# SUDOKU-BENCH: Evaluating creative reasoning with Sudoku variants

**Jeffrey Seely**[1] jeffrey@sakana.ai
**Yuki Imajuku**[1] imajuku@sakana.ai
**Tianyu Zhao**[1] tianyu@sakana.ai
**Edoardo Cetin**[1] edo@sakana.ai

**Llion Jones**[1] llion@sakana.ai

[1]Sakana AI, Japan

## Abstract

Existing reasoning benchmarks for large language models (LLMs) frequently fail to capture authentic creativity, often rewarding memorization of previously observed patterns. We address this shortcoming with SUDOKU-BENCH, a curated benchmark of challenging and unconventional Sudoku variants specifically selected to evaluate creative, multi-step logical reasoning. Sudoku variants form an unusually effective domain for reasoning research: each puzzle introduces unique or subtly interacting constraints, making memorization infeasible and requiring solvers to identify novel logical breakthroughs ("break-ins"). Despite their diversity, Sudoku variants maintain a common and compact structure, enabling clear and consistent evaluation. SUDOKU-BENCH includes a carefully chosen puzzle set, a standardized text-based puzzle representation, and flexible tools compatible with thousands of publicly available puzzles—making it easy to extend into a general research environment. Baseline experiments show that state-of-the-art LLMs solve fewer than 15% of puzzles unaided, highlighting significant opportunities to advance long-horizon, strategic reasoning capabilities.

## 1 Introduction

Large-scale language models excel at short-form deduction [12, 29], yet genuinely *creative* reasoning remains elusive. Many standard benchmarks, where current models already rival or surpass human performance [8, 22, 6], often reward the memorization of solution templates [2]. Once these templates are implicitly memorized, incremental accuracy gains offer limited insight into a model's capacity for novel reasoning. Benchmarks such as ARC [3] effectively resist memorization; however, their solutions, while novel to models, remain straightforward for humans, insufficiently capturing the depth of human creative reasoning.

We propose Sudoku variants (Fig. 1) as a unique domain addressing this gap. A Sudoku variant is a logical puzzle defined by a partially filled $n \times n$ grid, accompanied by visual constraints and even a problem-specific set of rules that can only be described in natural language. Yet, each puzzle still admits a unique solution—an $n \times n$ grid fulfilling its constraints. Puzzle creators introduce original rules or combine common constraints in novel ways. Hundreds of user-submitted Sudoku variants are published daily on platforms like Logic Masters Germany [1], deliberately designed to require *creative* insights and subtle logical breakthroughs. Such puzzles precisely target the type of novel, multi-step reasoning that memorization-focused and even popular reasoning benchmarks fail to consistently measure [31].

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

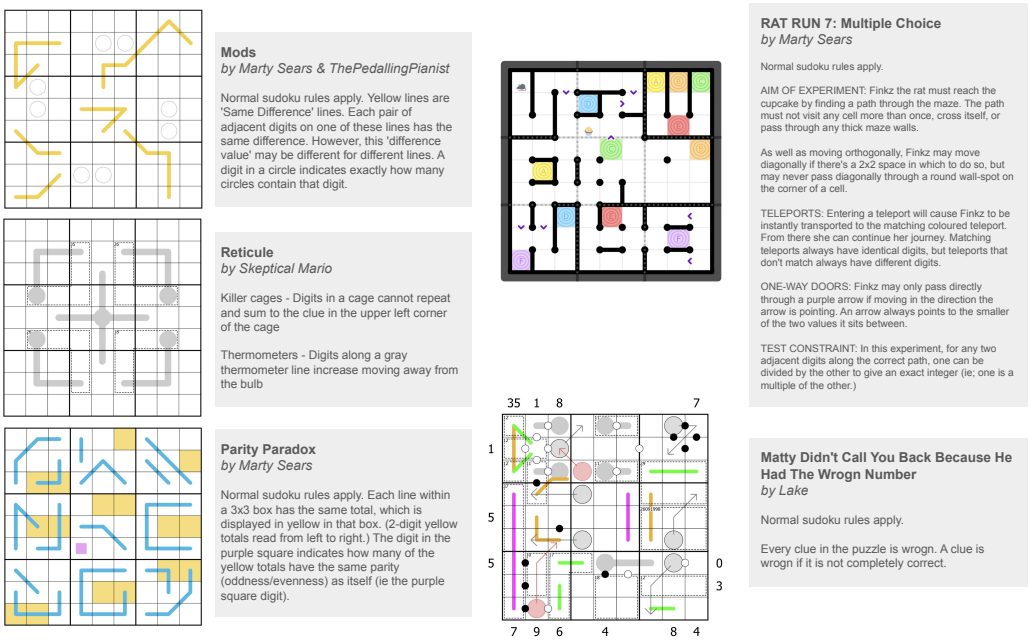



Figure 1: Each Sudoku variant has a unique set of constraints explicitly described in the puzzle rules. Puzzles may feature whimsical rules such as in *Rat Run*, or meta-level constraints, such as requiring all standard Sudoku rules to be intentionally violated.

This paper's contribution is twofold. First, we introduce open-source tools interfacing directly with the popular puzzle application **SudokuPad** [17], facilitating both agentic tool-use interaction and standardized textual puzzle representations. The agentic interaction provides an API to fetch images of the current board state and access to all the annotation tools available in **SudokuPad** that human solvers usually rely on. Our textual format isolates logical reasoning from visual processing, enabling effective evaluation with current language models. Second, we present SUDOKU-BENCH, a carefully curated benchmark of 100 Sudoku variants, selected in collaboration with hosts from the *Cracking the Cryptic* YouTube channel. These puzzles span a wide range of difficulties and reasoning styles, deliberately chosen to test model performance across diverse logical pathways and puzzle-specific "break-ins."

Our experiments showcase SUDOKU-BENCH poses a striking challenge for current state-of-the-art models. Without tool assistance, even the strongest publicly available LLM evaluated solves fewer than 15% of the benchmark. Notably, most of the successful completions come from the simplest subset of $4 \times 4$ puzzles, with performance rapidly collapsing with larger and less conventional grids. This is observed in both the one-shot configuration (prompt a model to solve a puzzle in one response) and a multi-step configuration (multi-turn interaction between the model providing at least one digit and the user providing the updated board state).

