

TTT-Bench: A Benchmark for Evaluating Reasoning Ability with Simple and Novel Tic-Tac-Toe-style Games

Prakamya Mishra *, Jiang Liu *, Jialian Wu, Xiaodong Yu,
Zicheng Liu, Emad Barsoum

Advanced Micro Devices, Inc. (AMD)
<https://prakamya-mishra.github.io/TTTBench>

Abstract

Large reasoning models (LRMs) have demonstrated impressive reasoning capabilities across a broad range of tasks including Olympiad-level mathematical problems, indicating evidence of their complex reasoning abilities. While many reasoning benchmarks focus on the STEM domain, the ability of LRMs to reason correctly in broader task domains remains underexplored. In this work, we introduce **TTT-Bench**, a new benchmark that is designed to evaluate basic strategic, spatial, and logical reasoning abilities in LRMs through a suite of four two-player Tic-Tac-Toe-style games that humans can effortlessly solve from a young age. We propose a simple yet scalable programmatic approach for generating verifiable two-player game problems for TTT-Bench. Although these games are trivial for humans, they require reasoning about the intentions of the opponent, as well as the game board’s spatial configurations, to ensure a win. We evaluate a diverse set of state-of-the-art LRMs, and **discover that the models that excel at hard math problems frequently fail at these simple reasoning games**. Further testing reveals that our evaluated reasoning models score on average $\downarrow 41\%$ & $\downarrow 5\%$ lower on TTT-Bench compared to MATH 500 & AIME 2024 respectively, with larger models achieving higher performance using shorter reasoning traces, where most of the models struggle on long-term strategic reasoning situations on simple and new TTT-Bench tasks.

1 Introduction

Recent advances in large reasoning models (LRMs) have driven significant breakthroughs across various reasoning tasks (Jaech et al., 2024; DeepSeek-AI, 2025; Team, 2025c; Zhao et al., 2024; Guan et al., 2025), including deductive, arithmetic, commonsense, relational, and symbolic reasoning

(Huang and Chang, 2023). The high quality of reasoning data and improvements in reinforcement learning techniques (Luo et al., 2025a; Zhang et al., 2024a; Rafailov et al., 2023; Xu et al., 2024; Grattafiori et al., 2024) have markedly improved the performance of LRMs on previously challenging mathematics benchmarks, such as GSM8K (Cobbe et al., 2021), MATH 500 (Lightman et al., 2024), and AIME 2024 (MAA, 2024). However, despite these advancements, the ability of LRMs to reason well on a broader set of reasoning problems, in domains other than maths & STEM, remains largely unexplored.

Many of the existing reasoning benchmarks focus on mathematical or STEM-related problems (Lightman et al., 2024; He et al., 2024; Rein et al., 2024; Suzgun et al., 2023; Zhang et al., 2024b; Mao et al., 2024), mainly due to their verifiable nature and the abundance of annotated problems (Li et al., 2025; Xu et al., 2025). This narrow focus limits the evaluation of LRMs to specific domains and leaves their broader reasoning capabilities largely untested (Huang and Chang, 2023). There is a clear need for benchmarks that evaluate a diverse set of reasoning capabilities (Chang et al., 2023), including strategic, spatial, and logical reasoning. Exploring this broader range of reasoning capabilities is particularly compelling when we consider human reasoning abilities. Even young children effortlessly perform intuitive reasoning tasks, particularly in strategic two-player games. They naturally anticipate their opponent’s moves, block threats, and plan multi-step strategies, relying on an instinctive grasp of strategy and spatial relationships rather than explicit or formal reasoning.

An open question that emerges is: *Whether reasoning models proficient in solving sophisticated mathematical problems can correctly reason on broader domains of problems that require basic strategic, spatial, and logical reasoning with straightforward rational and minimal requirements*

*Corresponding Authors: Prakamya Mishra (Prakamya.Mishra@amd.com) and Jiang Liu (Jiang.Liu@amd.com).

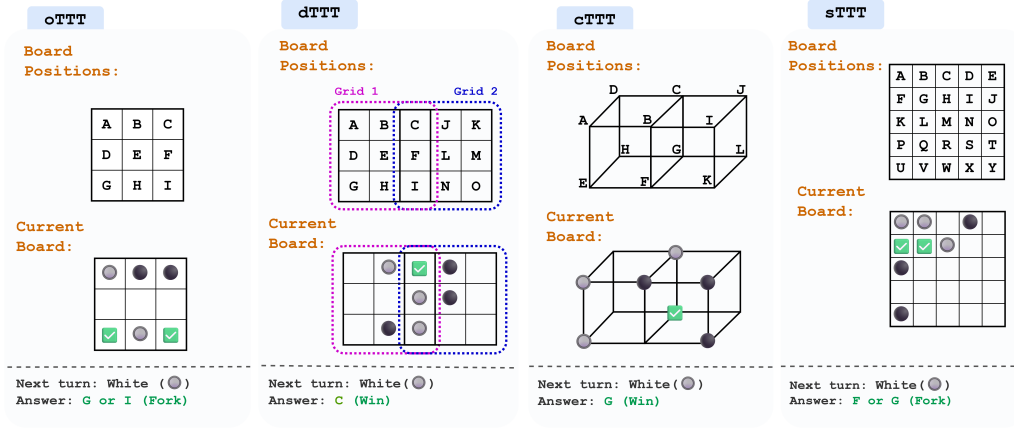


Figure 1: Visualization of four game types in TTT-Bench (oTTT, dTTT, cTTT, and sTTT). For each game, the illustration consists of the board position annotations (above), the current game state with Alice’s (white) and Bob’s (black), and the correct next move (represented by a check mark).

for specialized knowledge. Although there are a few efforts to use two-player board games to evaluate these capabilities in language models (Fontana et al., 2024; Qin et al., 2024; Costarelli et al., 2024; tse Huang et al., 2025), these benchmark games are well-documented in the literature and online, hurting their reliability due to the elevated risk of training of these LRMs on associated benchmark test data.

Motivated by this gap, we introduce **TTT-Bench**, a new benchmark specifically created to evaluate the reasoning capability of LRMs through a suite of **simple and novel two-player Tic-Tac-Toe-style games** (Figure 1). Although trivial for humans, these games require the following three basic reasoning capabilities: (1) **Spatial reasoning**: to interpret and act upon structured board, (2) **Strategic planning**: to anticipate an opponent’s next move and for selecting actions that maximize long-term success, and (3) **Logical reasoning**: essential in two-player games that require reasoning over future game trajectories. These capabilities are often cited as open challenges for LRMs in prior work, including recent benchmarks on planning and spatial tasks (Duan et al., 2024; Lin et al., 2025).

TTT-Bench is a simple and diverse benchmark consisting of four types of two-player games, namely oTTT, dTTT, cTTT, and sTTT (see Figure 1 for examples), and the questions in the benchmark test the reasoning capability of LRMs by posing the question of predicting the best move by a player. The objectives of these games are not only simple but are also introduced here for the first time (except oTTT), with no documented prior literature available online, making our benchmark uncontam-

inated and highly reliable. Briefly, for each task in TTT-Bench, two players play a strategic game on a configuration dependent on the task to reach a winning state that satisfies a task-dependent winning constraint, and the player who reaches the winning state first wins the game. The benchmark consists of programmatically generated board game questions along with their verifiable answers, where we employ a simple heuristic-based algorithm (Appendix 1) to identify the finite set of next best possible moves and their verdict as the answers to ensure the verifiability of the solutions.

The main contributions of this work are as follows:

- In this work, we introduce **TTT-Bench** benchmark to evaluate the reasoning capabilities of LRMs that are proficient in solving difficult math problems, on a broader domain of basic strategic, spatial, and logical reasoning tasks through a suite of four simple two-player Tic-Tac-Toe-style games.
- Evaluated a variety of state-of-the-art (SOTA) LRMs on TTT-Bench, and revealed a surprising finding: **models skilled at difficult math problems frequently struggle with these simpler reasoning tasks**.
- This work not only highlights a fundamental shortcoming in existing LRMs, but also provides a new, simple, and scalable approach for automated verifiable two-player game generation, fostering future research efforts in evaluating the reasoning capability of LRMs.

2 Related Work

Advances in Mathematical Reasoning with Large Language Models Recent developments in LRMs have led to significant improvements in their reasoning capabilities (OpenAI, 2025; Jaech et al., 2024; DeepSeek-AI, 2025; Team, 2025c; LG AI Research, 2025; Team et al., 2025) - exhibited by an exponential performance boost in complex reasoning tasks, especially on mathematical problem-solving benchmarks (Lightman et al., 2024; MAA, 2024). Contrary to massive reasoning models, efficiently trained small reasoning models (Luo et al., 2025b; Wen et al., 2025; Team, 2025a; Ye et al., 2025; Muennighoff et al., 2025) have also proven to achieve competitive results across these benchmarks. Rapid improvements in reasoning capabilities of LRMs can be attributed to various training methodologies (Yu et al., 2025; Shao et al., 2024), where Reinforcement learning with verifiable rewards (RLVR) (Lambert et al., 2024; DeepSeek-AI, 2025) has emerged as a predominant approach.

