Baba is LLM: Reasoning in a Game with Dynamic Rules

Abstract. Large language models (LLMs) are known to perform well on language tasks, but struggle with reasoning tasks. This paper explores the ability of LLMs to play the 2D puzzle game *Baba is You*, in which players manipulate rules by rearranging text blocks that define object properties. Given that this rule-manipulation relies on language abilities *and* reasoning, it is a compelling challenge for LLMs. Six LLMs are evaluated using different prompt types, including (1) simple, (2) rule-extended and (3) action-extended prompts. In addition, two models (Mistral, OLMo) are finetuned using textual and structural data from the game. Results show that while larger models (particularly GPT-4o) perform better in reasoning and puzzle solving, smaller unadapted models struggle to recognize game mechanics or apply rule changes. Finetuning improves the ability to analyze the game levels, but does not significantly improve solution formulation. We conclude that even for state-of-the-art and finetuned LLMs, reasoning about dynamic rule changes is difficult (specifically, understanding the use-mention distinction). The results provide insights into the applicability of LLMs to complex problem-solving tasks and highlight the suitability of games with dynamically changing rules for testing reasoning and reflection by LLMs.

Keywords: Large language models · reasoning · dynamic rule changes · games.

1 Introduction

Artificial Intelligence (AI) has a long history in using games as benchmarks for reasoning, decision-making, and problem-solving capabilities [Campbell et al., 2002, Silver et al., 2016, 2018, Brown and Sandholm, 2018, Berner et al., 2019, Schrittwieser et al., 2020]. This paper investigates the use of large language models (LLMs) in the 2D puzzle game *Baba is You* [Teikari, 2019]. In this game players must alter rules by manipulating text blocks. Solving puzzles in this environment requires understanding how rule changes affect the game state and to apply that understanding dynamically, implying a form of reasoning in which the model should be able to reflect on the effects of its own actions.

LLMs, based on the transformer architecture [Vaswani et al., 2023] have demonstrated strong performance in natural language processing tasks including text generation, machine translation, conversational agents and code generation [Naveed et al., 2024]. Techniques such as finetuning [Xu et al., 2023], reinforcement learning with human feedback (RLHF) [Chaudhari et al., 2024], and prompt-based learning [Kamath et al., 2024] have been developed to improve the performance of the transformer architecture. Beyond natural language

processing, LLMs are also emerging as agents in games [Bakhtin et al., 2022, Topsakal et al., 2024, Marincioni et al., 2024, Müller-Brockhausen et al., 2023].

Baba is You is compelling because of its dynamic rule system relying on two-level language-based mechanics, which can be understood in terms of the classical *mention* versus *use* distinction [Wilson, 2017, Saka, 1998]. Board games in general are based on pushing around pieces associated with a fixed meaning (mention). However, in *Baba is You* certain pieces can form a new game rule when they are aligned (use). Unlike games with fixed rules, *Baba is You* allows players to rewrite the logic of the game by manipulating the pieces. While LLM's strong language and general pattern-learning abilities suggest potential [Mirchandani et al., 2023], an initial study by Cloos et al. [2024] showed indeed that state-of-the-art LLMs struggle with the reasoning aspects of *Baba is You*, failing to generalize rule manipulation.

This paper evaluates how well LLMs solve *Baba is You* puzzles. We use two approaches: prompt-based learning and finetuning. We test six LLMs (GPT-4o, Gemini-Flash 1.5, OLMo 2 13B and 7B, Mistral 7B and Mixtral 8x7B) across three prompt types. We additionally finetune Mistral 7B and OLMo 7B using game data. Our contributions are as follows:

- Comparing different prompts in six LLMs, we find that prompt-based learning achieves weak results when dynamic rule changes are necessary—even for LLMs with enhanced reasoning capacities;
- Using a dataset for finetuning, we find that finetuning on two open LLMs is able to improve performance somewhat;
- Reasoning about dynamic rules changes, remains a challenging problem for current Reasoning LLMs; the deceptively simple puzzle game of *Baba is You* offers a challenging testbed for Reasoning LLMs.

All training scripts, prompts, and finetuning datasets of this work are publicly available [Anonymous, 2025a,c,b].

2 Related work

With the advent of LLMs, a new type of learning has emerged: prompt-based (or in-context) learning [Kamath et al., 2024]. This type of learning occurs at inference time, using a structured prompt that includes a task description, optional examples, and a query. To enhance LLM reasoning, chain of thought (CoT) prompting was introduced, where the model is guided to generated intermediate steps before answering [Wei et al., 2022]. In few-shot CoT prompts include a task or question, followed by a step-by-step reasoning example along with the final answer, and ending with a similar question or task. This approach showed better performance on complex reasoning tasks for large models. Kojima et al. [2022] proposed a zero-shot CoT template for reasoning, they unlock the reasoning step by adding the *Let's think step by step* sentence at the end of each prompt.

Wang et al. [2023a] introduce plan and solve prompting (PS), another zero-shot CoT method. It prompts the model to first plan a solution and then execute

it, using the sentences *Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan and solve the problem step by step.* They extend these sentences with more detailed instructions to reduce errors in the reasoning step. Further approaches on reasoning in LLMs can be found in [Chu et al., 2024, Dong et al., 2023, Huang and Chang, 2023, Plaat et al., 2024]. Another approach to teaching a pretrained LLM to perform a new task is finetuning, where the parameters of the model are adjusted [Jeong, 2024]. Finetuning adapts pretrained LLMs to specific task with smaller, domain-specific datasets. A common method is supervised finetuning (SFT), where labeled examples guide learning. Instruction-tuning, a variant of SFT, trains models on *(Instruction, Output)* pairs, where *Instruction* is a human instruction and *Output* is the desired response by the LLM for that instruction [Zhang et al., 2024]. Full finetuning updates all model parameters. This technique can be costly, as the pretrained model often contains billions of parameters, see early models such as GPT [Radford et al., 2018]. Parameter-efficient finetuning (PEFT) offers a lighter alternative by modifying only a small subset of parameters [Han et al., 2024]. A notable PEFT method is LoRA [Hu et al., 2021], which inserts low-rank matrices to approximate weight updates, reducing memory and compute costs.

The rise of AI agents achieving dominance in gaming begins in the 1990s with Deep Blue [Campbell et al., 2002], which defeated world chess champion Garry kasparov using brute-force search and domain expertise. Attention then shifted towards machine and reinforcement learning approaches [Silver et al., 2016, Plaat, 2020]. Currently, agents and LLMs are converging [Plaat et al., 2025]. In the field of AI agents playing games, ChessGPT [Feng et al., 2024] introduced a substantial game and language dataset for chess, upon which two models have been created. Li et al. [2024] evaluated LLMs in Minesweeper using different input formats, finding that GPT-4 outperformed GPT-3.5, although limitations remained. Noever and Burdick [2021] used GPT-2 to solve puzzles such as mazes, Sudoku and the Rubik's Cube, by training on solved examples, demonstrating a text-based alternative to traditional search methods. These studies highlight the potential of LLMs in solving puzzles and playing games.

