Step-by-Step Reasoning to Solve Grid Puzzles: Where do LLMs Falter?

Nemika Tyagi[1*] **Mihir Parmar**[1*] **Mohith Kulkarni**[1] **Aswin RRV**[1] **Nisarg Patel**[1]
Mutsumi Nakamura[1] **Arindam Mitra**[2] **Chitta Baral**[1]

[1]Arizona State University [2]Microsoft Research
{ntyagi8, mparmar3, chitta}@asu.edu

Abstract

Solving grid puzzles involves a significant amount of logical reasoning. Hence, it is a good domain to evaluate the reasoning capability of a model which can then guide us to improve the reasoning ability of models. However, most existing works evaluate only the final predicted answer of a puzzle, without delving into an in-depth analysis of the LLMs' reasoning chains (such as where they falter) or providing any finer metrics to evaluate them. Since LLMs may rely on simple heuristics or artifacts to predict the final answer, it is crucial to evaluate the generated reasoning chain beyond overall correctness measures, for accurately evaluating the reasoning abilities of LLMs. To this end, we first develop *GridPuzzle*, an evaluation dataset comprising 274 grid-based puzzles with different complexities. Second, we propose a new error taxonomy derived from manual analysis of reasoning chains from LLMs including GPT-4, Claude-3, Gemini, Mistral, and Llama-2. Then, we develop an LLM-based framework for large-scale subjective evaluation (i.e., identifying errors) and an objective metric, *PuzzleEval*, to evaluate the correctness of reasoning chains. Evaluating reasoning chains from LLMs leads to several interesting findings. We further show that existing prompting methods used for enhancing models' reasoning abilities do not improve performance on *GridPuzzle*. This highlights the importance of understanding fine-grained errors and presents a challenge for future research to enhance LLMs' puzzle-solving abilities by developing methods that address these errors[1].

1 Introduction

Recent advancements in LLMs such as GPT-4, Gemini, Claude-3 (Anthropic, 2024), Llama-2 (Touvron et al., 2023), and Mistral (Jiang et al., 2023) have achieved remarkable performance on a wide range of Natural Language Understanding (NLU) tasks previously thought to be exclusive to humans. Beyond NLU, exploring LLMs' logical reasoning abilities (Liu et al., 2021; Saparov and He, 2022; Parmar et al., 2024; Patel et al., 2024) on complex tasks such as puzzle-solving is under-explored. Past attempts have been made to evaluate models on logic-intensive grid-based puzzle-solving. However, they either do not focus on evaluating LLMs (Mitra and Baral, 2015; Jabrayilzade and Tekir, 2020) or do not evaluate LLMs independently, but rather use neuro-symbolic approaches (Ishay et al., 2023) that use external specialized solvers on LLM outputs. Here, we aim to evaluate the puzzle-solving abilities of LLMs by themselves, without the use of any external logic solvers.

To understand the reasoning capabilities of LLMs, it is important to evaluate reasoning chains, rather than the final predicted answer. There have been works that evaluate reasoning chains using objective metrics such as ROSCOE (Golovneva et al., 2022), CTC (Deng et al., 2021), and BARTScore (Yuan et al., 2021), however, they do not focus specifically on evaluating reasoning. Some prior works propose metrics for specific reasoning tasks, such as FOLIO (Han et al., 2022) and ProntoQA (Saparov and He, 2022). However, these methods rely on reference-based evaluation, do not focus on puzzle-solving, and do not aim to identify fine-grained errors in reasoning chains. To address these limitations, we propose a reference-free manual and automated subjective evaluation of reasoning chains to understand various fine-grained errors in reasoning chains for grid-based puzzle-solving.

Motivated by Mitra and Baral (2015), we first develop *GridPuzzle* (Figure 1), a comprehensive evaluation dataset consisting of grid-based puzzles with grid-size of 3×4, 3×5, 4×4, 4×5, and 4×6 with three levels of difficulty (easy, medium, and hard). Then, we evaluate LLMs including GPT-

[1]Data and source code are available at https://github.com/Mihir3009/GridPuzzle

*Equal Contribution

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 19898–19915
November 12-16, 2024