Evaluating Reasoning Ability of LRMs Despite these advancements, the present widely used reasoning benchmarks are predominantly based on mathematical and STEM-focused tasks, potentially limiting the assessment of these models' reasoning ability in much broader domains. Several works have demonstrated significant shortcomings of LRMs on intuitive reasoning tasks (Valmeekam et al., 2024a,b; Topsakal and Harper, 2024; Topsakal et al., 2024), where even SOTA reasoning models fall short in planning and reasoning. Such findings underscore the necessity for the development of benchmarks that assess a broader spectrum of reasoning, encompassing intuitive and strategic thinking.

3 TTT-Bench

TTT-Bench is a simple and scalable two-player Tic-Tac-Toe (TTT) style game benchmark for evaluating the reasoning ability of the present SOTA LRMs. This benchmark consists of four games, namely: **oTTT** (ordinary TTT), **dTTT** (double TTT), **cTTT** (TTT in cube), and **sTTT** (TTT with squares), which are described in detail below:

- **oTTT**: This is the ordinary TTT game, where two players play a game on a 3x3 grid. The points on the grid are labeled top to bottom and left to right as A, B, C, D, E, F, G, H, and

I. One player plays with white stone (○) and the other uses black stone (●). At each turn, the player places a stone of the corresponding color onto one of the positions that has not been occupied. *Whoever has three stones in a line (horizontal, vertical, or diagonal) wins.*

- **dTTT**: This is a variant of TTT with two (double) adjacent 3x3 grids. The points on the first grid are labeled top to bottom, left to right, as A, B, C, D, E, F, G, H, and I. The points on the second grid are labeled top to bottom, left to right, as C, J, K, F, L, M, I, N, and O. Here, note that the points C, F, and I are shared by the two grids. One player plays with white stone (○) and the other uses black stone (●). At each turn, the player places a stone of the corresponding color onto one of the positions that has not been occupied. *Whoever has three stones in a line (horizontal, vertical, or diagonal) on either grid wins.*
- **cTTT**: This is a variant of TTT which is played on two adjacent cubes. The points A, B, C, and D form the top rectangle in the first cube, and B, I, J, and C form the top rectangle in the second cube. Points E, F, G, and H form the bottom rectangle in the first cube, and F, K, L, and G form the bottom rectangle in the second cube. A-E, B-F, C-G, D-H, I-K, and J-L are the edges. Here, note that the points B, C, G, and F are shared by the two cubes. One player plays with a white sticker (○) and the other uses a black sticker (●). At each turn, the player places a stone of the corresponding color onto one of the positions that has not been occupied. *Whoever has four stickers on the same plane on either cube wins.*
- **sTTT**: This is a variant of TTT which is played on a large grid of five intersections between horizontal and vertical lines. There are five equally spaced horizontal lines, where the distance between two neighboring horizontal lines is one. Similarly, there are five equally spaced vertical lines, and the distance between two neighboring vertical lines is one. There are twenty-five intersection points between the five horizontal and vertical lines. These twenty-five points are labeled from top to bottom, left to right, as A, B, C, D, ..., Y. One player plays with white stone (○) and the other uses black stone (●). At each turn,

the player places a stone of the corresponding color onto one of the twenty-five points that have not been occupied. *Whoever has four stones that form either a unit square (with side length of one) or a “diagonal square” with side length equal to the square root of two wins.*

Examples of these games are illustrated in Figure 1, with the corresponding exact questions listed in the Appendix A. As seen from the illustrations, such games are easy for humans to play even from a young age, where a player’s victory is based on their ability to anticipate their opponent’s move, blocking threats, and if they can form basic winning strategies – skills demonstrating the simple reasoning ability of humans.

All the problems in this benchmark are formulated as a text-based board game question that starts with the description of the game, as well as the current game state with all the moves played by both players. The final TTT-Bench task questions are formulated to predict the next best move position by the current player (Appendix A), like: **Where should Bob play next?**. Such question formulation ensures that our benchmark solving complexity is low, as just predicting the next move out of a small set of possible available moves in our games guarantees a small solution search space, making our benchmark questions relatively easier compared to other works (Shojaee*† et al., 2025). Since the outcomes of these games are based on finite and deterministic logic, possible next-best moves can be easily identified. Leveraging this, we created an automated pipeline to generate new test samples for each of the four games with verifiable solutions, enabling this benchmark to not only have new and unique questions that the SOTA LRMs are not contaminated against, but also make our benchmark highly scalable in terms of the size, diversity and difficulty.

3.1 Automated TTT-Bench Generation

We employ a programmatic approach to generate a diverse set of questions corresponding to the four games present in the TTT-Bench. In this approach, for each game, we first generate all possible game states between two players having played N moves. We then filter out the states for further processing where 1) neither player has already won, and 2) where the player who made the N^{th} move has not yet established a "Winning Fork". Here, by "Win-

ning Fork", we refer to the game state in which a player has two or more potential immediate winning states, with each potential winning state being one turn (move) short of winning the game. In this state, independent of the next move of the opponent, the current player can win with 100% probability by pursuing either of the winning moves on their next chance. Refer to Appendix A for more illustrative examples of "Winning Fork" states. This leaves us with all the game states in which there is a deterministic set of positions in the board for the player playing the $(N + 1)^{th}$ move, that can lead to their victory in the game.

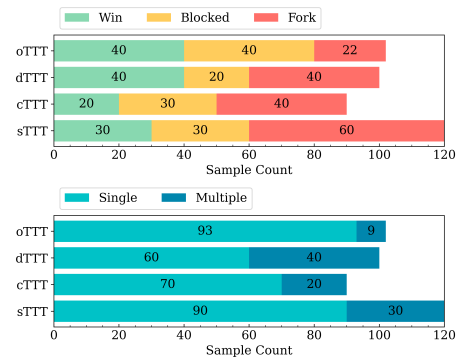


Figure 2: Distribution of TTT-Bench question solution types. **Above:** Distribution of questions and their game verdicts (Win, Blocked, or Fork) based on how the game concludes after the corresponding solution move is played. **Below:** Distribution of questions and their solution types (single answer or multiple next-best moves).

Verdict Category	Description
Win	The player wins the game by playing the $(N + 1)^{th}$ move.
Blocked	The player blocks the opponent’s immediate potential win by playing the $(N + 1)^{th}$ move.
Fork	The player establishes two or more potential winning states (with exactly one turn short of winning in all the states) by playing the $(N + 1)^{th}$ move.

Table 1: Description of game verdict categories in TTT-Bench games.

Our final test set consists of only those games in which N moves had been played, and the verdict of the game after the $(N + 1)^{th}$ move falls into one of the categories listed in Table 1. These categories ensure that the solutions generated for TTT-Bench questions, for all the games, are the optimal next-best moves. *The Algorithm 1 in Appendix C describes how we generated optimal next-best move solutions for the TTT-Bench questions*, where the game verdicts are either "Win", "Blocked", or "Fork". In the algorithm, we first check if there