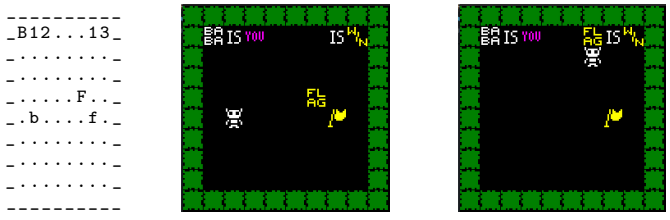


Fig. 1: ASCII representation of a level; Pictorial representation; By pushing (*mention*), Baba has created the Rule (*use*) FLAG IS WIN

3 Method

The game *Baba is You* [Teikari, 2019] is a 2D puzzle game with levels: a grid filled with objects and text blocks (see Figure 1). Text blocks can be used to create rules; these rules can be created from left to right or from top to bottom. The rule is active if there is at least one object, one verb, and one object or property aligned in a valid syntax (Figure 1, right-most panel shows activation of the rule *FLAG IS WIN*). The primary components of the game are:

- Objects: entities in the game, such as BABA, WALL, or ROCK.
- Verbs: These include IS and HAS, which create the backbone of syntactic rule construction.
- Properties: Attributes that determine how the object interacts within the environment, such as PUSH, STOP, WIN, or YOU.

Every solvable level should have a win condition and an object that can be controlled by the player. Rules dynamically define how objects behave in this game; an object does not really matter until there is a rule assigned to the object. Rules can be created, modified, or broken during gameplay by rearranging of text blocks. A level is considered complete when the object that is controlled by the player (IS YOU) touches the object designated as the win condition (IS WIN), or when the same object satisfies both rules. Figure 2 illustrates gameplay scenarios. The most common properties and rules are explained in Table 3 (Appendix).

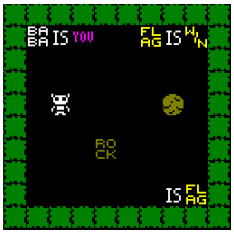
In this work we used a simplified version of the game: the jam version of *Baba is You*[1] and a section of the game mechanics of the "Game Module" from Baba is Y'all [Charity et al., 2020]. Levels are encoded as strings of characters, where each character corresponds to an object or text block, see the ASCII grid representation in the left panel of Figure 1. In the simplified version of the game, the player is able to perform four distinct actions:

- Move: Navigate the controlled object towards other objects or text blocks;
- Create a Rule: Push text blocks into a valid rule by aligning them with a controlled object;
- Break a Rule: Break an active rule by pushing a text block away from its syntactic alignment;
- Push: Interact with text blocks, or objects if they are set to PUSH.

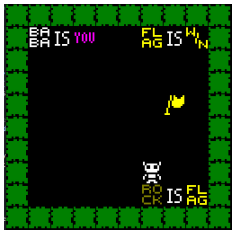
3.1 Experimental Setup

In this section we describe the methodologies used in our study, including data collection, experimental setup, and analysis techniques. We evaluated six different LLMs. *GPT-4o* [Hurst et al., 2024] (OpenAI) is a state-of-the-art reasoning model due to its strong performance on various tasks. GPT-4o can reason at inference time [Valmeekam et al., 2024], using methods such as reinforcement learning to call a model multiple times with different prompts. *Gemini 1.5-Flash* [Gemini, 2024] (Google DeepMind) is designed for cost efficiency and fast

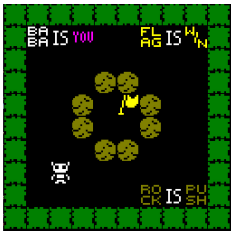
[1] https://hempuli.itch.io/baba-is-you



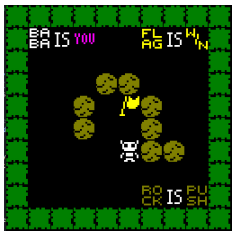
(a) Level 4: A flag object is the win condition, but there is currently no flag object to reach



(b) The rule ROCK IS FLAG is created and the rock object is transformed into a flag object

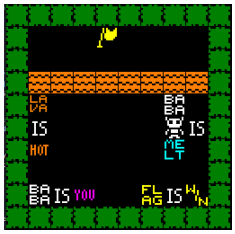


(c) Level 6: The flag object, which serves as the win condition, is surrounded by pushable rock objects



(d) Two rock objects are pushed away by the baba object to clear the path to the flag object

(e) Level 12: The path towards the flag object (win condition) is blocked by lava because it is hot and baba object is set to melt



(f) The rule BABA IS MELT is broken, this means baba will not melt when touching the lava, and the path toward the flag object is cleared

Fig. 2: Examples of different game mechanics using the flag as a win condition. Each sequence shows how obstacles are manipulated or rules are changed to create a path to the flag.

inference. *Mistral 7B instruct* [Jiang et al., 2023] (Mistral AI) is a 7B parameter model tuned for instruction-following tasks. *Mixtral 8x7B* [Jiang et al., 2024] uses a Mixture-of-Experts (MoE) design, activating only 12B of its 45B parameters per inference to reduce computational cost. *OLMo 7B and 13B instruct* [Jiang et al., 2023] (AI2) are open source models trained on the Dolma dataset, an open dataset that includes a mix of web content, academic publications, code, books, and encyclopedic materials. The OLMo models are designed with a focus on research accessibility, interpretability, and transparency.

To enable the model to understand and play the game *Baba is You*, we constructed three different prompts. (Please refer to the Appendix.)

- **Simple Prompt** The prompt consists of the game mechanics, the definition of the characters to interpret the level, and the definition of each property.
- **Rule-extended Prompt** Adds the rules that are active at the current level to the prompt.
- **Action-extended Prompt** Further expands the prompt by including a description of the possible actions, partially adapted and extended from Cloos et al. [2024].

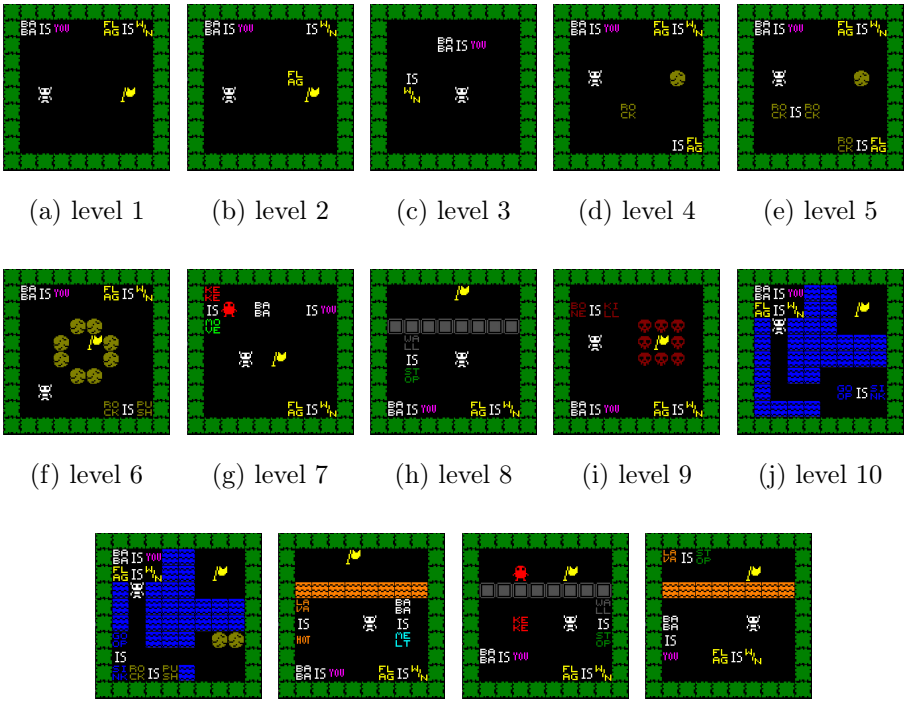
Each prompt ends with a question to solve the given grid level, followed by the ASCII grid level, and, at the end, a PS sentence (Plan-and-Solve [Wang et al., 2023b]) to activate CoT (Chain-of-Thought [Wei et al., 2022]). The prompts were constructed manually through iterative trial and error with GPT-4o. Outputs were reviewed for improvement, refined, and resubmitted until a satisfactory version was achieved.