Beyond benchmarking, Sudoku variants offer a fertile *laboratory* for reasoning research. An extensive, ever-growing supply of human-generated puzzles allows scalable difficulty progression, from simpler $4\times4$ puzzles suitable for small models to highly intricate $9\times9$ puzzles, the hardest of which can stump all but the best expert human solvers. Rich auxiliary data, including detailed expert solution transcripts and interaction traces, facilitate imitation learning. We include, as part of SUDOKU-BENCH thousands of hours of reasoning transcripts and actions taken when solving from *Cracking the Cryptic*, a popular YouTube channel dedicated to detailed demonstrations of solving Sudoku variants with over 250M views. This data is entirely available for researchers who wish to explore supervised approaches to learn and fine-tune models from human reasoning – qualitatively far beyond the depth and diversity of synthetic reasoning datasets with current state-of-the-art language models [11, 16].

The remainder of this paper proceeds as follows: Section 2 surveys Sudoku variants and their reasoning demands. Section 3 details the SUDOKU-BENCH dataset, text interface, and evaluation

framework. Section 4 presents baseline results and analyses of model failure modes. We review related work in Section 5, and conclude with open research directions in Section 6.

## 2 Background: Sudoku Variants

Traditional Sudoku involves completing a $9 \times 9$ grid such that each digit from 1 to 9 appears exactly once in every row, column, and $3 \times 3$ subgrid. This structure provides a foundation for numerous variants that introduce additional constraints. For instance, *Killer Sudoku* combines elements of Sudoku and Kakuro, requiring digits within outlined cages to sum to specified totals without repeats. *Thermometers* are paths of adjacent cells where digits must increase monotonically. Digits along *arrows* must sum to the digit in the circled cell at the base. *Kropki* dots between cells indicate specific relationships, such as consecutive numbers or a 1:2 ratio.

The availability of web-based puzzle-making tools allowed puzzle authors to invent their own variants. In early 2020, the puzzle-hosting site Logic Masters saw a surge in the number of puzzles posted. As of May 2025, more than 27,000 user-submitted variants are published on the site [1].

Puzzle creators frequently combine multiple constraints in unique ways. Often, these combined constraints result in puzzles starting with minimal or no digits, necessitating extensive logical reasoning to determine the initial placement, termed a "break-in." Such puzzles require solvers to meticulously explore the interaction of constraints, significantly diverging from the eager guessing often observed in reasoning LLMs (Section 4).

Beyond these standard constraint types, puzzle setters often employ meta-constraints, which involve deducing puzzle-specific parameters (e.g., "digits in a cage sum to an unknown value to be determined by solving," or "the line must be identified as either a palindrome or a renban sequence"). These meta-constraints add another layer of complexity and creative reasoning.

Puzzle authors are ultimately limited only by imagination, often developing whimsical and novel rulesets (e.g., puzzles themed around rats in mazes (Fig. 1)). Crucially, all Sudoku variants maintain a structured format: an $n \times n$ grid, natural-language puzzle rules, visual elements easily encoded as text, and a single unique solution. This structured yet flexible framework makes Sudoku variants exceptionally suitable for systematically investigating creative reasoning capabilities, meaning that the puzzles are very diverse and challenging but grounded and easy to verify if correct.

**Puzzle example: Ascension** We illustrate some of these features with an example. Figure 2a highlights the novel interaction between a knight's move restriction and arrow constraints.

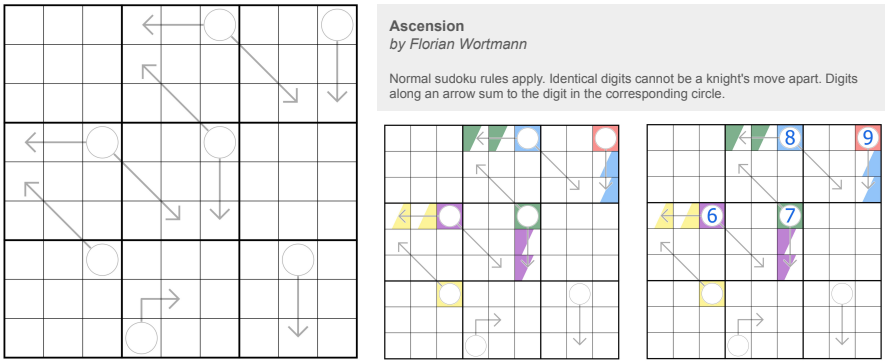
To find the puzzle's break-in, the solver must make three observations.

First, whatever the digit highlighted in green (`r4c6`, box 5), it must occur somewhere in box 2, but not in column 6 (by standard Sudoku rules), or along its arrow tip, or a knight's move away, thus can only occur in one of the two half-shaded cells `r1c4` or `r1c5`. This same pattern applies to the other cell groups highlighted by the other colors shown in the middle panel. The second observation is that since digits on the arrow must be smaller than the corresponding circled base, this creates a long-range chain dependency across the highlighted cells, namely, the circled cells shaded yellow, purple, green, blue, then red, must be monotonically increasing. This is a key insight but not enough to determine an exact digit yet.