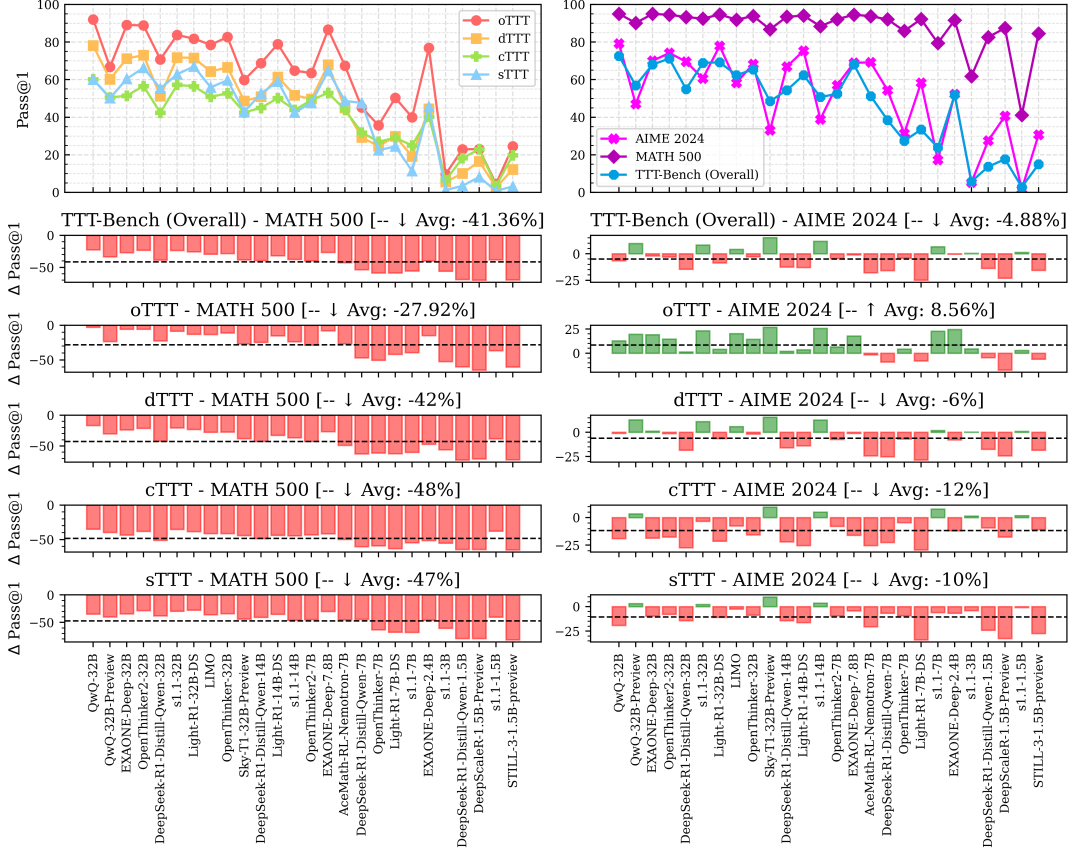


Figure 3: Pass@1 performance comparison of LLMs on TTT-Bench (Left) and math benchmarks: AIME 2024 & MATH 500 (Right). Exact performance numbers available in Table 3 of Appendix D

exists any winning moves at the $(N + 1)^{th}$ chance where N moves have already been played so far, and use those moves as the solution with the verdict of "Win". If there are no winning moves, then we checked if there are moves by which the opponent can win on the $(N + 1)^{th}$ chance, and used those as the solution with the verdict of "Blocked", since not playing those moves will ensure the opponents win. Finally, in case of no blocking moves as well, we identified moves that result in a fork for the current player that will ensure a future win, and used those moves as the solution with the verdict of "Fork".

Based on this automated game question-answer generation approach explained above, we generate a diverse final test set consisting of ~ 100 samples from each game. For each game, we used a set of predefined values of N such that the games were simple, solvable, and had a low branching factor. For oTTT we use $N = \{4, 5\}$, $N = \{6, 7\}$ for dTTT, $N = \{5, 6, 7\}$ for cTTT, and $N = \{6, 7\}$ for sTTT. As shown in Figure 2, not only is this dataset diverse across different game verdicts, but

it also consists of questions with both single and multiple next-best move solutions. Overall, the final TTT-Bench benchmark consists of 412 questions from four simple two-player TTT games. We release this benchmark for future research under the Apache 2.0 License.

4 Main Results

4.1 Evaluation Setup

In this work, we evaluate a comprehensive set of recent SOTA LLMs on TTT-Bench and conduct a head-to-head comparison of their performance on TTT-Bench versus two widely used mathematics benchmarks: AIME 2024 (MAA, 2024) (Olympiad-level math) & MATH500 (Lightman et al., 2024) (high school math), to thoroughly investigate the reasoning capabilities of these models. The parameter count of the models in our evaluations ranges from 1.5 billion to 32 billion parameters, which includes models like DeepSeek-R1-Distill-Qwen-1.5B / 7B / 14B / 32B (DeepSeek-AI, 2025), S1.1-1.5B / 3B / 7B / 14B / 32B (Muenighoff et al., 2025), EXAONE-Deep-2.4B / 7.8B

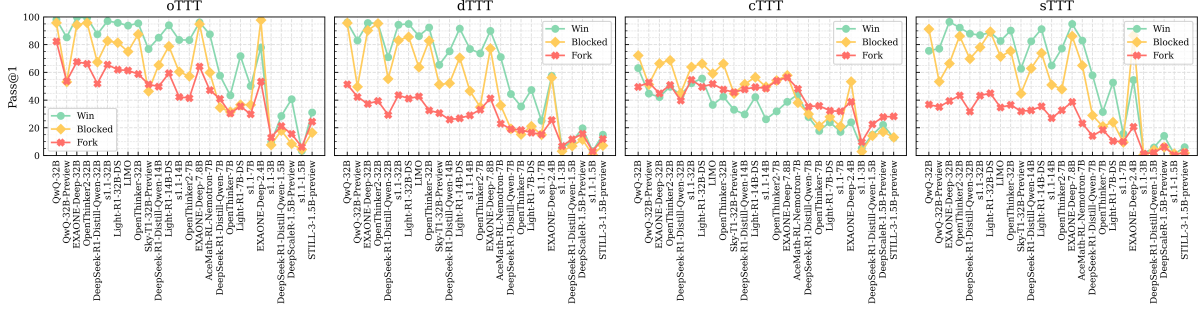


Figure 4: Pass@1 performance analysis of LRMs over individual solution verdict category ("Win", "Blocked", & "Fork") questions in TTT-Bench tasks. Exact performance numbers available in Table 4 of Appendix D.

/ 32B (LG AI Research, 2025), QwQ-32B / 32B-Preview (Team, 2025c), OpenThinker2-7B / 32B (Team, 2025b), and Light-R1-7B/14B/32B (Wen et al., 2025). To explore the reasoning capabilities of the frontier reasoning models, we also evaluated OpenAI’s O3-mini-medium as well as DeepSeek’s DeepSeek-R1 models on TTT-Bench. Our evaluation framework is based on the DeepScaler (Luo et al., 2025b) and VerL (Sheng et al., 2024) codebases. All of our evaluations are done with AMD Instinct MI300XTM GPUs. In the results for all the benchmarks, we report Pass@1 scores that are calculated based on $k = 16$ responses, where we generate k responses for each test sample with a temperature of 0.6, a top-p value of 0.95, and a response length of 28k tokens, and calculate its corresponding Pass@1 as:

$$Pass@1 = \frac{1}{k} \sum_{i=1}^k p_i \quad (1)$$

Here, p_i represents the correctness of the i^{th} response ($p_i = 1$ in the case of questions with multiple solutions if the model predicts either of the solutions correctly). The final solution predictions are extracted from the generated responses using regular expression extraction from the last boxed answer (Luo et al., 2025b). Only for frontier models, we report the Pass@1 scores based on just one response per test sample in the case of TTT-Bench, whereas we report their corresponding published results in the case of math benchmarks.

4.2 TTT-Bench Evaluation Results

Weak reasoning ability of LRMs on simple and intuitive tasks: In Figure 3, we report the performance of SOTA reasoning models individually on the four games in TTT-Bench as well as on AIME 2024 & MATH 500. In our analysis, we

	oTTT	dTTT	cTTT	sTTT	MATH 500	AIME 2024
o3-mini-medium	79.6	83.0	52.2	75.0	97.3*	79.6*
DeepSeek-R1	88.9	72.0	58.9	75.8	97.3*	79.8*

Table 2: Frontier LRM Pass@1 performance on TTT-Bench and math benchmarks. The (*) indicates that the results for math benchmarks we sourced from the respective model’s published results. The TTT-Bench task performances are based on a single response per test sample.

leverage the performance drop ($\Delta Pass@1$ as defined below) between the TTT-Bench tasks (x) and math benchmarks (y) as a metric to measure the divergence in reasoning ability of these models on different domain reasoning tasks:

$$\Delta Pass@1 = Pass@1_x - Pass@1_y \quad (2)$$

Given the simplicity of the TTT-Bench tasks, we expect Pass@1 scores of SOTA LRMs to be higher relative to their performance on math benchmarks (positive $\Delta Pass@1$), especially for models that do well on math benchmarks. **Surprisingly, we observed the opposite – the majority of the evaluated models had lower Pass@1 scores on TTT-Bench compared to their Pass@1 scores on MATH 500 (↓ Avg $\Delta Pass@1$: -41.36%) & AIME 2024 (↓ Avg $\Delta Pass@1$: -4.88%).** In the case of MATH 500, which consists of high school math problems, the drops are significant relative to all the TTT-Bench tasks across all the models, especially for the small models. Whereas in the case of the olympiad-level math benchmark AIME 2024, LRMs performed well on the oTTT task relative to AIME, with an ↑ Avg $\Delta Pass@1$ of +8.56%, whereas they struggled on other TTT-Bench tasks. The $\Delta Pass@1$ trends also illustrate the relative task reasoning difficulties over the TTT-Bench tasks, which are consistent across $\Delta Pass@1$ comparisons against both MATH 500