Evaluation of Reasoning In order to investigate how well LLMs perform in reasoning and solving *Baba is You* levels, a manual analysis was performed to examine the reasoning chains generated by LLMs. The reasoning chain can be divided into four distinct sections: the interpretation of the level, the formulation of the problem statement, the formulation of the solution for the problem, and the formulation of the actions that should be taken for the solution. The first two sections are part of the analysis of the level, while the latter two sections are part of the solution process so formulating an answer consists of four steps. Table 1 summarizes errors encountered in these steps. The error categories are used in Figure 4. If a step is correct, it is marked with a *c* label together with the number of the step, otherwise errors are classified according to the subcategories. The correctness frequency is shown in Figure 5.

To evaluate the LLMs, each model and prompt format was tested in 14 different levels, each of which tests a different aspect of the game, see Figure 3. Most of these levels are demo levels of the Keke AI competition [Charity and Togelius, 2022], except level 14. These levels require some logical thinking, but are relatively easy for humans, due to our natural ability to reason. However, LLMs encounter a challenge when confronted with the task of solving these levels. These models must not only interpret the rules and mechanics of the game from the text, but also apply them in the environment. Unlike humans, LLMs

Category	Sub-Category	Description
Level Interpretation (w1)	1 Hallucination	When information is completely out of context and not present in the levels
	2 Incorrect definitions	Incorrect classification of an object or text block or incorrect definition of a rule
	3 Incomplete Information	Absence of defining objects, text blocks, or rules that are present in the level
Formulate Problem Stmt (w2)	1 Transfer of errors	Wrong problem statement derived from a previous incorrect statements
	2 Incomplete information	Missing elements in the problem statement
	3 Wrong assumptions	Assuming a condition or rule applies without explicit evidence
	4 Hallucination	When information is completely out of context and not present in the levels
Formulating Solutions (w3)	1 Transfer of errors	Wrong solution derived from a previous incorrect statements
	2 Hallucination	When information is completely out of context and not present in the levels
	3 Wrong reasoning	Drawing incorrect conclusions due to misinterpreting rules, neglecting constraints, or failing to account for rule interactions and ambiguities
	4 Incomplete solution	Missing steps to fully solve the level
Formulating Actions (w4)	1 Transfer of errors	Wrong actions derived from a previous incorrect statements
	2 Incomplete actions	Missing actions to complete the level
	3 Wrong format	The actions are in the wrong format (only prompt 3)
	4 Wrong actions	Actions proposed are not solving the level
	5 Hallucination	When actions are completely out of context

Table 1: Error Categorization



(a) level 1 (b) level 2 (c) level 3 (d) level 4 (e) level 5

(f) level 6 (g) level 7 (h) level 8 (i) level 9 (j) level 10

(k) level 11 (l) level 12 (m) level 13 (n) level 14

Fig. 3: Levels used for the evaluation of the LLM models in playing *Baba is You*

have no inherent understanding of the world. These models rely entirely on the information provided to decide how to interact with and manipulate the game state. The levels are designed to assess specific components of the game, including rule creation, transformation, immutability and logical reasoning, which are needed to determine the model's ability to play the game *Baba is You*. Mistral and OLMo consistently produce identical outputs for repeated runs of the same prompt. In contrast, GPT-4o and Gemini Flash 1.5 exhibit variability. The accuracy was evaluated by running each prompt five times, a solution was correct if it appeared in at least three runs.

For finetuning, we combined three different datasets. Each dataset contains a specific type of data. The largest dataset [Bjorklund, 2025] consists of various questions designed to improve the model's reasoning ability. These questions cover a range of logical and analytical challenges. The second dataset [Anonymous, 2025a] contains questions specifically related to the game mechanics of *Baba is You*. It includes questions about the interactions between different game elements, the effects of specific rule changes, and the general logic of the game. The third dataset [Anonymous, 2025a] is the smallest and consists of different levels of *Baba is You*. In this dataset, the input corresponds action-extended

Dataset	Size
CoT-logic-reasoning	10500
Questions game mechanics	289
Levels & answers	15

Table 2: The size of the three datasets used for finetuning

prompt of the level description, while the output represents the expected solution that the model should generate. Together, these three data sets form a combined data set used for finetuning, see Table 2 for the sizes.

The dataset containing questions about the game mechanics of *Baba is You* was created through the following process. Initially, a set of questions was crafted, focusing on the rules and mechanics of the game. Then, GPT-4o was prompted to generate additional unique questions based on the ones we had already created. These generated questions and answers were reviewed and, when necessary, corrected. This iterative process allowed us to quickly build a solid dataset of questions related to the game's mechanics. The other dataset, which consists of levels and their solutions, is entirely handwritten. As a result, this dataset is smaller, as more time was spent on creating detailed solutions for each level rather than on increasing the dataset size. We trained Mistral 7B and OLMo 7B on the combined dataset using LoRA for parameter-efficient finetuning.

4 Results

In this section, we present the findings of our study, analyzing the outcomes based on the predefined metrics. We start with the prompt-based learning results.

4.1 Prompt-based learning

Simple prompt (1) The main challenge across models was to identify the active rules. GPT-4o performed better due to its improved recognition of objects and text blocks in the grid (Figure 4, prompt 2), though it still struggled with vertically placed rules. OLMo and Mistral had difficulty recognizing objects and text, which prevented them from formulating correct rules. As a result, most outputs for the simple prompt were incorrect (results are not shown).

Rule-extended (2) and Action-extended (3) GPT-4o performs relatively well, demonstrating a strong ability to understand the grid and formulate correct problem statements (Figure 4). This LLM also provides reasonable level solutions (Figure 5), making it potentially useful for assisting players. However, its action formulation is less reliable and often does not align with its own solutions. A key weakness lies in distinguishing which rules are breakable. In levels 13 and 14, GPT-4o incorrectly suggests breaking unbreakable rules to reach the flag, highlighting a difficulty in understanding spatial constraints. The action-extended

