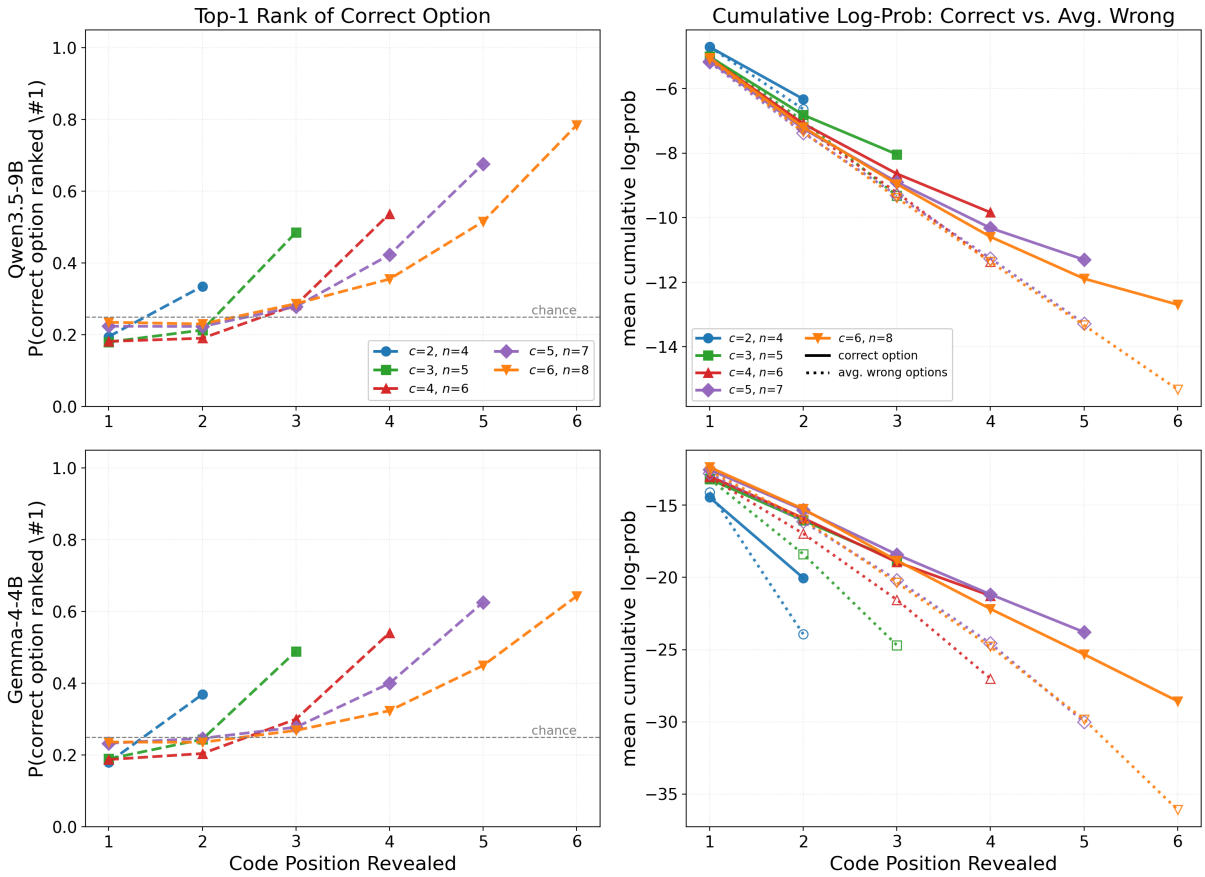

Figure 3: Position-resolved log-probability analysis for Qwen3.5-9B (top) and Gemma-4-4B (bottom). *Left:* probability that the correct option is ranked first after revealing each code position; curves stay near chance (0.25) and only rise close to the final position. *Right:* mean cumulative log-probability of the correct option (solid) versus the average wrong option (dotted); the two separate only at the final position. The discriminative signal emerges at latter symbols rather than accumulating uniformly across the code.

lenging board game environments. However, they all share the following limitations: *(i)* They are not easily scalable in terms of complexity. For example, increasing the board size in Tic-Tac-Toe does not significantly increase its complexity, whereas Mastermind becomes considerably more difficult when using longer secret codes. *(i)* Evaluating models solely based on game outputs does not determine whether a model is genuinely reasoning and pursuing a strategy to win or merely reproducing good moves memorized from large-scale pretraining data.