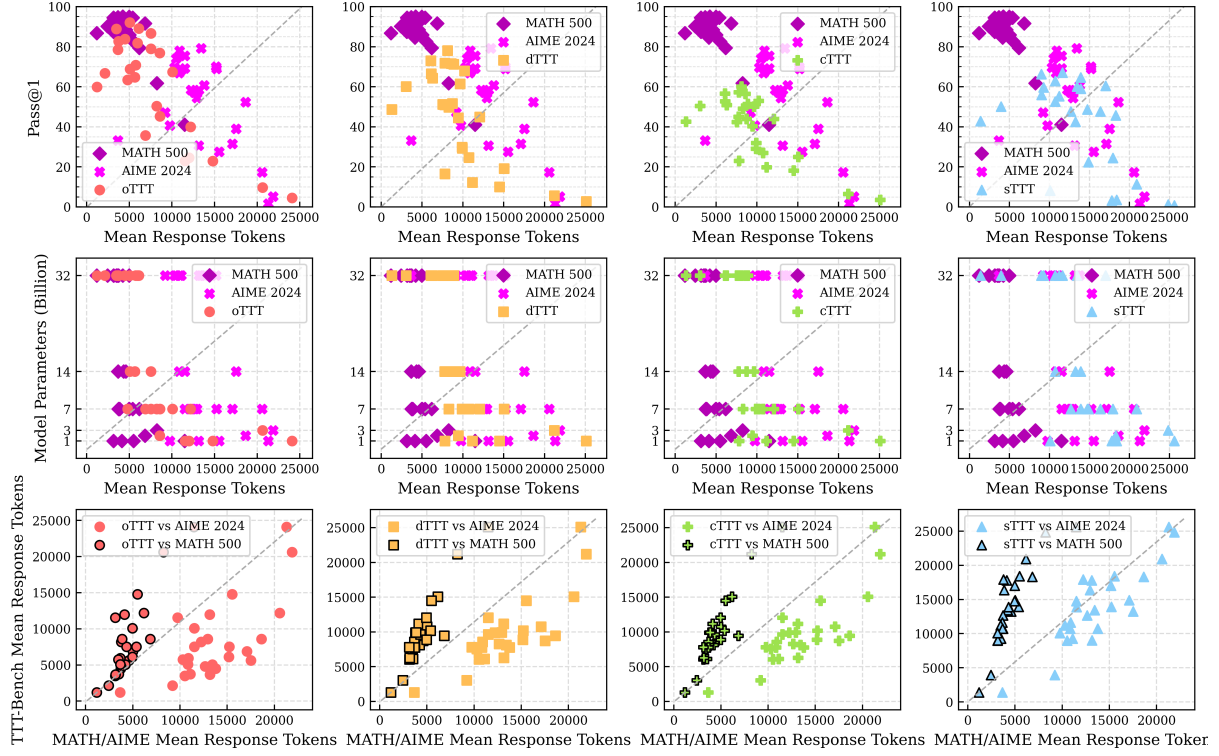


Figure 5: **Top:** Pass@1 performance of LRMs on MATH 500, TTT-Bench & AIME 2024 against the mean length of the corresponding solution responses. **Mid:** Model size against the corresponding model’s mean response lengths on MATH 500, TTT-Bench & AIME 2024. **Bottom:** The mean response lengths used for solving the TTT-Bench task against the response lengths for solving math benchmarks for different models.

& AIME 2024: $oTTT < dTTT < sTTT < cTTT$. We attribute this relative task difficulty order to the following two properties of these games: 1) Spatial constraints in the tasks, and 2) the number of possibilities to choose the next best move in the tasks. In Table 2, we also report the $Pass@1$ performance scores on TTT-Bench of flagship frontier models like o3-mini-medium & DeepSeek-R1. Due to resource constraints, we only evaluated these models with one response per test sample of TTT-Bench and reported their respective published numbers for MATH 500 & AIME 2024^{1 2}. From our results, we observe that even the frontier reasoning models find the TTT-Bench tasks as difficult as the standard math benchmarks. *These performance trends indicate poor reasoning ability of SOTA LRMs on TTT-Bench tasks that are simple, intuitive, and of low complexity, compared to their performance on both MATH 500 & AIME 2024 questions that are difficult and require relatively more knowledge and complex reasoning efforts.*

¹<https://openai.com/index/openai-o3-mini/>

²<https://huggingface.co/deepseek-ai/DeepSeek-R1>

Reasoning models struggle with simple long-term strategic reasoning: Since TTT-bench is generated using an automated pipeline that leveraged a heuristic-based algorithm (Appendix C) for generating the solutions and the respective game’s final verdict ("Win", "Blocked", or "Fork"), we investigate the type of reasoning tasks these LRMs are good at by analyzing their performance over these these individual solution verdict category questions. As shown in Figure 4, we observe that across the TTT-Bench tasks oTTT, dTTT, and sTTT, the performance of LRMs over questions that have the solution with verdict "Win" is consistently higher than the questions with verdict "Blocked", with the lowest for the questions with verdict "Fork". Whereas in the case of cTTT, the performance remained equally low for all the questions, independent of the solution verdict. This consistently high performance over questions with a solution verdict "Win" indicates that almost all the LRMs are capable of doing sound short-term thinking, where the solution is straightforward. On the contrary, a consistent dip in performance over slightly complex reasoning scenarios with

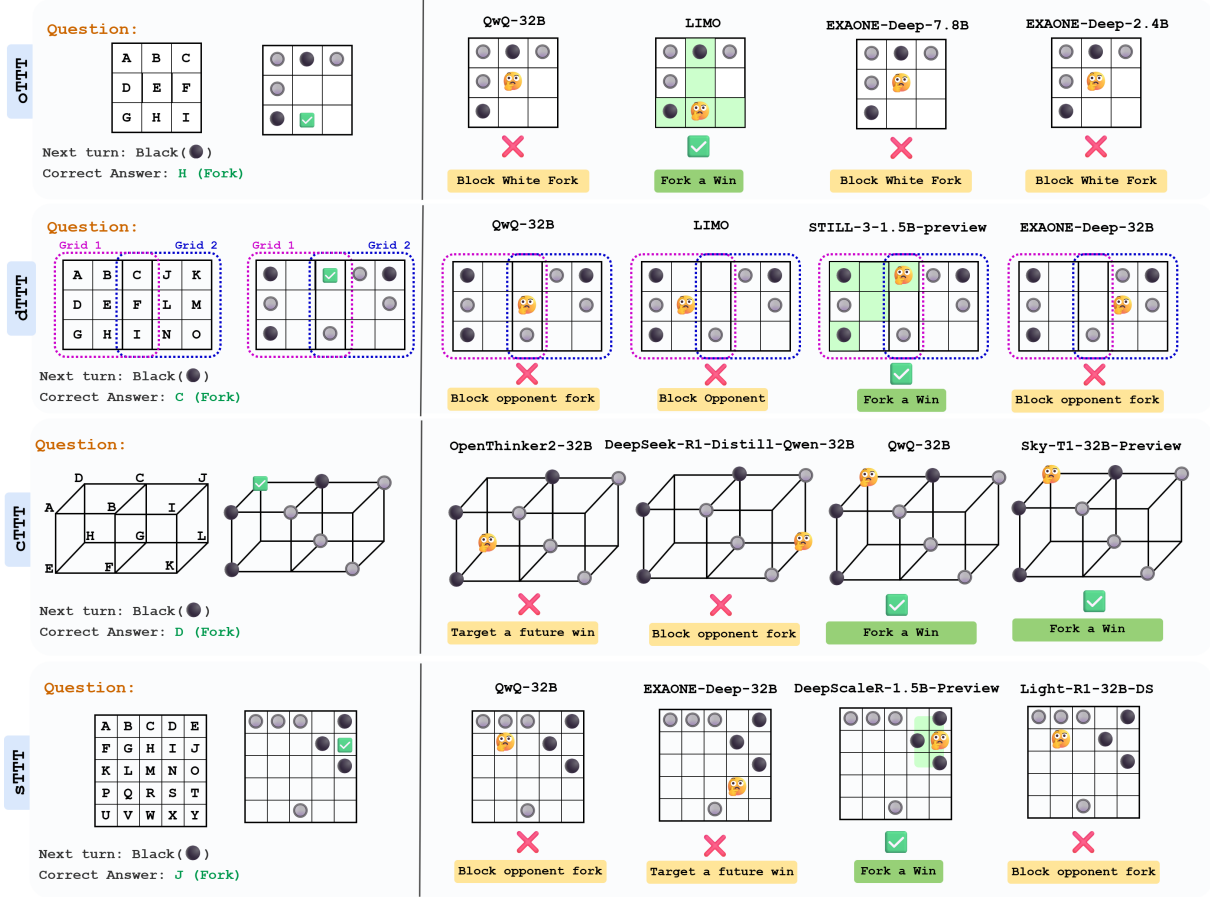


Figure 6: Examples of predicted solutions by SOTA LRMs for questions in TTT-Bench.