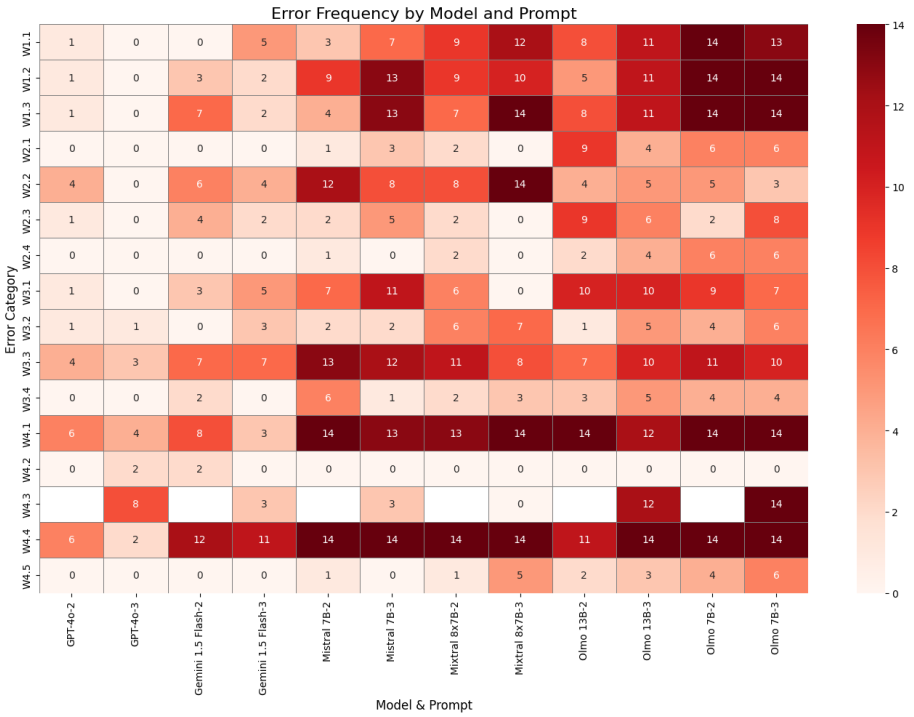


Fig. 4: Frequency of error step and subcategory (see Table 1) in the reasoning chains generated by the models. GPT-4o has the least errors. Furthermore, OLMo 7B generates the most errors in the reasoning output for the levels, primarily when defining the text blocks and objects in the grid. Finally, all models encounter difficulties in formulating actions on the grid itself.

prompt led to small improvements, with GPT-4o generating more accurate solutions and demonstrating better grasp of game mechanics. Still, it did not always strictly adhere to the action format. Gemini Flash 1.5 performs slightly worse than GPT-4o in identifying objects and text in the grid, often leading to incomplete or occasionally missing problem statements. Although these issues were not a major obstacle for solution generation, the model struggled to consistently describe the obstacles. Also, with the third prompt, hallucinations increased (those in the first step did not transfer to other steps). There was also more hallucination in formulating the solution. Interestingly, at level 12 using prompt 2, the LLM proposed breaking 'BABA IS MELT' and forming 'LAVA IS MELT', causing lava to disappear, followed by creating 'BABA IS WIN', which results in a successful outcome. This solution is intriguing because it is not the most straightforward approach, but a creative way to solve the puzzle.

Like GPT-4o, Gemini Flash 1.5 struggles with generating accurate actions. Notably, it rarely suggested rule-breaking with the rule-extended prompt but did

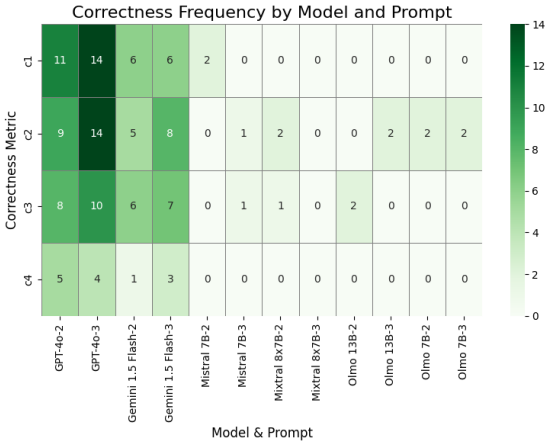


Fig. 5: Correctness frequency per step in the reasoning chain generated by the LLM models (see Section 3.1). GPT-4o has generated the most correct steps in the reasoning chain for the levels. Furthermore, GPT-4o and Gemini Flash 1.5 benefit from the action-extended prompt, while the smaller models encounter difficulties irrespective of prompt structuring.

so more often with the action-extended prompt. Overall, both models performed better with the action-extended prompt, showing fewer errors, and improved step-by-step reasoning (see Figures 6 and 7). In Table 4 some examples of error snippets of the reasoning chain of GPT-4o and Gemini Flash 1.5 are shown (Appendix). We also evaluated OLMo and Mistral models on 14 *Baba is You* levels. Overall, they performed significantly worse than GPT-4o and Gemini Flash 1.5, particularly in object and text block identification in the grid (Figure 4). Table 5 shows examples of error snippets of the reasoning chain of these models. OLMo-7B and 13B struggled with hallucinations and frequent misidentification of grid elements. They often failed to distinguish between objects and rules, which led to incoherent problem statements and solutions. Both models frequently misinterpreted "WIN" as the target object and showed little understanding of the rule mechanics or grid constraints.

Mistral 7B and Mixtral 8x7B performed slightly better in object recognition but continued to produce flawed solutions. Mixtral often skipped the problem statement entirely and jumped straight to solutions, omitting context and causing logical gaps. A recurring issue was the models' misunderstanding of the rule-breaking mechanism. Rather than removing rules, they often suggested alternative rules, missing the mechanic's intent. Additionally, both Mistral models struggled to track active rules, sometimes suggesting to create rules that already existed or were impossible to form given the available text blocks. One common error was assuming that movement required 'BABA IS MOVE,' indicating a lack of grasp of default game behavior.

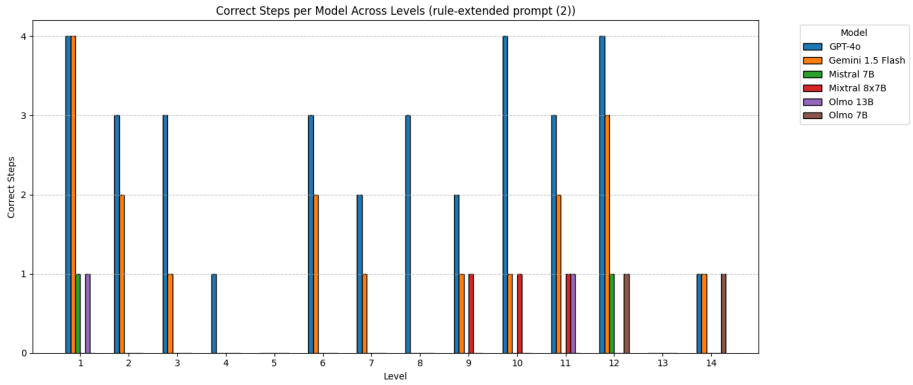


Fig. 6: Correct steps per model across the 14 *Baba is You* levels with the rule-extended prompt. The rule-extended prompt, which provides the active rules present in the level, improves performance across models but still highlights major differences in reasoning capabilities. GPT-4o outperforms other models, demonstrating stronger multi-step problem-solving skills. While Gemini 1.5 Flash show partial success, its performance remains inconsistent. The results suggest that simply providing active rules helps, but does not bridge, the gap in logical reasoning ability between smaller models and more advanced LLMs like GPT-4o.