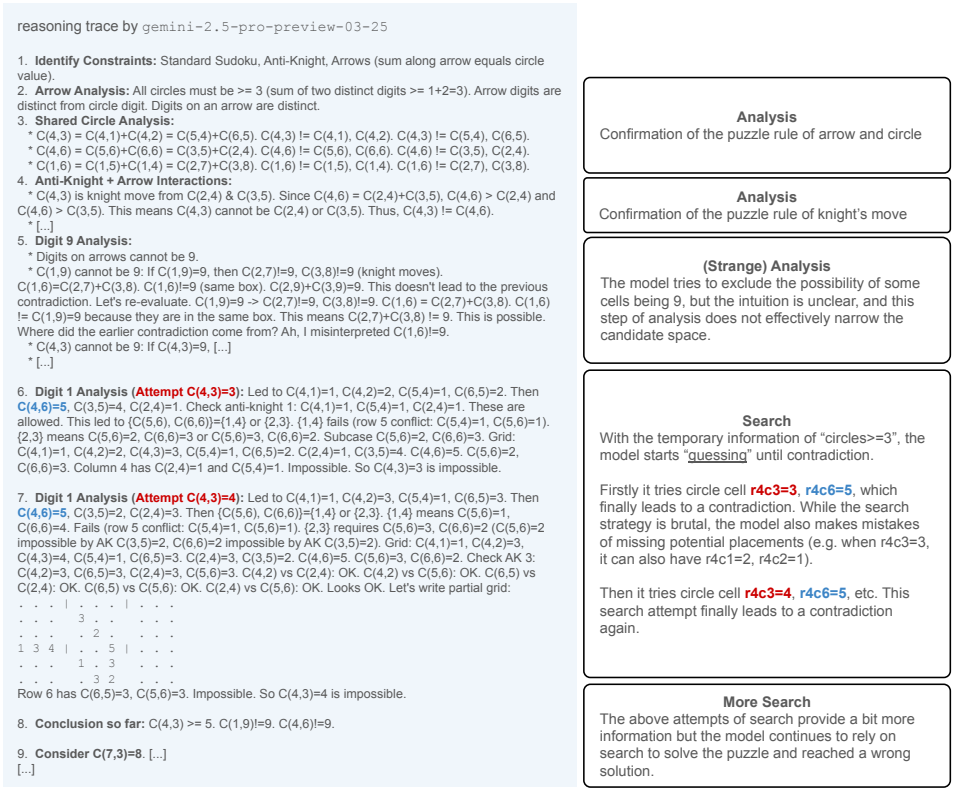
The third observation is that the purple cell must be the sum of three Sudoku digits, the two in its arrow tip `r4c1` and `r4c2`, but one of which is equal to the yellow cell of `r7c3`, which itself is the sum of two Sudoku digits by arrow rules. The only digit that can be the sum of three Sudoku digits and leave enough room for the monotonic chain along green, blue and red, is six. Therefore `r4c6` must be six and the subsequent digits in the monotonic chain are forced (right panel).

In a video demonstrating this puzzle solve, an expert solver discovered this break-in in about 4.5 minutes, and a full puzzle solve taking about 35 minutes.[1] In all LLMs we tested, no model was able to make progress. For example, we show the reasoning summary of `Gemini 2.5 Pro Preview` (Fig. 2b), which was able to successfully parse and identify the puzzle constraints, but quickly resorts to guesswork and search. This highlights that there is still a gap between how LLMs reason and how humans prefer to reason; LLMs can rely on brute-force but humans will prefer to save time

[1] `https://www.youtube.com/watch?v=-7OR_IK4Th8`




(a) Example of a logical break-in from the puzzle *Ascension*. Despite the initial grid being empty, the puzzle constraints collectively enforce a strict sequence of increasing digits from the lower left to the upper right shaded circles. Identifying and leveraging this subtle interplay forms the puzzle's essential insight. Though rated at an easy difficulty (2-star), current LLMs consistently fail to identify this logical entry point.




(b) `Gemini 2.5 Pro Preview`'s attempt to solve the puzzle *Ascension*. In contrast to the break-in by a human solver, the model failed to effectively narrow its search space and had to rely on a more brute-force search strategy, which did not lead to the correct solution.

Figure 2: *Ascension* example.

and energy by using precise logic to find shortcuts to correct digits. We hope to see this benchmark encouraging work on creating LLMs that reason in a more 'human-like' manner.

The *Ascension* example highlights two facets of Sudoku variants. First, although both knight-move and arrow constraints are commonplace, this specific interaction is unique to this particular puzzle. Therefore, the memorization-resistance of Sudoku variants is not exclusively due to the inclusion of novel rulesets; familiar constraints can induce a solving tactic never seen before. Indeed, some of the most difficult puzzles adopt deceptively simple rulesets. The second point is that for puzzles with few

or no given digits (as is common in variants), the search space is too large for initial guesswork to be effective. This also often necessitates a kind of meta-reasoning where one must decide at the outset what reasoning techniques should be applied, e.g. the use of coloring, set theory or looking at digit parity.

This pattern of needing to spend time at the beginning to understand how the constraints interact in a new manner is normal when humans tackle these puzzles. This also means that some of these initial deductions remain pertinent throughout the solve, meaning that in order to robustly solve some of these puzzles over 100s of steps will either require a form of memory, like a scratchpad, or a very long context window.

# 3 SUDOKU-BENCH: Dataset and Benchmark Design

We sought to select 100 puzzles that are representative of the breadth of Sudoku variants. To establish a graded evaluation curve, we selected 15 $4 \times 4$ puzzles, 15 $6 \times 6$ puzzles, and 70 $9 \times 9$ puzzles. The 15 $4 \times 4$ puzzles are included, in part, to measure progress in even modestly sized language models. Fifty of the $9 \times 9$ puzzles were curated by the hosts of *Cracking the Cryptic* exclusively for this benchmark. The selected puzzles evenly span difficulty ratings from novice-friendly "1-star" puzzles to expert-level "5-star" challenges that may require hours of careful analysis before any digits can be confidently placed. Twenty of the puzzles are difficult vanilla Sudokus, which were supplied by the puzzle company Nikoli, which popularized Sudoku in the 1980s. We aimed to create a smooth ramp in complexity such that an initial attempt at tackling the benchmark can yield some early success, but fully solving it will be vary challenging, and we hope that this benchmark will resist being solved for a significant time span.

**Text descriptions** Each puzzle is given a pure text representation. For instance, Fig 3 shows a simple $4 \times 4$ puzzle whose line paths are represented as a sequence of `rxcy` (row `x` column `y`) coordinates, and the location of the dot is described as the two cells it lies between. The rules, visual elements, grid size, and initial board state (if any digits are given) are sufficient to unambiguously specify the puzzle and converted into a prompt.

While some of the most recent reasoning models have shifted toward multimodal inputs, we found that most, including OpenAI's o3 model, struggle in converting $9 \times 9$ puzzles into accurate coordinates. Puzzle benchmarks such as Enigma [27] and VGRP [23] emphasize the visual aspect of puzzles and require multimodal models. Given that current frontier models still struggle in exact specification of the visual elements of Sudoku puzzles, we opted to specify all elements precisely in text to isolate the creative reasoning process itself from visual understanding.