"Blocked" & "Fork" solution verdicts indicates that the LRMs struggle to successfully solve simple long-term strategic reasoning tasks where the models need to explore more possibilities and think strategically to either block the opponent's win or to identify a position which can lead to a long-term assured win.

LRMs overthink on TTT-Bench, with an increase in model size resulting in improved performance and efficient use of chain-of-thought:

Reasoning models are usually trained with long-context data to enable a long chain-of-thought (COT) that has proven to evoke emergent reasoning capabilities like self-reflection and backtracking (DeepSeek-AI, 2025). Ideally, a model should use a longer response length with more thinking tokens for questions deemed difficult and requiring a complex thinking process, whereas a shorter response length is used for simple and trivial questions (Jin et al., 2024). In Figure 5, we compare 1) the task performance of the models against the mean length of the corresponding solution responses (**Top**), 2) the model scale against the cor-

responding mean response lengths (**Mid**), and 3) mean response lengths used for solving TTT-Bench task against the mean response lengths for solving math benchmarks for different models (**Bottom**). From these plots, we observe that LRMs solve MATH 500 questions with higher accuracy and shorter COT, whereas in the case of both AIME and TTT-Bench questions (especially dTTT, cTTT, & sTTT), models used longer COT with high variability and lower performance. This is counter-intuitive, as the question is that TTT-Bench tasks are relatively simple, straightforward, and don't require as much thinking as is required to solve AIME questions. Consequently, we find that the larger models achieve higher performance using shorter COT across all the benchmarks, where the order of response lengths used for solving different benchmarks is: MATH 500 < TTT-Bench < AIME 2024. A head-to-head comparison between the generated solution response lengths for solving TTT-Bench tasks *vs* math benchmarks reveal that all the LRMs use longer COT for questions in TTT-Bench tasks against the MATH 500 questions, whereas they generate somewhat similar length COT when

compared against AIME questions (especially in the case of sTTT against AIME). These findings suggest that these reasoning models consume similar amount thinking tokens for solving TTT-Bench tasks (even though they are trivial for humans) as used for solving olympiad-level math questions, and are found to produce long COT with circular, inconclusive, and repetitive thinking process for these trivial questions (examples of solution traces are provided in Appendix B), further indicating their inability to do simple, intuitive, and straightforward tasks.

To further understand the failure modes of SOTA LRMs, in Figure 6 we illustrate a few question examples from TTT-Bench and the answers predicted by different models, with their main reasons for those predictions based on the reasoning traces. These examples shed light on the simplicity of our benchmarks as well as how these models reach incorrect or suboptimal TTT-Bench predictions. In the oTTT examples, majority of models predict position "E" as the next move, which although is a decent move to block the white (○) stones from forming a fork ("A-E-I" & "D-E-F"), but they fail to identify a better move ("H") that can guarantee a win for the current player by creating a fork for itself ("B-E-H" & "G-H-I") to ensure a win. Again, in the case of dTTT, cTTT & sTTT, the models tend to predict a move that blocks a forking possibility of the opponent, rather than thinking towards a long-term win by placing a move that establishes a fork for itself and ensures a win. **These examples demonstrate that most of the time, SOTA LRMs produce suboptimal solutions on TTT-Bench, and struggle at long-term strategic planning towards the optimal outcome.**

5 Conclusion

In this work, we introduce **TTT-Bench**, a novel benchmark designed to assess the reasoning capabilities of LRMs on intuitive and low-complexity reasoning tasks, which are effortless for humans to solve from a young age. While these models achieve remarkable success on challenging mathematical benchmarks like MATH 500 & AIME 2024, our evaluations reveal a consistent and surprising shortfall in their performance on simple two-player games, highlighting a fundamental gap in their reasoning capabilities. Through TTT-Bench, which is generated using our proposed automated and verifiable benchmark generation framework, we

demonstrate that existing LRMs tend to overthink trivial tasks, exhibit performance degradation with longer outputs, and often miss optimal strategies, even when they possess the underlying capabilities to solve significantly harder problems. This discrepancy underscores the importance of diversifying evaluation protocols for reasoning models to include intuitive and simple benchmarks beyond the conventional STEM-focused paradigms.

6 Limitations

While **TTT-Bench** offers new insights into the reasoning capabilities of LRMs, several limitations must be acknowledged:

Narrow game domain. TTT-Bench focuses exclusively on a specific class of simple two-player board games inspired by Tic-Tac-Toe. Although these games are intuitive and require basic spatial and strategic reasoning, they represent only a small slice of the broader spectrum of intuitive reasoning tasks. As such, the conclusions drawn from TTT-Bench may not directly generalize to other forms of intuitive reasoning, such as commonsense planning, physical reasoning, or visual-spatial puzzles.

Text-only formulation. All TTT-Bench tasks are presented in purely textual form, even though the reasoning involved is highly spatial. While this setup allows for scalable and consistent evaluation using language models, it may introduce unnecessary difficulty due to the lack of visual context. Some model failures might stem from difficulty in mentally visualizing board states rather than limitations in reasoning per se.

Model access and decoding constraints. Some evaluations—especially for closed-source or API models—are limited to a single sample per query, unlike open models where multiple generations are averaged. This discrepancy may affect the comparability of results and lead to under- or over-estimation of certain models' performance.

References

- Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2023. [A survey on evaluation of large language models](#). *Preprint*, arXiv:2307.03109.
- 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.
- Anthony Costarelli, Mat Allen, Roman Hauksson, Grace Sodunke, Suhas Hariharan, Carlson Cheng, Wenjie Li, Joshua Clymer, and Arjun Yadav. 2024. [Gamebench: Evaluating strategic reasoning abilities of llm agents](#). *Preprint*, arXiv:2406.06613.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- 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 *Advances in Neural Information Processing Systems*, volume 37, pages 28219–28253. Curran Associates, Inc.
- Nicoló Fontana, Francesco Pierri, and Luca Maria Aiello. 2024. [Nicer than humans: How do large language models behave in the prisoner’s dilemma?](#) *Preprint*, arXiv:2406.13605.
- 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.
- Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. [rstar-math: Small llms can master math reasoning with self-evolved deep thinking](#). *Preprint*, arXiv:2501.04519.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. [OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand. Association for Computational Linguistics.
- Jie Huang and Kevin Chen-Chuan Chang. 2023. [Towards reasoning in large language models: A survey](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenye Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. [The impact of reasoning step length on large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1830–1842, Bangkok, Thailand. Association for Computational Linguistics.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.
- LG AI Research. 2025. [Exaone deep: Reasoning enhanced language models](#). *arXiv preprint arXiv:2503.12524*.
- Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhiwei Li, Bao-Long Bi, Ling-Rui Mei, Junfeng Fang, Zhijiang Guo, Le Song, and Cheng-Lin Liu. 2025. [From system 1 to system 2: A survey of reasoning large language models](#). *Preprint*, arXiv:2502.17419.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. [Let’s verify step by step](#). In *The Twelfth International Conference on Learning Representations*.
- Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. 2025. [ZebraLogic: On the scaling limits of llms for logical reasoning](#). *Preprint*, arXiv:2502.01100.
- Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. 2025a. [Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct](#). *Preprint*, arXiv:2308.09583.
- Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025b. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. <https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2>. Notion Blog.
- Mathematical Association of America MAA. 2024. [American invitational mathematics examination](#).
- Yujun Mao, Yoon Kim, and Yilun Zhou. 2024. [Champ: A competition-level dataset for fine-grained analyses of llms’ mathematical reasoning capabilities](#). *Preprint*, arXiv:2401.06961.