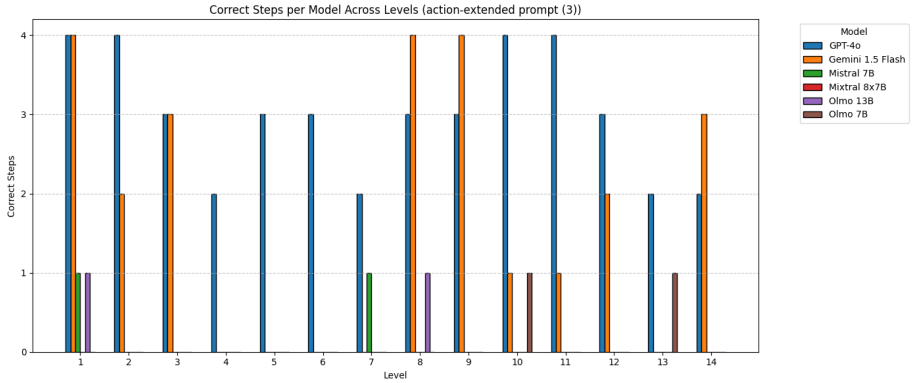


Fig. 7: Correct steps per model across the 14 *Baba is You* levels with the action-extended prompt. The action-extended prompt, which provides additional details about possible actions, leads to notable improvements for some models, particularly Gemini 1.5 Flash and GPT-4o. However, GPT-4o remains the strongest performer, consistently solving more steps across all levels. While some smaller models show slight improvements, their overall performance remains limited, suggesting that improving prompts alone is not sufficient to overcome their reasoning limitations. These results highlight the importance of both prompt design and underlying model capability in tackling complex rule-based reasoning tasks.

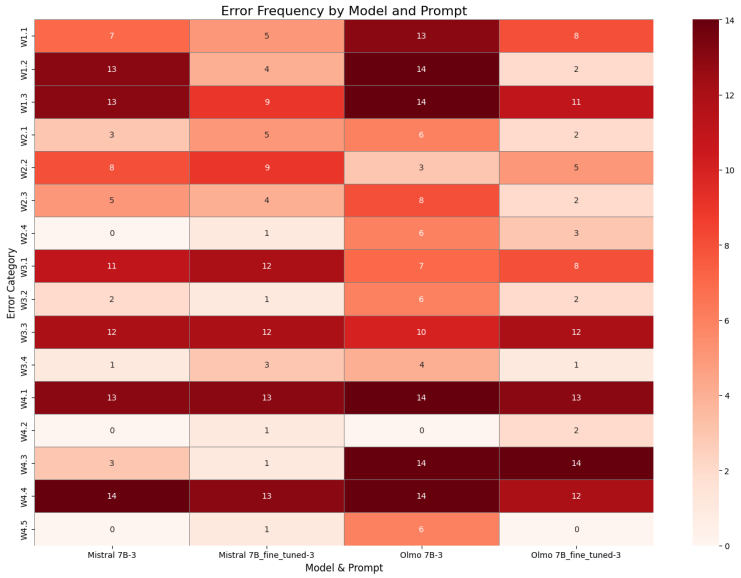


Fig. 8: Frequency of errors per step and subcategory in the reasoning chains generated by the models when solving the *Baba is You* levels. We see that both finetuned models have fewer errors in the analyzing part of the level in the reasoning chain after finetuning compared to the original model.

Unlike GPT-4o and Gemini 1.5 Flash, which showed improvements with structured prompts, OLMo and Mistral models did not consistently benefit from action-extended prompts. The solutions remained equally flawed. These results highlight key limitations of smaller models: difficulty distinguishing game entities, tracking rule states, and reasoning through rule-breaking mechanics. While Mistral models showed slight improvement over OLMo, neither models demonstrated strong puzzle-solving ability. In levels 4 and 5, most models misinterpreted the presence of the rule "FLAG IS WIN" as implying the flag's existence, overlooking the need to create or transform the flag. In level 5, some models incorrectly assumed the flag was inside the rock due to "FLAG IS ROCK," revealing confusion between rule-based transformation and object persistence.

4.2 Finetuning

Next, we discuss the finetuning results on Mistral and OLMo. Finetuning Mistral 7B improved classification of the objects and text blocks in the grid (Figure 9). There were fewer misclassifications and incomplete information problems (Figure 8). However, this did not improve problem statements or solving of levels, which was often incomplete with wrong assumptions about grid-objects.

After finetuning, OLMo 7B improved the formulation of the problem statement (Figure 9) and achieved a reduction in classification errors for objects and

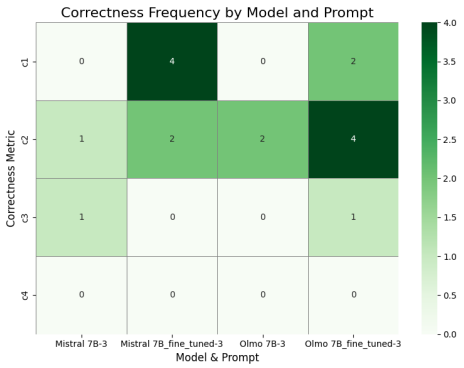


Fig. 9: Correctness frequency per step in the reasoning chain generated by the LLM models. Both finetuned models have more correct steps after finetuning. For Mistral 7B there is an improvement in classification of objects and text blocks in the grid. For OLMo 7B there is an improvement in the problem statement formulation.

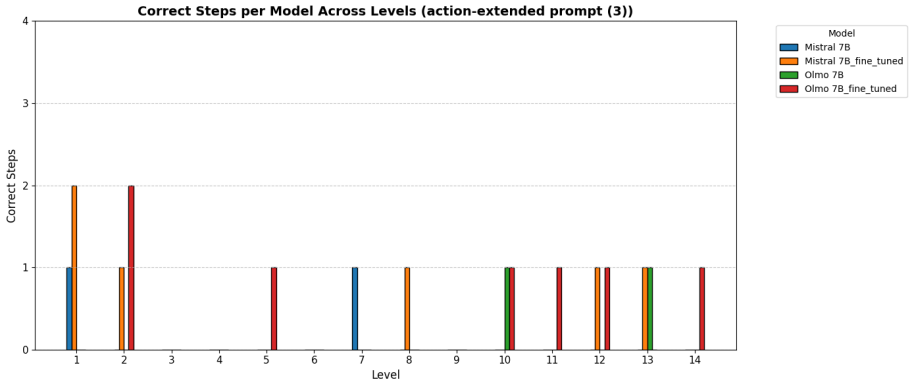


Fig. 10: Correct steps per model across the 14 Baba is You levels with the action-extended prompt. The finetuned models have more correct steps across the levels but still not enough to fully solve the levels.

text blocks (Figure 8). However, the model still struggles with correctly distinguishing between them. The generated solutions suggest that the finetuned model still has difficulty grasping the game mechanics, rarely proposing actions such as breaking or creating rules. Additionally, it sometimes treats text blocks as the objects you control. Finetuning the models with textual data from the game *Baba is You* led to improvements in level analysis for both models. In the case of Mistral 7B, there was an improvement in classifying text blocks and objects, while for OLMo 7B, the problem statement formulation showed better results. However, for both models, there was no clear improvement in solving the puzzle, as the generated solutions and actions still contained many errors.