6 Conclusion

We introduced MASTERMINDEVAL, a Mastermind-based benchmark that evaluates language models through three paradigms: agentic evaluation, single-step deduction evaluation, and log-likelihood ranking. In the generative settings, performance drops sharply as game complexity grows, and even classical human-level configurations remain far from saturated. Our analysis surfaces two robust limitations of current models: *(i)* they rarely combine prior feedback into a single consistent deduction, relying instead on iterative, heuristic refinement, and *(ii)* once they adopt an inconsistent hypothesis, they rarely recover. We further find that extended "thinking" substantially improves both agentic play and single-step deduction, with the largest gains in the latter. In contrast, log-likelihood ranking accuracy holds up and even rises with complexity, but this signal concentrates at the final code positions and does translate into the ability to actively eliminate guesses that are not valid. MASTERMINDEVAL thus offers a simple and scalable framework for probing language model reasoning, both under test-time compute and directly from pretrained parameters, while remaining robust to data contamination by construction.

Limitations

While our benchmark provides valuable insights into the reasoning capabilities of language models through the game of Mastermind, our evaluation has several limitations. Due to computational constraints, we were only able to systematically assess smaller open-source models, restricting our analysis to models that could be run efficiently on available hardware. Additionally, we did not train any models; however, methods such as reinforcement learning via feedback (RLVF) (Lambert et al., 2025) could potentially improve both strategic play and deductive reasoning in language models.

Another limitation lies in the scope of qualitative information measurement. While we can track the remaining state space in each game configuration using Knuth's algorithm, contributions from other research areas, such as game theory and psychology, could provide a more fine-grained understanding of how models process structured information. In particular, a more precise quantification of the informational content in each hint could further refine our analysis of LLM reasoning capabilities.

Furthermore, we cannot determine whether models are truly reasoning in MASTERMINDEVAL or merely relying on pattern-matching learned during pretraining. The paper does not establish whether games of Mastermind are present in the pretraining data, leaving open the possibility that performance is influenced by prior exposure rather than reasoning ability. Verifying this through controlled experiments or data ablation would help clarify the extent to which MastermindEval measures genuine reasoning. Additionally, the study does not analyze how results in MastermindEval correlate with established reasoning benchmarks, making it difficult to assess its validity as a reasoning metric.

Another limitation is the absence of human evaluation. Since performance in MastermindEval appears to correlate with model size, it would be worth investigating whether it also aligns with measures of human intelligence. One straightforward approach would be to test whether people with higher educational attainment perform better in the task. Exploring this connection could provide additional support for MastermindEval as a reasoning benchmark. Future work incorporating human evaluations would help determine whether the task meaningfully reflects reasoning ability beyond model scaling trends.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

François Bonnet and Simon Viennot. 2016. Nash equilibrium in mastermind. In *International Conference on Computers and Games*, pages 115–128. Springer.

Sebastian Bordt, Suraj Srinivas, Valentyn Boreiko, and Ulrike von Luxburg. 2025. How much can we forget about data contamination? *Preprint*, arXiv:2410.03249.

Murray Campbell, A.Joseph Hoane, and Feng hsiung Hsu. 2002. Deep blue. *Artificial Intelligence*, 134(1):57–83.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. 2024. GTBench: Uncovering the strategic reasoning capabilities of LLMs via game-theoretic evaluations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang (Lorraine) Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*, volume 36, pages 70293–70332. Curran Associates, Inc.

Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. 2024. Test of time: A benchmark for evaluating llms on temporal reasoning. *Preprint*, arXiv:2406.09170.

Gemma Team. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, and 5 others. 2024. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *Preprint*, arXiv:2411.04872.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, and 1 others. 2024. OLMo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*.

Jiayi Gui, Yiming Liu, Jiale Cheng, Xiaotao Gu, Xiao Liu, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2024. Logicgame: Benchmarking rule-based reasoning abilities of large language models. *Preprint*, arXiv:2408.15778.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *Preprint*, arXiv:2402.01680.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.

Md. Najib Hasan, Md Mahadi Hassan Sibat, Mohammad Fakhruddin Babar, Souvika Sarkar, Monowar Hasan, and Santu Karmaker. 2026. Pitfalls of evaluating language models with open benchmarks. *Preprint*, arXiv:2507.00460.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Mehran Kazemi, Quan Yuan, Deepti Bhatia, Najoung Kim, Xin Xu, Vaiva Imbrasaite, and Deepak Ramachandran. 2023. Boardgameqa: A dataset for natural language reasoning with contradictory information. In *Advances in Neural Information Processing Systems*, volume 36, pages 39052–39074. Curran Associates, Inc.

Donald Ervin Knuth. 1977. The computer as master mind. *Journal of Recreational Mathematics*, 9:1–6.

Barteld Kooi. 2005. Yet another mastermind strategy. *ICGA Journal*, 28(1):13–20.

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

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2025. Tulu 3: Pushing frontiers in open language model post-training. *Preprint*, arXiv:2411.15124.

Wenye Lin, Jonathan Roberts, Yunhan Yang, Samuel Albanie, Zongqing Lu, and Kai Han. 2024. Beyond outcomes: Transparent assessment of llm reasoning in games. *Preprint*, arXiv:2412.13602.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, and 101 others. 2026. Ministral 3. *Preprint*, arXiv:2601.08584.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3622–3628. International Joint Conferences on Artificial Intelligence Organization. Main track.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. 2021. SPARTQA: A textual question answering benchmark for spatial reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4582–4598, Online. Association for Computational Linguistics.