- 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.
- OpenAI. 2025. Openai o3 and o4-mini system card.
- Zhanyue Qin, Haochuan Wang, Deyuan Liu, Ziyang Song, Cunhang Fan, Zhao Lv, Jinlin Wu, Zhen Lei, Zhiying Tu, Dianhui Chu, Xiaoyan Yu, and Dianbo Sui. 2024. [UNO arena for evaluating sequential decision-making capability of large language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7630–7645, Miami, Florida, USA. Association for Computational Linguistics.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. [GPQA: A graduate-level google-proof q&a benchmark](#). In *First Conference on Language Modeling*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Guangming Sheng, Chi Zhang, Zilinfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*.
- Parshin Shojaei*,†, Iman Mirzadeh*, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. [The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity](#).
- 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.
- Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.
- NovaSky Team. 2025a. Sky-t1: Train your own o1 preview model within \$450. <https://novasky-ai.github.io/posts/sky-t1>. Accessed: 2025-01-09.
- OpenThoughts Team. 2025b. Open Thoughts. <https://open-thoughts.ai>.
- Qwen Team. 2025c. [Qwq-32b: Embracing the power of reinforcement learning](#).
- Oguzhan Topsakal, Colby Jacob Edell, and Jackson Bailey Harper. 2024. [Evaluating large language models with grid-based game competitions: An extensible llm benchmark and leaderboard](#). *Preprint*, arXiv:2407.07796.
- Oguzhan Topsakal and Jackson B. Harper. 2024. [Benchmarking large language model \(llm\) performance for game playing via tic-tac-toe](#). *Electronics*, 13(8).
- Jen tse Huang, Eric John Li, Man Ho Lam, Tian Liang, Wenxuan Wang, Youliang Yuan, Wenxiang Jiao, Xing Wang, Zhaopeng Tu, and Michael R. Lyu. 2025. [How far are we on the decision-making of llms? evaluating llms’ gaming ability in multi-agent environments](#). *Preprint*, arXiv:2403.11807.
- Karthik Valmeekam, Kaya Stechly, Atharva Gundawar, and Subbarao Kambhampati. 2024a. [Planning in strawberry fields: Evaluating and improving the planning and scheduling capabilities of llm o1](#). *Preprint*, arXiv:2410.02162.
- Karthik Valmeekam, Kaya Stechly, and Subbarao Kambhampati. 2024b. [Llms still can’t plan; can llms? a preliminary evaluation of openai’s o1 on planbench](#). *Preprint*, arXiv:2409.13373.
- Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. 2025. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *arXiv preprint arXiv:2503.10460*.
- Fengli Xu, Qianye Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Si-jian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. 2025. [Towards large reasoning models: A survey of reinforced reasoning with large language models](#). *Preprint*, arXiv:2501.09686.
- Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024. [Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing](#). *Preprint*, arXiv:2406.08464.
- Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. [Limo: Less is more for reasoning](#). *Preprint*, arXiv:2502.03387.

- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2025. [Dapo: An open-source llm reinforcement learning system at scale](#). *Preprint*, arXiv:2503.14476.
- Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. [rest – mcts* : Llm self-training via process reward guided tree search](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 64735–64772. Curran Associates, Inc.
- Jiaxin Zhang, Zhong-Zhi Li, Ming-Liang Zhang, Fei Yin, Cheng-Lin Liu, and Yashar Moshfeghi. 2024b. [GeoEval: Benchmark for evaluating LLMs and multi-modal models on geometry problem-solving](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1258–1276, Bangkok, Thailand. Association for Computational Linguistics.
- Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024. [Marco-ol: Towards open reasoning models for open-ended solutions](#). *Preprint*, arXiv:2411.14405.

A TTT-Bench Examples

Here we provide the questions from TT-Bench and the solution illustrations for the examples in Figure 1. For the solutions, we also provide the verdict and the states contributing to the corresponding verdict.

oTTT test sample example in Figure 1

Question:

Alice and Bob are playing a game on a 3x3 grid. The points on the grid are labeled top to bottom, left to right, as A,B,C,D,E,F,G,H,I. Alice plays white. Bob plays black. At each turn, the player places a stone of the corresponding color onto one of the positions that has not been occupied. Whoever has three stones in a line (horizontal, vertical, or diagonal) wins. Alice first places a white stone at A. Bob places a black stone at B. Alice places a white stone at H. Bob places a black stone at C. Where should Alice play next?

Game Visualization:

Board

Positions:

A	B	C
D	E	F
G	H	I

Current

Board:

○	●	●
✓	○	✓

Solution:

G (Fork with A-D-G & G-H-I) or I (Fork with A-E-I & G-H-I)

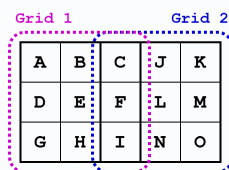
dTTT test sample example in Figure 1

Question:

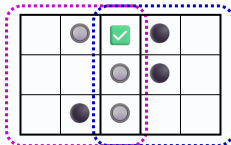
Alice and Bob are playing a game on two adjacent 3x3 grids. The points on the first grid are labeled top to bottom, left to right, as A,B,C,D,E,F,G,H,I. The points on the second grid are labeled top to bottom, left to right, as C,J,K,F,L,M,I,N,O. Note that points C,F, I are shared by the two grids. Alice plays white. Bob plays black. At each turn, the player places a stone of the corresponding color onto one of the positions that has not been occupied. Whoever has three stones in a line (horizontal, vertical, or diagonal) on either grid wins. Alice first places a white stone at I. Bob places a black stone at J. Then Alice at F. Bob places a black stone at L. Then Alice at B. Bob places a black stone at H. Where should Alice play next?

Game Visualization:

Board
Positions:



Current
Board:



Solution:

C (Win with C-F-I)

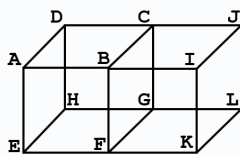
cTTT test sample example in Figure 1

Question:

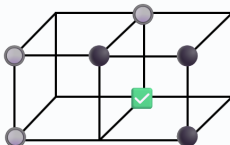
Alice and Bob are playing a game on two adjacent cubes. ABCD forms the top rectangle in the first cube, and BIJC forms the top rectangle in the second cube. EFGH forms the bottom rectangle in the first cube, and FKLG forms the bottom rectangle in the second cube. AE is an edge, BF is an edge, CG is an edge, DH is an edge, IK is an edge, and JL is an edge. Note that vertices B, C, G, F are shared by the two cubes. Alice and Bob play a game where they take turns to put stickers on the vertices of the cubes that have not been occupied. Alice plays white stickers. Bob plays black stickers. The person who has four stickers on the same plane on either cube wins. Alice first places a white stone at A. Then Bob places a black stone at B. Then Alice places a white stone at C. Then Bob places a black stone at I. Then Alice places a white stone at E. Then Bob places a black stone at K. Where should Alice play next?

Game Visualization:

Board
Positions:



Current
Board:



Solution:

G (Win with A-C-G-E)

sTTT test sample example in Figure 1

Question:

Alice and Bob are playing a game on a board. There are 5 equally spaced horizontal lines where the distance between two neighboring horizontal lines is 1. Similarly, there are 5 equally spaced vertical lines, and the distance between two neighboring vertical lines is 1. There are 25 intersection points between the 5 horizontal lines and 5 vertical lines. These 25 points are labeled from top to bottom, left to right, as A, B, C, D, . . . , Y. Alice plays white. Bob plays black. At each turn, the player places a stone of the corresponding color onto one of the 25 points that have not been occupied. Whoever has four stones that form either a unit square (with side length of 1) or a “diagonal square” with side length equal to the square root of 2 wins. For example, ABGF is a unit square. FBHL is a diagonal square. Alice first places a white stone at A. Then Bob places a black stone at D. Then Alice places a white stone at B. Then Bob places a black stone at K. Then Alice places a white stone at H. Then Bob places a black stone at U. Where should Alice play next?

Game Visualization:

Board

Positions:

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

Current Board:

○	○		●	
✓	✓	○		
●				
●				

Solution:

F (Fork with A-B-F-G & F-B-H-L) or G (Fork with A-B-G-F & A-C-H-G)

B Solution Trace Examples

Here we provide the reasoning traces produced by different models for question examples in Figure 1.

LIMO Solution Trace for oTTT question in Figure 6 (Correct)

Question:

Alice and Bob are playing a game on a 3x3 grid. The points on the grid are labeled top to bottom, left to right, as A,B,C,D,E,F,G,H,I. Alice plays white. Bob plays black. At each turn, the player places a stone of the corresponding color onto one of the positions that has not been occupied. Whoever has three stones in a line (horizontal, vertical, or diagonal) wins. Alice first places a white stone at A. Bob places a black stone at B. Alice places a white stone at C. Bob places a black stone at G. Alice places a white stone at D. Where should Bob play next? Let's think step by step and output the final answer within \boxed{ }.

Correct Answer: H (Forms fork with B-E-H & G-H-I)

Game Visualization:

Board
Positions:

A	B	C
D	E	F
G	H	I

Current
Board:



Fork a Win

Solution:

<think>

...

</think>

Therefore, if Bob plays at H, Alice must play at I to prevent Bob from winning. Then, Bob plays at F, leading to a draw. If Alice makes a mistake and plays at F, Bob wins. Therefore, playing at H is a good move because it puts pressure on Alice to play correctly. If she doesn't, Bob wins. If she does, the game is a draw.

Therefore, considering all options, Bob should play at H. This creates a threat that Alice must respond to, and if she responds incorrectly, Bob can win. If she responds correctly, the game is a draw, which is still a better outcome than losing.

Therefore, the best move for Bob is to play at H.