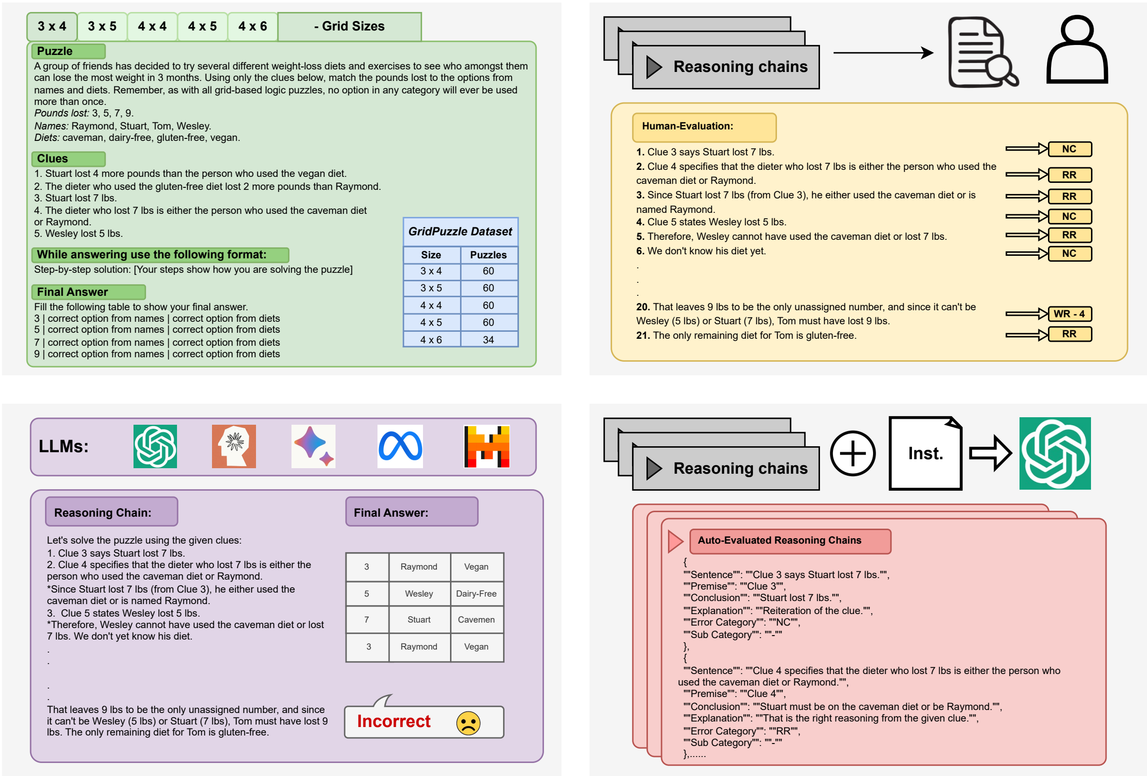

Figure 1: Schematic representation of proposed pipeline. Begins with the data collection of *GridPuzzle* dataset (top left) and evaluating various LLMs in zero-shot CoT setting (bottom left), then analyzing reasoning chains of LLMs manually to find various error types (top right) and automate this analysis process using LLM to check the correctness of reasoning chain by finding errors (bottom right).

4, Gemini-Pro, Claude-3, Llama-2, and Mistral on *GridPuzzle* in zero-shot-CoT setting (Figure 1). Experimental results show that LLMs do not fare well and achieve a maximum of 5.1% accuracy.

To investigate the reasoning chains, we manually analyze them (Figure 1) to find fine-grained errors (further details in section 3.3). Based on this, we propose a new error taxonomy comprising five broad categories, and nine fine-grained subcategories (Tables 1 and 2), providing deeper insights into the primary causes of the LLMs' reasoning failures. However, scaling manual analysis to a larger set is time-consuming and laborious. Hence, we propose to leverage LLMs as auto-evaluators by creating prompts that utilize error taxonomy to automate the analysis of reasoning chains and help in identifying errors (Figure 1). While evaluating w.r.t. manual annotation, our auto-evaluator model achieves $\sim 86\%$ agreement, hence providing quality error categorization.

Beyond identifying errors and the accuracy of the final answer, we propose *PuzzleEval*, an LLM-based framework to evaluate reasoning chains for grid-based puzzles. *PuzzleEval* involves a multi-stage evaluation using GPT-4o. First, we identify key logical conclusions from the reasoning chain; second, we extract key logical concepts from these conclusions; and finally, we measure the presence of these logical concepts in the final gold answer to assess the correctness of the reasoning chain. Evaluating reasoning chains based on error categorization and *PuzzleEval* reveals interesting findings such as LLMs show lower accuracy despite having more error-free reasoning steps, open-source models lack reasoning skills compared to closed-source models, and the most dominant error categories are wrong reasoning and elimination. Additionally, we employ existing prompting methods such as Plan-and-Solve and Self-discover, demonstrating that these methods do not improve performance on *GridPuzzle*. We believe that our findings will inspire future work in the automated in-depth evaluation of reasoning chains for broader reasoning tasks and enhance the reasoning abilities of models.