5 Discussion & Conclusion

In the puzzle game *Baba is You* the goal is to win by following rules and by creating new rules, tasks that involve both language and reasoning abilities. LLMs must be able to move (*mention*) game pieces in such a way that they align to form new rules (*use*). This study explores how various LLMs perform on 14 relatively simple game levels: how well they are able to solve levels by understanding the consequences of rule manipulation and spatial understanding. We used prompt-based-learning first, finetuning second. Among the models evaluated with prompt-based-learning, GPT-4o and Gemini Flash 1.5 consistently outperform smaller models in identifying objects, interpreting game mechanics, and generating partially correct solution paths. However, even these more advanced models struggled to accurately interpret the grid as a two-dimensional space, often overlooking critical constraints such as rule-breakability. Finetuning on structured textual data led to improvements, Mistral 7B showed better classification and OLMo 7B improved in formulating problem statements, but neither model demonstrated substantial gains in full solution generation. Mistral and OLMo continued to struggle with core aspects of the game such as distinguishing between text and object blocks and understanding how rule creation or breaking is physically performed in the game.

This work shows that while high-end models like GPT-4o and Gemini Flash 1.5 can reason through parts of *Baba is You* levels, they still struggle to fully understand the game. Furthermore, without explicit prompts that include active rules and structured action formats, their performance drops significantly. A common limitation across all models is that they fail to interpret the grid as a two-dimensional space, leading to incorrect or overly simplistic solutions. Even GPT-4o often fails to recognize which rules can be broken, and it is unclear whether LLMs truly grasp the mechanism of rule manipulation through moving text blocks. Smaller models such as Mistral and OLMo, even when finetuned, frequently misinterpret game elements and fail to demonstrate a solid understanding of the mechanics.

Complex reasoning tasks such as *Baba is You* pose three types of challenges to an LLM: challenges of (1) representation, (2) reasoning, and (3) reflection [Madaan et al., 2023, Schultz et al., 2024, Plaat et al., 2024].

Representation First, the LLM must be able to represent puzzle states correctly. In Chess, work on ChessGPT has shown that pretraining and finetuning can teach an LLM to recognize positions and solve problems correctly [Karvonen, 2024, Feng et al., 2024]. In OthelloGPT, Li et al. [2023], Nanda et al. [2023] have used mechanistic interpretability to show how pretrained LLMs represent boards internally. In our study of *Baba is You* LLMs were not pretrained on the game, and the LLMs have difficulty with the spatial interpretation of the board. Further finetuning and pretraining may be necessary for improvement.

Reasoning Second, in order to correctly manipulate the state representations, the LLM must be able to reason with the rules, for example, to follow chains of thought [Wei et al., 2022]. Schultz et al. [2024], Zhang et al. [2025] show that by pretraining and finetuning on textual representations of Chess, LLMs can learn to reason well enough to play correct games (although not yet at a high level of play). In *Baba is You*, we also saw that finetuning was able to enhance reasoning.

Reflection Third, in *Baba is You* the LLM must be able to reflect on its own reasoning to understand the effect (*use*) of the rules that it composes (*mention*). Reasoning LLMs typically apply reinforcement learning to reflect on their own actions, using an external algorithm to control the self-reflection process [Madaan et al., 2023, Shinn et al., 2023, Yao et al., 2023]. In *Baba is You*, the LLMs achieve weak use-mention-type reasoning about dynamic rules, with prompts based on Plan-and-Solve [Wang et al., 2023b]. Achieving accurate use-mention reflection in *Baba is You* may require such explicit external algorithms or methods such as analogical prompting [Yasunaga et al., 2023].

Limitations & Further work Chain of thought (CoT) prompting [Wei et al., 2022] has spawned active research in methods for reasoning. This research used plan and solve [Wang et al., 2023b], a zero-shot CoT prompting method. Further research may explore other prompting methods, for example using explicit step-by-step prompting [Shinn et al., 2023, Press et al., 2023, Madaan et al., 2023].

The evaluation was conducted on 14 relatively simple levels, which may not reflect model performance on more challenging puzzles. More levels, with varying difficulty, could provide deeper insights. Furthermore, the finetuning dataset that we used was small, with only 15 examples for level solutions and 298 game mechanics questions. This restricts the model's exposure to the game's complexity. Expanding the dataset with more varied levels, solution paths, and mechanism-related questions may improve generalization and reasoning performance. In general, larger models tend to perform better at test time inference [Muennighoff et al., 2023], as the performance of GPT-4o in our work also indicated. Therefore, especially for reasoning and reflection, further research with larger models is warranted. Additionally, more extensive finetuning with adjusted hyperparameters (e.g., learning rate, batch size, or epochs) might yield better results. Finally, error analysis in this work was performed manually, introducing potential subjectivity. Future work could implement automated evaluation tools to ensure more consistent and scalable assessment.

Bibliography

Anonymous. Llm_babaisyou. https://github.com/anon, 2025a.

Anonymous. Mistral_7b_instruct-baba. https://huggingface.co/anon, 2025b.

Anonymous. Olmo_7b_instruct-baba. https://huggingface.co/anon, 2025c.

Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, et al. Dota 2 with Large Scale Deep Reinforcement Learning, December 2019. URL http://arxiv.org/abs/1912.06680. arXiv:1912.06680 [cs].

Isaiah Bjorklund. cot-logic-reasoning, 2025. URL https://huggingface.co/datasets/isaiahbjork/cot-logic-reasoning.

Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, January 2018. ISSN 0036-8075, 1095-9203. https://doi.org/10.1126/science.aao1733.

Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

M Charity and Julian Togelius. Keke AI Competition: Solving puzzle levels in a dynamically changing mechanic space. In *2022 IEEE Conference on Games (CoG)*, pages 570–575, August 2022. https://doi.org/10.1109/CoG51982.2022.9893650. ISSN: 2325-4289.

Megan Charity, Ahmed Khalifa, and Julian Togelius. Baba is y'all: Collaborative mixed-initiative level design. In *2020 IEEE Conference on Games (CoG)*, pages 542–549, 2020. https://doi.org/10.1109/CoG47356.2020.9231807.

Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. RLHF Deciphered: A Critical Analysis of Reinforcement Learning from Human Feedback for LLMs, April 2024. URL http://arxiv.org/abs/2404.08555. arXiv:2404.08555 [cs].

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. A survey of chain of thought reasoning: Advances, frontiers and future. In *Association for Computational Linguistics*, 2024.

Nathan Cloos, Meagan Jens, Michelangelo Naim, Yen-Ling Kuo, Ignacio Cases, Andrei Barbu, and Christopher J Cueva. Baba is ai: Break the rules to beat the benchmark. *arXiv preprint arXiv:2407.13729*, 2024.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey on in-context learning. In *Association for Computational Linguistics*, 2023.

Xidong Feng, Yicheng Luo, Ziyan Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Mguni, Yali Du, and Jun Wang. Chessgpt: Bridging policy learning and language modeling. *Advances in Neural Information Processing Systems*, 36, 2024.

Team Gemini. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, December 2024. URL http://arxiv.org/abs/2403.05530. arXiv:2403.05530 [cs].

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey, September 2024. URL http://arxiv.org/abs/2403.14608. arXiv:2403.14608 [cs].

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, October 2021. URL http://arxiv.org/abs/2106.09685. arXiv:2106.09685 [cs].