****Final Answer****

\boxed{H}

QwQ-32B Solution Trace for dTTT question in Figure 6 (Wrong)

Question:

Alice and Bob are playing a game on two adjacent 3x3 grid. The points on the first grid are labeled top to bottom, left to right, as A,B,C,D,E,F,G,H,I. The points on the second grid are labeled top to bottom, left to right, as C,J,K,F,L,M,I,N,O. Note that points C,F, I are shared by the two grids. Alice plays white. Bob plays black. At each turn, the player places a stone of the corresponding color onto one of the positions that has not been occupied. Whoever has three stones in a line (horizontal, vertical, or diagonal) on either grid wins. Alice first places a white stone at I. Bob places a black stone at G. Then Alice at J. Bob places a black stone at K. Then Alice at D. Bob places a black stone at A. Then Alice at M. Where should Bob play next? Let's think step by step and output the final answer within \boxed{ }.

Correct Answer: C (Forms fork with A-B-C & G-E-C)

Game Visualization:

Board Positions:

Grid 1			Grid 2	
A	B	C	J	K
D	E	F	L	M
G	H	I	N	O

Current Board:

●			●	●
●		👤		●
●		●		

✗

Block opponent fork

Solution:

<think>

...

</think>

Alice and Bob are playing a game on two adjacent 3x3 grids. The key steps and reasoning are as follows:

1. **Current Positions**:

- **Grid1**:

- Alice: I (bottom-right), D (middle-left)

- Bob: G (bottom-left), A (top-left)

- **Grid2**:

- Alice: J (top-middle), M (middle-right), I (bottom-left)

- Bob: K (top-right)

2. **Alice's Potential Threats**:

- In **Grid1**, Alice has D and I. To form a line, she needs a third stone in a row, column, or diagonal.

- In **Grid2**, Alice has J, M, and I. These positions do not form a line yet, but she could aim for a diagonal or column.

3. **Critical Blocking Moves**:

- **Grid1's Column3 (C, F, I)**: Alice has I. If she takes C and F, she can form a vertical line. Bob can block this by taking F.

- **Grid1's Row2 (D, E, F)**: Alice has D. Taking F would block her from completing this row.

- **Grid2's Diagonal (K-L-I)**: If Alice takes L, she could form a diagonal with I and L. However, Bob's K blocks this.

4. **Strategic Considerations**:

- Taking F in **Grid1** blocks both the vertical column3 and the horizontal row2 threats.

- Taking F also affects `**Grid2's column1**`, preventing Alice from forming a vertical line there.

After evaluating all possibilities, the optimal move for Bob is to place a stone at `**F**`, which blocks Alice's potential threats in both grids effectively.

`\boxed{F}`

DeepSeek-R1-Distill-Qwen-32B Solution Trace for cTTT question in Figure 6 (Wrong)

Question:

Alice and Bob are playing a game on two adjacent cubes. ABCD forms the top rectangle in the first cube and BIJC forms the top rectangle in the second cube. EFGH forms the bottom rectangle in the first cube and FKLJ forms the bottom rectangle in the second cube. AE is an edge, BF is an edge, CG is an edge, DH is an edge, IK is an edge, and JL is an edge. Note that vertices B,C,G,F are shared by the two cubes. Alice and Bob plays a game where they take turns to put stickers on the vertices of the cubes that have not been occupied. Alice plays white stickers. Bob plays black stickers. The person who has four stickers on the same plane on either cube wins. Alice first places a white stone at B. Then Bob places a black stone at A. Then Alice places a white stone at G. Then Bob places a black stone at C. Then Alice places a white stone at J. Then Bob places a black stone at E. Then Alice places a white stone at K. Where should Bob play next? Let's think step by step and output the final answer within `\boxed{}`.

Correct Answer: D (Forms fork with A-D-H-E & D-C-E-F)

Game Visualization:

Board

Positions:

Current Board:

Block opponent fork

Solution:

<think>

...

</think>

Alice and Bob are playing a game on two adjacent cubes. The goal is to place stickers on vertices such that four stickers of the same color form a plane on either cube. Alice goes first and places white stickers, while Bob places black stickers. The current moves are as follows:

1. Alice places a white sticker at B.
2. Bob places a black sticker at A.
3. Alice places a white sticker at G.
4. Bob places a black sticker at C.
5. Alice places a white sticker at J.
6. Bob places a black sticker at E.
7. Alice places a white sticker at K.

The vertices involved are A, B, C, D, E, F, G, H, I, J, K, L. The shared vertices are B, C, F, G. Alice's current stickers are at B, G, J, K. Bob's current stickers are at A, C, E. The remaining vertices are D, H, I, L, F.

To determine where Bob should place his next sticker, we need to consider the potential planes where Alice could win and block her. Alice is threatening multiple planes, but Bob can only block one.

After analyzing the potential planes and considering the current state of the game, Bob should place his next sticker at L to block Alice's potential win on the back face (C, G, L, J) and the bottom face (F, K, L, G). This move also blocks Alice from completing these planes, preventing her from winning on her next turn.

Thus, the final answer is:

`\boxed{L}`

QwQ-32B Solution Trace for sTTT question in Figure 6 (Wrong)

Question:

Alice and Bob are playing a game on a board. There are 5 equally spaced horizontal lines where the distance between two neighboring horizontal lines is 1. Similarly, there are 5 equally spaced vertical lines, and the distance between two neighboring vertical lines is 1. There are 25 intersection points between the 5 horizontal lines and 5 vertical lines. These 25 points are labeled from top to bottom, left to right, as A, B, C, D, . . . , Y. Alice plays white. Bob plays black. At each turn, the player places a stone of the corresponding color onto one of the 25 points that have not been occupied. Whoever has four stones that form either a unit square (with side length of 1) or a "diagonal square" with side length equal to the square root of 2 wins. For example, ABGF is a unit square. FBHL is a diagonal square. Alice first places a white stone at A. Then Bob places a black stone at E. Then Alice places a white stone at B. Then Bob places a black stone at I. Then Alice places a white stone at C. Then Bob places a black stone at O. Then Alice places a white stone at W. Where should Bob play next? Let's think step by step and output the final answer within `\boxed{}`.

Correct Answer: J (Forms fork with D-E-J-I & I-J-O-N)

Game Visualization:

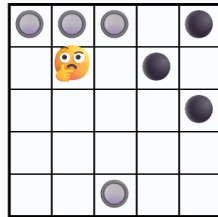
Board

Positions:

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

Current

Board:



Block opponent fork

Solution:

<think>

...

</think>

Alice and Bob are playing a game on a 5x5 grid where the goal is to form a unit square (side length 1) or a diagonal square (side length square root 2) with four stones. Alice's moves so far are at A, B, C, and W, while Bob's moves are at E, I, and O.

Key considerations:

1. **Alice's stones**: A(0,0), B(0,1), C(0,2), W(4,2).
2. **Bob's stones**: E(0,4), I(1,3), O(2,4).
3. **Threat analysis**: Alice's top-row stones (A, B, C) could form part of a unit square if combined with stones below. The critical points for such squares include G(1,1) and H(1,2).
4. **Blocking potential squares**: The unit square involving A, B, and needing F(1,0) and G(1,1) is a key threat. Since Alice has A and B, blocking G(1,1) would prevent her from completing this square in future moves.

After evaluating all potential squares and critical points, the most strategic move for Bob to block Alice's potential threats is to place at G(1,1).

\boxed{G}

C Algorithm for generating TTT-Bench solutions

On the next page, we provide the detailed algorithm used for the automated generation of TTT-Bench questions and their corresponding solutions.