2 Related Work

Puzzle-solving Task Puzzle-solving task provides insights into LLMs' logical reasoning. Giadikiaroglou et al. (2024) categorize puzzles into (1) rule-based and (2) rule-less puzzles. Rule-less puzzles include riddles (Lin et al., 2021), MCQs (Zhao and Anderson, 2023), programming puzzles (Schuster et al., 2021), and commonsense reasoning puzzles (Gu et al., 2023); however, in our work we focus on rule-based puzzles. In rule-based puzzles, past attempts have explored Sudoku (Noever and Burdick, 2021), Rubik's Cube, 8-puzzle, Game of 24 (Yao et al., 2024), crosswords (Yao et al., 2024), chess puzzles (Feng et al., 2024), card games (Gupta, 2023), BoardgameQA (Kazemi et al., 2024), and Lateral Thinking Puzzles (Huang et al., 2024). However, grid-based puzzle solving is under-explored. Mitra and Baral (2015) proposed a grid-based puzzle dataset and Dziri et al. (2023) studied compositionality in LLMs using Grid Puzzle, but these works do not provide any insights into the performance of recent LLMs. Motivated by this, we propose a systematically curated grid-based puzzle dataset, *GridPuzzle*, and provide an evaluation of various LLMs in puzzle-solving.

Automatic Evaluation of Reasoning Chains Previous works (Dalvi et al., 2021; Saparov and He, 2022; Han et al., 2022) have focused on reference-free evaluation, which is not reliant on gold-reasoning chains. Recently, ROSCOE (Golovneva et al., 2022) proposed a suite of metrics to measure the semantic consistency, logicality, informativeness, fluency, and factuality of reasoning chains, while the ReCEval framework (Prasad et al., 2023) evaluates reasoning chains based on two key properties: correctness and informativeness. Recent evaluation methods such as *LLM evaluation* (Chiang and Lee, 2023) and G-Eval (Liu et al., 2023) leverage LLMs to measure the quality of reasoning chains. *LLM evaluation* involves presenting task instructions and a text sample to LLMs, asking them to rate the sample's quality on a 5-point Likert scale, whereas the latter incorporates automatic chain-of-thought generated by the LLM describing the detailed evaluation steps. Additionally, Tyen et al. (2023)'s attempt to use GPT-4 as an evaluator in a few-shot setting, shows that evaluating reasoning chains remains a challenge. Furthermore, AutoRace (Automatic Reasoning Chain Evaluation) (Hao et al., 2024) proposed a fully automated approach for evaluating reasoning chains that adapt to different tasks without human effort. However, these methods do not evaluate reasoning chains at the level of fine-grained error types and do not provide detailed task-specific insights. To address this, we propose LLM-based reference-free evaluation methods that identify fine-grained errors and assess the correctness of generated reasoning chains.

3 Evaluation of Reasoning Chains

3.1 *GridPuzzle*

To develop this dataset, we extract logic grid puzzles of various grid sizes from Puzzle Baron's Logic Puzzles[2]. Specifically, we compile logic grid puzzles of size 3×4, 3×5, 4×4, 4×5, and 4×6. Each grid size has three levels of difficulty (easy, medium, and hard) except 4×6. This particular grid size has only two difficulty levels (Easy and Medium). Statistics corresponding to each grid size are presented in Figure 1 (top left).

Error Category	Description
WW	Wrong Premise and Wrong Conclusion
WR	Wrong Premise and Right Conclusion
RW	Right Premise and Wrong Conclusion
RR	Right Premise and Right Conclusion
NC	No Conclusion statement or no reasoning involved

Table 1: Proposed error taxonomy for broad categories based on manual analysis. If a statement starts with "so, therefore, hence, this means, this implies, etc." and/or is not followed by any premise, consider the previous statement's conclusion or the previous NC as the premise.

3.2 Manual Evaluation

To explore where exactly these LLMs falter in performing reasoning, we conduct a detailed manual analysis of the reasoning chains generated by them while solving grid-based puzzles. Details of the annotation guidelines provided to the human evaluators are given in the Appendix D. Our manual analysis process consists of three steps. First, we begin by segmenting the reasoning chains into individual sentences, allowing us to categorize errors more precisely. Second, we identify the premise and conclusion for each sentence and determine their respective correctness. We refrain from subdividing sentences into multiple premises or conclusions to maintain simplicity for finding errors. At last, each sentence is categorized as either containing a single premise and conclusion or being a declarative statement without a conclusion. Then, we begin assessing potential issues or errors in the reasoning chains. Now, we follow an exhaustive

[2] https://logic.puzzlebaron.com/

Category	Source	Sub-Category	Description
Wrong Premise or No Conclusion	From the clues (Example: From clue 4,....)	(1) Hallucination	When information is completely out of context and not present in clues.
		(2) Incomplete Information	Lacks necessary information to make a particular conclusion.
		(3) Assumptions	Statements not derived from clues directly; might include assumed information relevant to the clue.
	Derived Conclusions using clues given in puzzle which was not inherently given in the clues.	(4) Error Propagation	Premise derived from a previous incorrect conclusion.
		(5) Incomplete Information	Lacks necessary information to make a particular conclusion.
		(6) Wrong Assumption	The derived assumption is incorrect.
Wrong Conclusion	Derived using the premise (which itself is either taken directly from the clues or derived)	(a) Wrong Reasoning	The reasoning is incorrect, regardless of the premise's accuracy.
		(b) Error propagation	Conclusion is incorrect due to an erroneous premise.
		(c) Wrong Elimination	All premises are present, but not all conclusions are correctly derived.