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In *Assoc for Computational Linguistics*, 2023.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Cheonsu Jeong. Fine-tuning and utilization methods of domain-specific llms. *arXiv preprint arXiv:2401.02981*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, et al. Mistral 7B, October 2023. URL http://arxiv.org/abs/2310.06825. arXiv:2310.06825 [cs].

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, et al. Mixtral of Experts, January 2024. URL http://arxiv.org/abs/2401.04088. arXiv:2401.04088 [cs].

Uday Kamath, Kevin Keenan, Garrett Somers, and Sarah Sorenson. Prompt-based Learning. In *Large Language Models: A Deep Dive: Bridging Theory and Practice*, pages 83–133. Springer Nature Switzerland, Cham, 2024. ISBN 978-3-031-65647-7. https://doi.org/10.1007/978-3-031-65647-7_3.

Adam Karvonen. Emergent world models and latent variable estimation in chess-playing language models. *arXiv preprint arXiv:2403.15498*, 2024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *ICLR*, 2023.

Yinghao Li, Haorui Wang, and Chao Zhang. Assessing Logical Puzzle Solving in Large Language Models: Insights from a Minesweeper Case Study. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 59–81, Mexico City, Mexico, 2024. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.4.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

Alessandro Marincioni, Myriana Miltiadous, Katerina Zacharia, Rick Heemskerk, Georgios Doukeris, Mike Preuss, and Giulio Barbero. The effect of llm-based npc emotional states on player emotions: An analysis of interactive game play. In *2024 IEEE Conference on Games (CoG)*, pages 1–6. IEEE, 2024.

Suvir Mirchandani, Fei Xia, Pete Florence, Brian Ichter, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, and Andy Zeng. Large Language Models as General Pattern Machines, October 2023. URL http://arxiv.org/abs/2307.04721. arXiv:2307.04721 [cs].

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36:50358–50376, 2023.

Matthias Müller-Brockhausen, Giulio Barbero, and Mike Preuss. Chatter generation through language models. In *2023 IEEE Conference on Games (CoG)*, pages 1–6. IEEE, 2023.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. *arXiv preprint arXiv:2309.00941*, 2023.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A Comprehensive Overview of Large Language Models, October 2024. URL http://arxiv.org/abs/2307.06435. arXiv:2307.06435 [cs].

David A. Noever and Ryerson Burdick. Puzzle solving without search or human knowledge: An unnatural language approach, 2021. URL https://api.semanticscholar.org/CorpusID:237431487.

Aske Plaat. *Learning to play: reinforcement learning and games*. Springer Nature, 2020.

Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. Reasoning with Large Language Models, a Survey, July 2024. URL http://arxiv.org/abs/2407.11511. arXiv:2407.11511 [cs].

Aske Plaat, Max van Duijn, Niki van Stein, Mike Preuss, Peter van der Putten, and Kees Joost Batenburg. Agentic large language models, a survey. *arXiv preprint arXiv:2503.23037*, 2025.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and Narrowing the Compositionality Gap in Language Models, October 2023. URL http://arxiv.org/abs/2210.03350. arXiv:2210.03350 [cs].

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training, 2018.

Paul Saka. Quotation and the use-mention distinction. *Mind*, 107(425):113–135, 1998.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

John Schultz, Jakub Adamek, Matej Jusup, Marc Lanctot, Michael Kaisers, Sarah Perrin, Daniel Hennes, Jeremy Shar, Cannada Lewis, Anian Ruoss, et al. Mastering board games by external and internal planning with language models. *arXiv preprint arXiv:2412.12119*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Arvi Teikari. Baba is you. *Game [PC].(March 2019). Hempuli Oy, Finland*, 2019.

Oguzhan Topsakal, Colby Jacob Edell, and Jackson Bailey Harper. Evaluating large language models with grid-based game competitions: an extensible llm benchmark and leaderboard. *arXiv preprint arXiv:2407.07796*, 2024.

Karthik Valmeekam, Kaya Stechly, Atharva Gundawar, and Subbarao Kambhampati. Planning in Strawberry Fields: Evaluating and Improving the Planning and Scheduling Capabilities of LRM o1, October 2024. URL http://arxiv.org/abs/2410.02162. arXiv:2410.02162 [cs].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, August 2023. URL http://arxiv.org/abs/1706.03762. arXiv:1706.03762 [cs].

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*, 2023a.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models, May 2023b. URL http://arxiv.org/abs/2305.04091. arXiv:2305.04091 [cs].

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Shomir Wilson. A bridge from the use-mention distinction to natural language processing. *The Semantics and Pragmatics of Quotation*, pages 79–96, 2017.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment, December 2023. URL http://arxiv.org/abs/2312.12148. arXiv:2312.12148 [cs].

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H Chi, and Denny Zhou. Large language models as analogical reasoners. *arXiv preprint arXiv:2310.01714*, 2023.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction Tuning for Large Language Models: A Survey, March 2024. URL http://arxiv.org/abs/2308.10792. arXiv:2308.10792 [cs].

Yinqi Zhang, Xintian Han, Haolong Li, Kedi Chen, and Shaohui Lin. Complete chess games enable llm become a chess master. *arXiv preprint arXiv:2501.17186*, 2025.

A Prompts and Error Snippets

<object 1> IS <object 2>	Transforms all instances of object 1 into object 2
<object 1> IS <object 1>	Set an object to be itself and therefore become immutable for the transformation rule.
<object> IS WIN	Makes the object the win condition; anything controlled by you that touches or is the object wins.
<object> IS YOU	Objects that are set to be YOU can be controlled by you. All objects will move simultaneously.
<object> IS PUSH	Objects set to PUSH can be moved by either an object controlled by you or by other movable objects, shifting them one tile forward and occupying the new tile if possible.
<object> IS MOVE	Objects move one tile per turn in their facing direction, reversing direction if they hit an object, and if an object is both YOU and MOVE, it moves two tiles per turn.
<object> IS STOP	Objects becomes impassible, important is that the nature of the PUSH rule overrides the blocking mechanism.
<object> IS SINK	Objects that are set to SINK will destroy themselves as well as any other object that comes in contact with them.
<object> IS HOT, <object> IS MELT	Objects set to HOT will destroy anything set to MELT that touches them. If an object is both HOT and MELT, it will destroy itself.
<object> IS KILL	Objects set to KILL destroy all objects controlled by you when they touch them, while remaining intact themselves. Note that the nature of the PUSH rule overrides this mechanic.