Algorithm 1: Algorithm for generating TTT-Bench solutions

Total number of positions: T , **Number of moves played:** $N = n_c + n_o$, **Number of positions in winning lines:** n ,
Number of winning lines: t , **Current player positions:** $P^c : \{p_1^c, \dots, p_{n_c}^c\}$, **Opponent positions:** $P^o : \{p_1^o, \dots, p_{n_o}^o\}$,
Available Position: $A : \{a_1, \dots, a_{T-N}\}$, **Winning States:** $W : \{(w_1^1, \dots, x_n^1), \dots, (w_1^t, \dots, x_n^t)\}$

Function check_won(P_{new}^c):
 for $(w_1^i, \dots, x_n^i)$ **in** W **do**
 if $\forall (w_j^i \in P_{new}^c | w_j^i \in (w_1^i, \dots, x_n^i))$ **then**
 return True;
 end
 end
 return False;

Function get_wins(P^c, A):
 $w_moves \leftarrow \{\}$
 for a_i **in** A **do**
 $P_{new}^c \leftarrow P^c + a_i$
 if check_won(P_{new}^c) **then**
 $w_moves \leftarrow w_moves + a_i$
 end
 end
 return w_moves ;

Function get_forks(P^c, A):
 $f_moves \leftarrow \{\}$
 for a_i **in** A **do**
 $P_{new}^c \leftarrow P^c + a_i, A^{potential} \leftarrow a_j \in A; a_j \neq a_i, threat_count \leftarrow 0$
 for $(w_1^i, \dots, x_n^i)$ **in** W **do**
 $W^{current} \leftarrow w_j^i | w_j^i \in (w_1^i, \dots, x_n^i) \ \& \ w_j^i \in P_{new}^c$
 $W^{opponent} \leftarrow w_j^i | w_j^i \in (w_1^i, \dots, x_n^i) \ \& \ w_j^i \notin P_{new}^c \ \& \ w_j^i \notin A^{potential}$
 if $len(W^{current}) == n - 1 \ \& \ len(W^{opponent}) == 0$ **then**
 $threat_count \leftarrow threat_count + 1$
 end
 end
 if $threat_count \geq 2$ **then**
 $f_moves \leftarrow f_moves + a_i$
 end
 end
 return f_moves ;

Function get_solution(P^c, P^o, A):
 $w_moves \leftarrow$ get_wins(P^c, A) ▷ Check if current player has wining moves.
 if $len(w_moves) \geq 1$ **then**
 return $w_moves, "Win"$;
 end
 $opp_w_moves \leftarrow$ get_wins(P^o, A) ▷ Check if current player has blocking moves.
 if $len(opp_w_moves) == 1$ **then**
 return $opp_w_moves, "Blocked"$;
 end
 else
 return None;
 end
 $f_moves \leftarrow$ get_forks(P^c, A) ▷ Check if current player has forking moves.
 if $len(f_moves) \geq 1$ **then**
 return $f_moves, "Fork"$;
 end

D Performance Results

In Table 3 and Table 4 we report the Pass@1 performance numbers used in Figure 3 and Figure 4 respectively.

Models	oTTT	dTTT	cTTT	sTTT	AIME 2024	MATH 500
QwQ-32B	92.03	78.13	60.14	60.10	79.17	95.00
QwQ-32B-Preview	66.73	60.13	50.49	50.16	47.08	90.16
EXAONE-Deep-32B	89.09	71.25	51.46	60.42	70.00	94.95
OpenThinker2-32B	88.85	73.00	56.60	66.30	74.17	94.53
DeepSeek-R1-Distill-Qwen-32B	70.77	51.19	42.36	55.31	69.58	93.28
s1.1-32B	83.76	71.81	57.22	62.86	60.63	92.46
Light-R1-32B-DS	81.80	71.56	56.46	67.03	77.92	94.69
LIMO	78.55	64.31	50.83	55.89	58.33	91.91
OpenThinker-32B	82.66	66.56	52.71	59.69	68.33	93.83
Sky-T1-32B-Preview	59.87	48.69	42.57	42.86	33.13	86.75
DeepSeek-R1-Distill-Qwen-14B	68.81	50.94	45.00	52.66	66.88	93.45
Light-R1-14B-DS	78.92	61.50	50.14	59.06	75.42	94.04
s1.1-14B	64.77	51.69	43.89	42.50	38.96	88.38
OpenThinker2-7B	63.48	49.75	49.10	47.66	57.08	92.16
EXAONE-Deep-7.8B	86.64	67.94	52.99	64.64	68.96	94.54
AceMath-RL-Nemotron-7B	67.40	44.94	43.89	48.65	69.17	93.76
DeepSeek-R1-Distill-Qwen-7B	45.22	29.25	31.88	47.66	54.38	92.05
OpenThinker-7B	35.66	24.63	27.01	22.40	31.46	85.90
Light-R1-7B-DS	50.31	29.81	29.03	24.48	58.33	92.18
s1.1-7B	39.95	19.19	25.00	11.35	17.29	79.45
EXAONE-Deep-2.4B	76.84	44.50	40.35	45.68	52.29	91.60
s1.1-3B	9.68	5.63	6.53	1.20	5.21	61.81
DeepSeek-R1-Distill-Qwen-1.5B	22.92	10.06	18.19	3.49	27.50	82.58
DeepScaleR-1.5B-Preview	23.04	16.50	22.99	8.18	40.63	87.43
s1.1-1.5B	4.47	2.69	3.47	0.78	1.67	41.10
STILL-3-1.5B-preview	24.51	12.25	19.79	3.18	30.63	84.59

Table 3: Pass@1 performance of LRMs on TTT-Bench, AIME 2024, and MATH500.

Models	oTTT			dTTT			cTTT			sTTT		
	Win	Blocked	Fork	Win	Blocked	Fork	Win	Blocked	Fork	Win	Blocked	Fork
QwQ-32B	98.12	95.77	82.39	96.09	95.62	51.41	63.12	72.29	49.53	75.62	91.25	36.77
QwQ-32B-Preview	85.31	53.12	53.69	83.12	49.69	42.34	44.69	51.25	52.81	77.08	53.33	35.1
EXAONE-Deep-32B	99.84	94.3	67.61	95.62	90.31	37.34	42.19	66.46	44.84	96.46	66.46	39.38
OpenThinker2-32B	99.38	95.77	66.19	95.47	95.31	39.38	49.69	68.75	50.94	92.29	86.25	43.33
DeepSeek-R1-Distill-Qwen-32B	87.5	67.46	51.99	70.94	55.31	29.38	43.75	45.0	39.69	87.92	69.79	31.77
s1.1-32B	97.03	82.72	65.62	94.38	83.12	43.59	52.19	63.96	54.69	86.88	78.12	43.23
Light-R1-32B-DS	95.78	81.43	61.93	95.0	85.62	41.09	55.62	66.25	49.53	88.75	89.38	45.0
LIMO	93.75	75.0	61.36	86.25	63.75	42.66	36.56	59.17	51.72	82.71	71.46	34.69
OpenThinker-32B	95.47	87.5	58.81	92.34	82.81	32.66	42.5	66.25	47.66	90.21	75.42	36.56
Sky-T1-32B-Preview	76.88	46.32	51.42	65.47	51.25	30.63	33.12	44.79	45.62	62.71	44.79	31.98
DeepSeek-R1-Distill-Qwen-14B	85.0	65.26	49.72	75.31	52.19	25.94	29.69	51.46	47.81	82.5	62.92	32.6
Light-R1-14B-DS	94.06	78.86	59.38	91.56	70.62	26.88	42.19	56.46	49.38	91.25	73.96	35.52
s1.1-14B	83.44	60.48	42.33	76.88	46.56	29.06	26.25	49.58	48.44	65.0	51.04	26.98
OpenThinker2-7B	83.28	57.17	41.48	73.75	35.0	33.12	31.87	54.17	53.91	77.29	47.92	32.71
EXAONE-Deep-7.8B	96.09	94.85	64.2	90.0	77.19	41.25	38.75	57.92	56.41	95.0	86.25	38.65
AceMath-RL-Nemotron-7B	87.5	59.93	47.16	71.09	36.25	23.12	44.06	38.12	48.12	82.92	65.0	23.33
DeepSeek-R1-Distill-Qwen-7B	57.66	34.56	40.91	44.38	19.38	19.06	27.81	30.0	35.31	57.92	28.96	14.17
OpenThinker-7B	43.44	31.62	30.4	35.47	15.31	18.44	17.81	21.25	35.94	31.46	21.04	18.54
Light-R1-7B-DS	71.72	36.76	35.51	47.34	21.25	16.56	24.06	27.92	32.34	52.71	24.17	10.52
s1.1-7B	50.16	36.58	29.83	25.16	15.0	15.31	17.19	21.04	31.87	15.83	9.58	10.0
EXAONE-Deep-2.4B	77.97	97.79	53.41	57.5	56.25	25.62	24.06	53.33	38.75	54.58	86.67	20.73
s1.1-3B	9.53	7.72	13.07	5.62	3.12	6.88	5.31	2.92	9.84	1.46	0.62	1.35
DeepSeek-R1-Distill-Qwen-1.5B	28.59	18.01	21.31	10.0	6.88	11.72	15.0	14.37	22.66	5.83	3.96	2.08
DeepScaleR-1.5B-Preview	40.62	8.46	15.62	19.69	11.56	15.78	22.19	17.08	27.81	14.17	6.04	6.25
s1.1-1.5B	3.44	4.41	6.25	3.28	0.62	3.12	12.81	13.12	28.28	1.04	1.46	0.31
STILL-3-1.5B-preview	31.09	16.54	24.43	15.00	7.19	12.03	12.81	13.12	28.28	6.04	1.88	2.40

Table 4: Pass@1 performance of LRMs over individual solution verdict category ("Win", "Blocked", & "Fork") questions in TTT-Bench tasks.