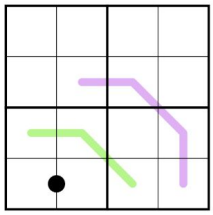
Each puzzle's text representation has been precomputed for puzzles on SUDOKU-BENCH. We provide the code for extracting text descriptions from a puzzle specified in SudokuPad, allowing researchers to utilize this harness in other puzzles.

Note that many of the puzzles would benefit from visual reasoning, some even potentially requiring it, since many of the break-ins are geometric and use symmetry, or have some rules that reference the shapes in the puzzle. Some puzzles can be very visually dense (See Bottom-Right in Fig 1) and current vision model we tested are not powerful enough to extract all the features, like the small numbers. We suspect that solving this benchmark using vision would represent a significant improvement over current multimodal LLMs.

## 3.1 Expert reasoning traces

A core question is whether advancing reasoning capabilities in LLMs can benefit from adopting more "human-like" thinking. In reinforcement learning models, pretraining on human supervision is common, while other work has shown that RL from scratch yields better performance in contained environments [25, 9, 14, 18]. Vanilla Sudoku is an interesting domain in that the strategies that humans use differ so significantly from search-based solvers [21], and this effect is especially pronounced in Sudoku variants.

The YouTube channel *Cracking the Cryptic* offers a particularly unique opportunity to explore the benefits of imitation learning. The channel contains over 3,000 published videos demonstrating the solving process of Sudoku variants. Notably, the hosts must verbally describe their thinking process,



**Differences Count - part 1**
*by Sujoyku and Marty Sears*

**Initial grid:**
. . . .
. . . .
. . . .
. . . .

**Visual elements:**
- line, color: pale green, coords: r1c1 r2c2 r4c3
- line, color: plum, coords: r2c2 r2c3 r3c4 r4c4
- circle, color: black, location: between r4c1 and r4c2

**Rules:**
Normal 4x4 sudoku rules apply.
Digits separated by a black dot are in a 1:2 ratio.
The difference between two adjacent digits on a coloured line indicates exactly how many pairs of adjacent digits along that line have that exact difference.

Figure 3: A text representation of a puzzle. The rules, initial grid, and a text description of visual elements are sufficient to unambiguously specify the puzzle.

explaining to the viewer each logical deduction. A typical puzzle takes the hosts around 60 minutes to solve, while some of the more difficult puzzles featured on the channel are over 3 hours in length. We developed a dataset consisting of the audio transcripts of each solve, together with a sequence of SudokuPad actions extracted from the video. The actions were extracted using a machine learning model trained on ground truth actions simulated on SudokuPad and then applied to video frames. This dataset is hosted on HuggingFace[2] under an MIT license in agreement with the hosts of the channel.

### 3.2 Dataset format

The SUDOKU-BENCH puzzle dataset[3] contains three subsets, `challenge_100`, `nikoli_100`, and `ctc`. The `challenge_100` is described above and represents the core benchmark. Additional puzzle data include `nikoli_100`, a collection of hand-made vanilla Sudokus supplied by Nikoli for this benchmark (20 of which are featured in `challenge_100`). The `nikoli_100` are designed to highlight creative or human-like reasoning in their solution paths, and may be applicable to many of the research approaches that use vanilla Sudoku as a testbed (Section 5). The `ctc` includes 2,565 Sudoku variants that have been solved on *Cracking the Cryptic*. Due to the breadth and variety of Sudoku variants, the text representation of each puzzle in `ctc` has not undergone manual checking, and an unambiguous representation of the board would require a screenshot in some cases.

### 3.3 SudokuPad environment

We also provide tools for interacting with SudokuPad in an agentic environment. SudokuPad enables common note-taking strategies used by human solvers, including color-coding cells (as in Fig. 2a) or providing candidate digits or pencil marks to cells. Our simple harness allows models to directly interface with the application to make use of these tools. Using SudokuPad in-the-loop may fit well with related benchmarks that evaluate reasoning models (including vision language models) in simple game environments [19, 23]. Our evaluation in this paper (Section 4) uses text interaction (relying only on SudokuPad for the initial puzzle data extraction). We make all of these SudokuPad tools available for researchers on our repository `https://github.com/SakanaAI/Sudoku-Bench`.

### 3.4 Evaluation Framework

**Multi-step and single-shot** We evaluate models in both multi-round and single-shot configurations. In a multi-round setup, we prompt the model to analyze the board and give at least one valid digit

[2] `huggingface.co/datasets/SakanaAI/Sudoku-CTC-Reasoning`
[3] `huggingface.co/datasets/SakanaAI/Sudoku-Bench`

placement per response. We clarify that this is a committed digit(s) that cannot be undone (in the model's reasoning trace, any amount of internal backtracking is possible in order to deduce the digit). Once the digit is placed, the user displays the updated board state. We continue until the puzzle is solved or the LLM misplaces any digit. In the multi-round setting, we track both the solve rate and correct digit placements per puzzle. To keep the context window manageable, we keep the most recent 5 responses from the LLM in context, while always keeping the first user message with the puzzle specification and instructions. We report the averages as **average solve rate** and **average correct digits**. In our evaluation, we run a single evaluation per model and per puzzle, so the average is across the 100 puzzles in the set.

In the single-shot configuration, we prompt the model to provide a solution in a single response. A single-shot configuration is appropriate for evaluating models with sufficiently large context, or for a more straightforward evaluation of the smaller $4 \times 4$ puzzles. In the single-shot setting, we report only the **average solve rate**.

# 4 Baseline Performance and Analysis

We evaluated the current generation of state-of-the-art large language models on SUDOKU-BENCH, revealing substantial difficulty posed by these Sudoku variants. Table 1 summarizes model performance across puzzle sizes and interaction modes on benchmark. Even leading models such as `o3 mini high` and `Gemini 2.5 pro preview` demonstrated solve rates below 15% for the complete set. Notably, performance varied significantly by puzzle size: models generally solved smaller $4 \times 4$ puzzles at rates between 40% to 73%, but performance sharply declined for $6 \times 6$ grids and dropped nearly to zero on $9 \times 9$ puzzles, underscoring the rapid escalation in complexity.