Philipp Mondorf and Barbara Plank. 2024. Liar, liar, logical mire: A benchmark for suppositional reasoning in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7114–7137, Miami, Florida, USA. Association for Computational Linguistics.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, and 50 others. 2026. Olmo 3. *Preprint*, arXiv:2512.13961.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13679–13707, Bangkok, Thailand. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5. Alibaba Cloud.

John Schultz, Jakub Adamek, Matej Jusup, Marc Lanctot, Michael Kaisers, Sarah Perrin, Daniel Hennes, Jeremy Shar, Cannada A. Lewis, Anian Ruoss, Tom Zahavy, Petar Veličković, Laurel Prince, Satinder Singh, Eric Malmi, and Nenad Tomasev. 2025. Mastering board games by external and internal planning with language models. In *Forty-second International Conference on Machine Learning*.

David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4515, Hong Kong, China. Association for Computational Linguistics.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 431 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*. Featured Certification.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *Preprint*, arXiv:1811.00937.

Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. WIQA: A dataset for "what if..." reasoning over procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6076–6085, Hong Kong, China. Association for Computational Linguistics.

Eitan Wagner, Nitay Alon, Joseph M. Barnby, and Omri Abend. 2024. Mind your theory: Theory of mind goes deeper than reasoning. *Preprint*, arXiv:2412.13631.

Yuqing Wang and Yun Zhao. 2024. TRAM: Benchmarking temporal reasoning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6389–6415, Bangkok, Thailand. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Xiaobao Wu, Liangming Pan, Yuxi Xie, Ruiwen Zhou, Shuai Zhao, Yubo Ma, Mingzhe Du, Rui Mao, Anh Tuan Luu, and William Yang Wang. 2025. AntiLeakBench: Preventing data contamination by automatically constructing benchmarks with updated real-world knowledge. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18403–18419, Vienna, Austria. Association for Computational Linguistics.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, and 10 others. 2023. The rise and potential of large language model based agents: A survey. *Preprint*, arXiv:2309.07864.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Jianzhu Yao, Kevin Wang, Ryan Hsieh, Haisu Zhou, Tianqing Zou, Zerui Cheng, Zhangyang Wang, and Pramod Viswanath. 2025. SPIN-bench: How well do LLMs plan strategically and reason socially? In *Second Conference on Language Modeling*.

Yinqi Zhang, Xintian Han, Haolong Li, Kedi Chen, and Shaohui Lin. 2025. Complete chess games enable LLM become a chess master. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 1–7, Albuquerque, New Mexico. Association for Computational Linguistics.

Xin Zhou, Martin Weyssow, Ratnadira Widyasari, Ting Zhang, Junda He, Yunbo Lyu, Jianming Chang, Beiqi Zhang, Dan Huang, and David Lo. 2025. Lessleak-bench: A first investigation of data leakage in llms across 83 software engineering benchmarks. *Preprint*, arXiv:2502.06215.

A Appendix

A.1 Dataset Statistics

We show the dataset statistics for the pre-played games in Table 4.

(c, n)	Multipe-Choice	Deduc.
$(2, 4)$	30,4k	30,4k
$(3, 5)$	37,1k	37,1k
$(4, 6)$	36,1k	36,1k
$(5, 7)$	1,2k	3,1k
$(6, 8)$	1,3k	3,2k
Total	106,1k	109,8k

Table 4: Number of examples per configuration (c, n) for multiple-choice and the deductive reasoning variant (Deduc.).

Evaluation Prompt: Mastermind Guesser

You are playing Mastermind as the guesser. A secret color code has been chosen and your goal is to figure it out through a series of guesses.

Game rules:
- The secret code consists of `{code_length}` colors. Duplicates are allowed.
- The only allowed colors are: `{possible_colors}`.
- You have at most `{max_guesses}` guesses to find the code.
- After each guess you will receive feedback:
  1. How many colors are correct AND in the correct position.
  2. How many colors are correct but in the WRONG position.
- A color is only counted once in the feedback, with exact matches taking priority.

How to respond:
- You must always make a guess. Never ask the user to guess for you.
- End every response with your guess on its own line in this exact format:
  `FINAL GUESS: ['color1', 'color2', ...]`
- Only use colors from the allowed list above.
- Use the feedback from previous guesses to narrow down the secret code.

Figure 4: Prompt used to instruct the language model as the codebreaker in the agentic Mastermind evaluation. Placeholders in braces are filled with the configuration values for each game instance.

A.2 Game Prompt

In Figure 4, we show the initial prompt used to play Mastermind. We then continue with a user–assistant interaction until the maximum number of guesses is reached or the game is solved.