Table 3: Explanation of common rules in the game BABA IS YOU

LLM Model	Error Snippets
GPT-4o	**Level 13:** Goal: To win the level, BABA (controlled by the player) must reach the FLAG (f). However, the WALL blocks the direct path. Strategy: Modify the rules to bypass or remove the obstacle created by the WALL: Remove or break the "WALL IS STOP" rule. Create an accessible path for BABA to reach the FLAG.
	Level 4: Identify Barriers: BABA (controlled by the player) must reach the FLAG, but there is no direct path. ROCK (r) is blocking BABA's movement, and it has no properties assigned. Use Rules to Clear the Path: Assign ROCK IS PUSH by moving the text blocks R, 1, and 5 (if present) to form the rule. This will allow BABA to push the ROCK (r) out of the way. Reach the FLAG: With the ROCK out of the way, BABA can move directly to the FLAG.
Gemini 1.5 Flash	**Level 4:****Objects:** We have Baba (b), a Rock (r), and Flags (f). **Text Blocks:** BABA (B), IS (1), YOU (2), FLAG (F), and WIN (3). **Current Rules:** BABA IS YOU (you control Baba) and FLAG IS WIN (reaching a flag wins the level). **Plan** Currently, the "FLAG IS WIN" rule is broken because the IS (1) and WIN (3) are not next to the FLAG (F). We need to move the IS and WIN next to the FLAG to make the rule active again.
	Level 5: Plan The goal is to reach the Flag. However, the initial rules make all Rocks into Flags, which can be confusing and potentially block the path. We need to change the rules to make the Flag a win condition again and allow Baba to move freely.
	Level 13:**Solution** 1. **Breake_Rule[GOOP IS SINK]:** Remove the rule "GOOP IS SINK." This will prevent the Goop from destroying itself. 2. **Make_Rule[ROCK IS HOT]:** Create the rule "ROCK IS HOT." This will make the Rocks hot. 3. **Make_Rule[GOOP IS MELT]:** Create the rule "GOOP IS MELT." This will make the Goop melt when it touches the hot Rocks.

Table 4: Examples of error snippets in the reasoning chains of GPT-4o and Gemini 1.5 Flash LLM models, where the red-colored text indicates incorrect reasoning. These errors highlight challenges such as hallucinations, incorrect assumptions, and overlooking key constraints.

LLM Model	Error Snippets
Mistral 8x7B	**Level 1:** Step 3 Devise a plan - We need to move the BABA object to the FLAG object. - To do this, we need to change the rule BABA IS YOU to BABA IS MOVE. - Then, we need to push the ROCK object to the WALL object to create a path for the BABA object to reach the FLAG object.
	Level 3: Step 1: Understand the problem and extract the relevant objects, text blocks, and rules. Objects: BABA, FLOOR, GRASS, WALL, SKULL Text Blocks: 1, 2, 3 Rules: BABA IS YOU Step 2: Devise a plan to solve the problem. 1. Make BABA IS YOU to BABA IS MOVE. 2. Move BABA to the position of the SKULL. 3. Make SKULL IS YOU.
	Level 4: Step 2: Extract the relevant objects, text blocks and rules - Objects: BABA, SKULL, FLAG, FLOOR, GRASS, ROCK, WALL, KEKE, GOOP, LOVE, and empty space. - Text blocks: BABA, IS, YOU, FLAG, IS, WIN. - Rules: BABA IS YOU, FLAG IS WIN. Step 3: Devise a plan - Make BABA the object you control. - Make FLAG the object you need to reach to win the level.
Mistral 7B	**Level 1:** The goal is to move the BABA to the FLAG to win the level. Plan: 1. Move the BABA to the flag by pushing the blocks in the way. 2. Make the BABA move towards the flag. 3. Make the BABA push the flag towards itself.
	Level 3: Plan: 1. Move BABA to the 1 text block. 2. Change BABA into the flag (f) using the 1 text block. 3. Move the flag to the exit (the empty space next to the 3 text block).
	Level 9: The goal is to move the flag to the exit (position B12) without getting killed by the skull.
OLMo 13B	**Level 3:** **Objective:** Reach the WIN block to win the level.
	Level 4: Plan:** 1. **Make Baba movable:** Ensure Baba has the "MOVE" property.
	Level 9: 1. **Neutralize SKULL:** - We cannot directly push SKULL because it will kill BABA. - We need to use text blocks to change SKULL's properties. - **Make Rule:** SKULL IS STOP. This will make SKULL impassable and safe for BABA.
OLMo 7B	**Level 1:** ### Plan 1. **Identify the Goal:** Move BABA to the FLAG (F13). 2. **Current Obstacles:** BABA starts at B12, and there are walls (W) and empty spaces (.) blocking the path.
	Level 2: 2. **Initial Setup:** - BABA is currently next to a WALL (W) and a KEKE (K). - There's a FLAG (F) adjacent to BABA.
	Level 9: ### Plan: 1. **Move BABA away from SKULL:** To avoid immediate death, move BABA away from SKULL. 2. **Create a path to FLAG:** Use the environment to create a path to FLAG. 3. **Utilize GOOP:** GOOP can be used to block SKULL or create a bridge

Table 5: Examples of error snippets in the reasoning chains of OLMo and Mistral LLM models, where the red-colored text indicates incorrect reasoning. These errors highlight challenges such as hallucinations, incorrect assumptions, and wrong reasoning steps.

Model Input

You are helping to solve a gridworld game. In *Baba is You*, the player can change the game rules by moving text blocks around. The grid contains object text and property text.
Text Blocks:
Object Text: Words representing game objects.
Property Text: Words that describe actions or properties.
Active Rules:
A rule is formed when the text blocks are aligned in a valid way, either horizontally or vertically. Valid rule formats include:
`<object_text> IS <property_text>`: Grants a property to an object.
`<object1_text> IS <object2_text>`: Changes one object into another.
`<object1_text> IS <object1_text>`: Makes an object immutable.
The goal is to use these rules to solve the level by moving the text blocks and controlling the objects.

The level is displayed in a 2D grid, where each character has the following meaning:
Text blocks in the game which always can be pushed:
`<object_text>`:
B = BABA, S = SKULL, F = FLAG, O = FLOOR, A = GRASS, L = LAVA, R = ROCK, W = WALL, K = KEKE, G = GOOP, V = LOVE

`<property_text>`:
1 = IS, 2 = YOU, 3 = WIN, 4 = KILL, 5 = PUSH, 6 = STOP, 7 = MOVE, 8 = HOT, 9 = MELT, 0 = SINK

Objects in the game:
`<object>`:
b = object baba, s = object skull, f = object flag, o = object floor, a = object grass, l = object lava, r = object rock, w = object wall, k = object keke, g = object goop, v = object love, _ = border, . = empty space

`<object_text> IS YOU`: Makes the object you control in the game. You can move it and push blocks.
`<object_text> IS WIN`: The object you need to reach or to be to win the level.
`<object_text> IS STOP`: Makes the object impassable (you can't move through it).
`<object_text> IS MELT`: Makes the object melt when touched by something marked as HOT.
`<object_text> IS HOT`: Makes the object destroy any object marked as MELT when they touch it. If the same object is both HOT and MELT it self-destructs.
`<object_text> IS MOVE`: Makes the object move one step in its direction every turn.
`<object_text> IS KILL`: Makes the object destroy anything you control when touched, but it stays intact.
`<object_text> IS PUSH`: Lets you push the object or have it pushed by other moving objects.
`<object_text> IS SINK`: Makes the object destroy itself and anything it touches when it is touched.

Question: Give a solution to the following grid level:

```
__________\n_B12..F13_\n_........_\n_........_\n_........_\n_.b....f._\n
_........_\n_........_\n_........_\n__________\n
```

Let's first understand the problem, extract the relevant objects, text blocks and rules (explain the rules) in the level and make a plan to solve the problem. Then let's carry out the plan by giving the intermediate actions (using common sense). Solve the problem step by step and show the solution.

Fig. 11: **Simple prompt**: consisting only of a short game description and definitions of the characters and rules. Followed by a question to solve a level with at the end a sentence to activate zero-shot CoT.

The Rule-extended prompt and the Action-extended prompt can be found on GitHub [Anonymous, 2025a,c,b].