Comparing single-shot to multi-step evaluation modes, allowing iterative feedback slightly improved outcomes for smaller puzzles but did not meaningfully impact results for larger puzzles. The minimal difference between modes suggests that the fundamental difficulty for these models lies not merely in incremental reasoning but in effectively identifying initial logical breakthroughs.

| **Model** | **Multi-step correct placements** | | | | **Multi-step solve rate (%)** | | | | **Single-shot solve rate (%)** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 4×4 | 6×6 | 9×9 | **All** | 4×4 | 6×6 | 9×9 | **All** | 4×4 | 6×6 | 9×9 | **All** |
| O3 Mini High | 9.7 | 0.7 | – | **–** | 60.0 | 0.0 | – | **–** | 73.3 | 6.7 | 2.9 | **14.0** |
| Gemini 2.5 Pro | 11.6 | 0.6 | 1.8 | **3.1** | 73.3 | 0.0 | 0.0 | **11.0** | 60.0 | 13.3 | 0.0 | **11.0** |
| Qwen 3.235B A22B | 6.5 | 1.1 | 0.7 | **1.7** | 40.0 | 0.0 | 0.0 | **6.0** | 53.3 | 0.0 | 0.0 | **8.0** |
| Qwen 3.30B A3B | 1.3 | 0.0 | 0.3 | **0.4** | 6.7 | 0.0 | 0.0 | **1.0** | 46.7 | 0.0 | 0.0 | **7.0** |
| DeepSeek R1 | 9.5 | 0.8 | 1.1 | **2.3** | 60.0 | 0.0 | 0.0 | **9.0** | 40.0 | 0.0 | 0.0 | **6.0** |
| Grok 3 Mini | 8.5 | 0.7 | 0.9 | **2.0** | 53.3 | 0.0 | 0.0 | **8.0** | 40.0 | 0.0 | 0.0 | **6.0** |
| Qwen QwQ 32B | 5.0 | 0.7 | 0.6 | **1.3** | 26.7 | 0.0 | 0.0 | **4.0** | 40.0 | 0.0 | 0.0 | **6.0** |
| Qwen 3 32B | 4.3 | 0.5 | 0.5 | **1.0** | 26.7 | 0.0 | 0.0 | **4.0** | 40.0 | 0.0 | 0.0 | **6.0** |
| Claude 3.7 Sonnet (Thinking) | 8.1 | 1.1 | – | **–** | 40.0 | 0.0 | – | **–** | 33.3 | 0.0 | 0.0 | **5.0** |
| GPT 4.1 | 2.3 | 0.2 | 0.3 | **0.6** | 13.3 | 0.0 | 0.0 | **2.0** | 13.3 | 0.0 | 0.0 | **2.0** |
| Gemini 2.0 Flash | 0.5 | 0.1 | 0.2 | **0.2** | 0.0 | 0.0 | 0.0 | **0.0** | 0.0 | 0.0 | 0.0 | **0.0** |
| Gemma 3 27B IT | 0.1 | 0.1 | 0.5 | **0.3** | 0.0 | 0.0 | 0.0 | **0.0** | 0.0 | 0.0 | 0.0 | **0.0** |
| Llama 4 Maverick | 0.2 | 0.5 | 0.4 | **0.4** | 0.0 | 0.0 | 0.0 | **0.0** | 0.0 | 0.0 | 0.0 | **0.0** |

Table 1: **Sudoku-Bench leaderboard.** Performance comparison of various LLMs on Sudoku-Bench. Percentage of puzzles completely solved for each evaluation mode (multi-step vs. single-shot), stratified by grid size. The right-most **All** columns aggregate across grid sizes (15 puzzles for 4×4 and 6×6, 70 for 9×9). In the multi-step setting, a model is prompted to provide any number of digits in its response, with the user providing an updated board state at each turn. Interaction is terminated if the model makes an incorrect placement. The average number of correct placements are presented in the first column set. In the single-shot setting the model is prompted to solve the entire puzzle in a single response. "–" indicates that fewer than the required number of responses were available due to cost limitations, so an aggregate could not be computed.

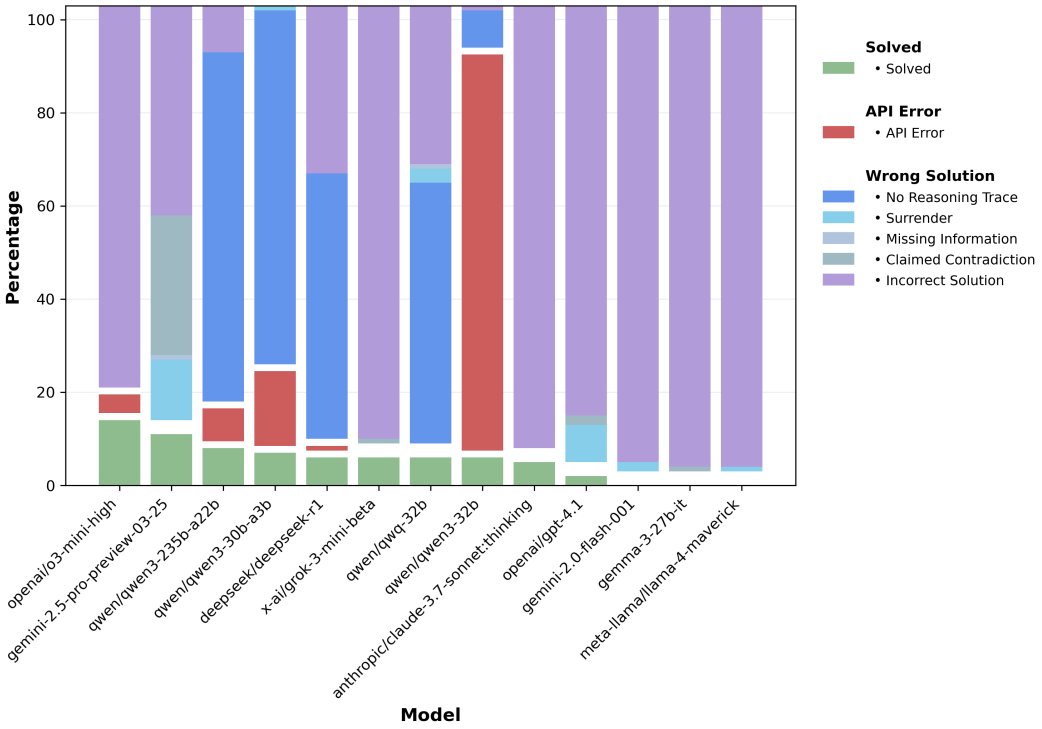



Figure 4: Response categorization for the single-shot setting.

**Categorizing model failures** Analyzing model failures indicated several recurring patterns which we categorize in Fig. 4. The most common failure mode was presenting with confidence an *Incorrect Solution*. Other failure modes included *Surrender* (model explicitly gives up), *Missing Information* (model incorrectly claims puzzle information or given constraints are incomplete), and *Claimed Contradiction* (model mistakenly identifies contradictions in the puzzle rules). Of note is *Missing Information*. Since variants are not as densely represented in the training set of foundation models compared to vanilla Sudoku, it appears the new rules and variants throw them off, most notably due to the fact that variants typically have fewer starting digits (often none) compared to the minimum of 17 in a vanilla $9 \times 9$ Sudoku. In addition, a part of model responses contain *No Reasoning Trace* so we cannot make a fine-grained categorization of its error type, otherwise we use `Claude-3.5-Haiku` to classify a wrong solution response into one of the other four error types.

## 5 Related Work

SUDOKU-BENCH complements existing benchmarks designed to evaluate advanced reasoning in artificial intelligence, with a particular focus on Sudoku variants as a structured domain for assessing creative and logical deduction.

**Benchmarks targeting creative deductive insight** Benchmarks such as the Abstraction and Reasoning Corpus (ARC; 3) present diverse tasks to test reasoning and generalization beyond pattern memorization. SUDOKU-BENCH similarly introduces novel constraints for each puzzle, resisting memorization through a continuous influx of unique puzzles. Unlike ARC, which emphasizes tasks simple for humans but challenging for AI, Sudoku variants span a broader difficulty spectrum, including puzzles challenging even for expert human solvers. Nonetheless, Sudoku puzzles offer recognizable logical breakthroughs readily appreciated by human novices, making SUDOKU-BENCH a valuable resource for precise evaluation of creative reasoning.

**Puzzle-centric reasoning datasets** Several benchmarks focus on puzzle-solving for evaluating reasoning skills [5]. For instance, PUZZLES [4] compiles canonical logic puzzles; Tyagi et al. [26]

systematically analyze grid puzzle-solving by LLMs; and ENIGMAEVAL [27] evaluates a large suite of problems from puzzle competitions. Recent additions include VGRP-BENCH [23] for visual-grid reasoning, LOGICGAME [7] for rule-based reasoning, and PUZZLEPLEX [13] for evaluating conversational agents' reasoning. BALROG [19] evaluates LLM and VLM reasoning in complex game environments and could be extended using tools from SUDOKU-BENCHto include SudokuPad as an environment.

**Sudoku as a reasoning testbed** The standard Sudoku puzzle has been extensively utilized in machine learning research. Models include Recurrent Relational Networks [20] employing message-passing, differentiable SATNet consistency layers [28], masked-denoising and diffusion methods [10, 30], and Kuramoto-inspired oscillator dynamics [15]. Further, large language models have achieved human-level accuracy through structured prompting and reasoning decomposition [12]. [24] showed a high solve rate on vanilla Sudokus by training on a sequence of steps from a solver. SUDOKU-BENCH extends this research tradition by incorporating diverse and novel puzzle constraints, enabling evaluations that specifically target multi-step, strategic, and creative reasoning.

# 6 Discussion

**The role of tool use** Evaluating model reasoning can be distinguished by whether external tools, such as constraint solvers or code execution environments, are available. Without tool use, the evaluation specifically assesses the model's intrinsic reasoning capabilities, including logical deduction, maintaining global consistency, and internally generating creative insights, akin to solving puzzles by hand. This approach emphasizes pure cognitive reasoning skills and has been the primary evaluation mode presented in our baselines (Section 4).

Conversely, allowing tool use tests the model's ability to translate a given puzzle into a formal representation suitable for external solvers, effectively interact with these tools, and interpret solver results correctly. Standard Sudoku puzzles become straightforward when a solver is employed. Variants that only employ standard constraints such as arrows, cages, etc, are also easily solved by code execution. A third category of puzzles require natural language understanding and are not straightforward to interpret as a constraint satisfaction problem. This third category is itself a meaningful test for reasoning models with tool-use enabled. However, our current intention is to assess the reasoning required to find a puzzle's "break-in," and many puzzles such as *Ascension* from Fig. 2a are easily solved by tool-use, but the solution path would be substantially different than that intended by the puzzle setter. Therefore we selected the 100 puzzles of SUDOKU-BENCH for evaluating models without tool-use. Future work could consider a separate tool-use track, potentially with a different collection of puzzles.

**Limitations** The current evaluation results are limited to text interfaced models. We would like to incorporate evaluation of VLMs in the future when they are capable of reading puzzles.

**Societal impact** The release of SUDOKU-BENCH provides a platform for assessing large language models on difficult Sudoku variant puzzles that challenge even experienced human solvers. Our evaluation results, consistent with findings from previous research, demonstrate that LLMs employ fundamentally different solving strategies compared to human approaches. As LLMs continue to advance in capability, we anticipate a future where human puzzle creators can learn from and incorporate AI-discovered strategies to develop even more intriguing and sophisticated variants, creating a synergy between human and artificial intelligence.

**Conclusion** We introduced SUDOKU-BENCH, a unified benchmark built around modern Sudoku variants that systematically stress long-horizon deduction, rule-interpretation, and strategic planning. In addition, the benchmark is uniquely suited for evaluating creative reasoning via the rich and varied collection of break-ins featured in most puzzles. The benchmark includes a curated puzzle corpora with textual representations, providing a controlled substrate for measuring how well language models cope with novel, tightly coupled constraints. Baseline experiments show that frontier LLMs solve fewer than 15% of instances without external tools, and performance falls sharply on $9 \times 9$ variants—evidence that substantial headroom remains for improvements.

# References


[1] Logic masters germany. https://logic-masters.de. Accessed: 2025-05-13.

[2] S. Bubeck, V. Chandrasekaran, R. Eldan, J. A. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y.-F. Li, S. M. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, and Y. Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4. *ArXiv*, abs/2303.12712, 2023. URL https://api.semanticscholar.org/CorpusID:257663729.

[3] F. Chollet. On the measure of intelligence, 2019. URL https://arxiv.org/abs/1911.01547.

[4] B. Estermann, L. A. Lanzendörfer, Y. Niedermayr, and R. Wattenhofer. Puzzles: A benchmark for neural algorithmic reasoning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 127059–127098. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e5d1eaadeed651ba1021c09149db4b92-Paper-Datasets_and_Benchmarks_Track.pdf.

[5] P. Giadikiaroglou, M. Lymperaiou, G. Filandrianos, and G. Stamou. Puzzle solving using reasoning of large language models: A survey.

[6] E. Glazer, E. Erdil, T. Besiroglu, D. Chicharro, E. Chen, A. Gunning, C. F. Olsson, J.-S. Denain, A. Ho, E. de Oliveira Santos, O. Järviniemi, M. Barnett, R. Sandler, M. Vrzala, J. Sevilla, Q. Ren, E. Pratt, L. Levine, G. Barkley, N. Stewart, B. Grechuk, T. Grechuk, S. V. Enugandla, and M. Wildon. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai, 2024. URL https://arxiv.org/abs/2411.04872.

[7] J. Gui, Y. Liu, J. Cheng, X. Gu, X. Liu, H. Wang, Y. Dong, J. Tang, and M. Huang. LogicGame: Benchmarking rule-based reasoning abilities of large language models.

[8] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

[9] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, G. Dulac-Arnold, J. Agapiou, J. Z. Leibo, and A. Gruslys. Deep q-learning from demonstrations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

[10] J. Kim, K. Shah, V. Kontonis, S. Kakade, and S. Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions.

[11] D. Li, S. Cao, T. Griggs, S. Liu, X. Mo, E. Tang, S. Hegde, K. Hakhamaneshi, S. G. Patil, M. Zaharia, et al. Llms can easily learn to reason from demonstrations structure, not content, is what matters! *arXiv preprint arXiv:2502.07374*, 2025.

[12] J. Long. Large language model guided tree-of-thought.

[13] Y. Long, T. Jiang, Y. Zhao, A. Cohan, and D. Shasha. PuzzlePlex: A benchmark to evaluate the reasoning and planning of large language models on puzzles.

[14] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6382–6393, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[15] T. Miyato, S. Löwe, A. Geiger, and M. Welling. Artificial kuramoto oscillatory neurons.

[16] N. Muennighoff, Z. Yang, W. Shi, X. L. Li, L. Fei-Fei, H. Hajishirzi, L. Zettlemoyer, P. Liang, E. Candès, and T. Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

[17] S. Neumann. Sudokupad, 2021. URL https://sudokupad.app/.

[18] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[19] D. Paglieri, B. Cupiał, S. Coward, U. Piterbarg, M. Wolczyk, A. Khan, E. Pignatelli, Kuciński, L. Pinto, R. Fergus, J. N. Foerster, J. Parker-Holder, and T. Rocktäschel. BALROG: Benchmarking agentic LLM and VLM reasoning on games.

[20] R. B. Palm, U. Paquet, and O. Winther. Recurrent relational networks.

[21] R. Pelánek. Difficulty rating of sudoku puzzles by a computational model. In *The Florida AI Research Society*, 2011. URL https://api.semanticscholar.org/CorpusID:6431985.

[22] L. Phan, A. Gatti, Z. Han, N. Li, J. Hu, H. Zhang, C. B. C. Zhang, M. Shaaban, J. Ling, S. Shi, et al. Humanity's last exam, 2025. URL https://arxiv.org/abs/2501.14249.

[23] Y. Ren, K. Tertikas, S. Maiti, J. Han, T. Zhang, S. Süsstrunk, and F. Kokkinos. VGRP-bench: Visual grid reasoning puzzle benchmark for large vision-language models.

[24] K. Shah, N. Dikkala, X. Wang, and R. Panigrahy. Causal language modeling can elicit search and reasoning capabilities on logic puzzles.

[25] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587): 484–489, Jan. 2016. doi: 10.1038/nature16961.

[26] N. Tyagi, M. Parmar, M. Kulkarni, A. Rrv, N. Patel, M. Nakamura, A. Mitra, and C. Baral. Step-by-step reasoning to solve grid puzzles: Where do LLMs falter? In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19898–19915, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1111. URL https://aclanthology.org/2024.emnlp-main.1111/.

[27] C. J. Wang, D. Lee, C. Menghini, J. Mols, J. Doughty, A. Khoja, J. Lynch, S. Hendryx, S. Yue, and D. Hendrycks. Enigmaeval: A benchmark of long multimodal reasoning challenges, 2025. URL https://arxiv.org/abs/2502.08859.

[28] P.-W. Wang, P. L. Donti, B. Wilder, and Z. Kolter. SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver.

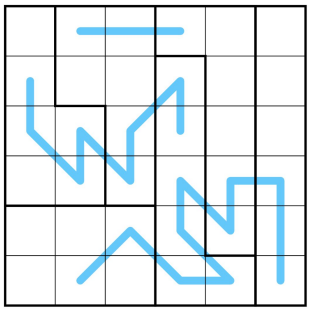
[29] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022. URL https://arxiv.org/abs/2201.11903.

[30] J. Ye, J. Gao, S. Gong, L. Zheng, X. Jiang, Z. Li, and L. Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning.

[31] H. Zhang, J. Da, D. Lee, V. Robinson, C. Wu, W. Song, T. Zhao, P. Raja, C. Zhuang, D. Slack, et al. A careful examination of large language model performance on grade school arithmetic. *Advances in Neural Information Processing Systems*, 37:46819–46836, 2024.

# A Appendix

We show a $6 \times 6$ puzzle (*Sumthings*) sovled by `Gemini 2.5 Pro Preview` in Fig 5. The model adopted a similar strategy as in the *Ascension* puzzle. In this puzzle, the model successfully reduces the search space to a reasonably small size and uses search to find the correct solution. However, as demonstrated by the previous example of *Ascension*, such strategy becomes ineffective when the puzzle complexity increases and one has to rely on "break-in" techniques to effectively reduce the search space.



**Sumthings**
*by Kennet's Dad*

Place the digits 1-6, once each, in every row, column, and region. Region borders divide the blue line into segments with the same sum.

(a) Description of the puzzle *Sumthings*.

reasoning trace by `gemini-2.5-pro-preview-03-25`

**1. Understand the Rules and Setup**
* [...]
**2. Analyze the Blue Lines and Segments**
* [...]
* **Line 2:** r3c4, r2c4, r3c3, r4c3, r3c2, r4c2, r3c1, r2c1
  * Segments: {r3c4, r2c4} (R4), {r3c3, r4c3} (R2), {r3c2, r4c2, r3c1, r2c1} (R1).
  * Constraint: Sums are equal. Let this sum be S2.
  * `S2 = r3c4 + r2c4`
  * `S2 = r3c3 + r4c3`
  * `S2 = r3c2 + r4c2 + r3c1 + r2c1`
  * The minimum sum for 4 distinct digits is 1+2+3+4=10. The maximum sum for 2 distinct digits is 5+6=11. Therefore, **S2 must be 10 or 11**.
* [...]
  * The minimum sum for 4 distinct digits is 10. The maximum sum for 2 distinct digits is 11. Therefore, **S3 must be 10 or 11**.

**3. Determine the Line Sums (S2 and S3)**
* [...]
* Let A = digits in {r2c4, r3c4}. Let B = digits in {r4c4, r5c4, r6c4, r6c5}. A and B are disjoint sets of digits.
  * If S2=10, A={4, 6}. If S2=11, A={5, 6}.
  * If S3=10, B={1, 2, 3, 4}. If S3=11, B={1, 2, 3, 5}.
* Test combinations:
  * S2=10, S3=10: A={4,6}, B={1,2,3,4}. Union = {1,2,3,4,6}. Missing 5. Impossible.
  * S2=10, S3=11: A={4,6}, B={1,2,3,5}. Union = {1,2,3,4,5,6}. Possible.
  * S2=11, S3=10: A={5,6}, B={1,2,3,4}. Union = {1,2,3,4,5,6}. Possible.
  * S2=11, S3=11: A={5,6}, B={1,2,3,5}. Union = {1,2,3,5,6}. Missing 4. Impossible.
* So, **either (S2=10, S3=11) or (S2=11, S3=10)**.

**4. Use the r1c4 = r6c5 Relationship**
[...]
**5. Test the Two Possible Scenarios for S2 and S3**
* **Scenario 1: S2=10, S3=11**
[...]
* **Scenario 2: S2=11, S3=10**
[...]
[...]

**Analysis**
Understanding the rules.

**Analysis**
The model effectively reduces the possible cases of two blue lines' segment sum (S2 and S3) to two candidates (10 or 11).

**Analysis**
From the 2 x 2 = 4 combinations of S2 and S3, it further reduces it to **2 possible cases**, which forms a small constrained space for latter search.

**Search**
The model starts to search by starting from **(S2=10, S3=11) or (S2=11, S3=10)** and guessing other numbers on the lines until a contradiction.

Due to the small search space, the model can find the only correct placements of the lines and reach a correct global solution in the end.

(b) Gemini 2.5 Pro's solution to the puzzle *Sumthings*.

Figure 5: *Sumthings* example.

# NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state exactly what we contribute (benchmark, dataset, tools, baselines) and do not extend claims beyond those results (Secs. 1, 6).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations of the work in Section 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper provides open access to the benchmark dataset on HuggingFace and the code for evaluation on GitHub, as detailed in Sec. 3. The README files in these repositories contain instructions to reproduce the baseline experimental results presented in Sec. 4.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: All code can be accessed at `https://github.com/SakanaAI/Sudoku-Bench`. The dataset can be accessed at `https://huggingface.co/datasets/SakanaAI/Sudoku-Bench`. Both URLs contain sufficient information in the READMEs.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: The paper does not involve any training experiments. Details of test experiments are provided in the open-source code at `https://github.com/SakanaAI/Sudoku-Bench`.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: Due to the cost of LLM APIs, we report the evaluation results of a single run for each model.

Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: Experiments in the paper are conducted using LLM APIs.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. The work focuses on creating a benchmark for evaluating AI reasoning using Sudoku puzzles, promotes open science through publicly available datasets and tools, and does not involve human subjects in a way that would raise ethical concerns beyond those addressed by data/asset licensing.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss the societal impact of the work in Section 6.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: The released assets are Sudoku puzzles, textual descriptions, expert reasoning traces, and tools to interact with them. These are not considered to have a high risk for misuse in the way pretrained language models or image generators might. The data is intended for research purposes to advance AI reasoning capabilities.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: For this project we formed a legal partnership with the hosts of *Cracking the Cryptic* to provide content from the channel freely available for research use under an MIT license, with the intent of supporting open source science and foster collaboration with the research community. The majority of the puzzles we include in our benchmarks are featured on the channel. Additionally, many Sudoku variant puzzles are posted to the *Cracking the Cryptic* discord server which states that the assets are fair use. We additionally met with the creator of the SudokuPad app for permission to provide tooling for the app

and provide our interfacing tools to both the research and puzzle communities. We provide acknowledgments to puzzle creators featured in our repository, `https://github.com/SakanaAI/Sudoku-Bench`. We were unable to reach out to each individual puzzle setter given the large number of setters for this domain. The dedicated vanilla Sudoku dataset, from Nikoli, is discussed in Section 3.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [NA]

Justification: This paper does not include any puzzles that are not already publicly available.

Guidelines:

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

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

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

Answer: [NA]

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

Guidelines:

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

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

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

Answer: [NA]

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

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.