Table 2: Proposed error taxonomy for sub-categories based on manual analysis. These sub-categories are defined for cases where either the conclusion or premise is incorrect ("RW" or "WR") or both are incorrect ("WW"). For "WW", the error sub-categories might appear in any combinations between (1-6) and (a-c) such as '1a', '4b', or '6c'.

approach to create fine-grained error categories. We begin with 30 reasoning chains (6 puzzles x 5 reasoning chains from LLMs) to manually identify potential errors. Next, we categorize these errors in a structured format. We then add another 30 reasoning chains to see if any new types of errors emerge. If new errors are identified, we refine our categories accordingly. This process is repeated until we evaluate a total of 150 reasoning chains and no new types of errors are found. Based on this method, we have carefully filtered and categorized several errors made by LLMs, presenting them as five broad categories and nine sub-categories.

3.3 Proposed Error Taxonomy

Broad Categories As shown in Table 1, we present five main categories: "WW" - *Wrong Premise Wrong Conclusion*, "WR" - *Wrong Premise Right Conclusion*, "RW" - *Right Premise Wrong Conclusion*, "RR" - *Right Premise Right Conclusion*, and "NC" - *No Conclusion*. These acronyms of broad categories are self-explanatory. For instance, the category "WW" comprises sentences where the sentence consists of a wrong premise as well as a wrong conclusion. Interestingly, we also find the "WR" category consists of instances where a wrong premise still leads to a correct conclusion. Additionally, sentences containing only information from clues or premises from previous steps fall under "NC". We conduct further investigation as to why the premises and conclusions become incorrect.

Sub-categories: Wrong Premise As shown in Table 2, we identified the source of the premise to determine the origin of errors: (i) '***From Clues***' – where the premise is directly borrowed from one of the clues without any further reasoning, and (ii) '***Derived***' – where the premise is inferred from either the clues or the previous conclusions. From Table 2, there are six possible reasons associated with two different sources for the wrong premise. When the premise originates from the source (i), we find three types of errors: **Hallucination** – When some factual information from the clues is distorted or completely made up; **Incomplete information** – When the information is correctly borrowed from the clues but it is not sufficient to make a particular conclusion; **Assumptions** – This is a special category where the premise is not derived but also not given exactly in the clues. It is often related to one of the clues and is of the form, "Let's assume" or "Assuming that." When source is *derived*, we find three different errors: **Error Propagation** – This occurs when a previously incorrect conclusion becomes the basis for a flawed premise, thereby extending the error from one conclusion to the next; **Incomplete information** – When the derived premise is not sufficient to make a particular conclusion; and lastly, **Wrong Assumption** – When the LLM reasoner clearly states that a premise was an assumption but it was incorrectly derived.

Sub-categories: Wrong Conclusion As shown in Table 2 (source), conclusions are always logically derived from a fixed set of premises. For having a wrong conclusion in any reasoning step, we find 3 errors responsible: **Error Propagation** – When a conclusion is wrong strictly due to some error in the preceding premise; **Wrong Elimination** – When the conclusion is wrong because the LLM reasoner failed to eliminate all the unfit choices correctly. This case is specific to the grid-based puzzle task but is inherently an erroneous deduction on the

Examples of reasoning chain evaluated by GPT-4o
Sentence:"Therefore, Zeno must be 69%, and UCLA must be 62%.", **Premise:** "If Zeno were 55%, there would be no score 7% lower than 55% for UCLA.", **Conclusion:** "Zeno must be 69%, and UCLA must be 62%", **Explanation:** "The conclusion is incorrect as UCLA is already known to be 62% from clue 2.", **Error Category:** "RW.", **Premise:** "A",
Sentence:"Since the third performer used flashlights, it must be either Lora or Carmen.", **Premise:** "The performer who used flashlights was either Lora or Carmen.", **Conclusion:** "The third performer must be either Lora or Carmen.", **Explanation:** "The conclusion is based on the incorrect premise that the third performer used flashlights.", **Error Category:** "WW", **Sub Category** "4B",

Table 3: Examples of reasoning chain evaluated by Auto-evaluator (GPT-4o).

LLM's end; **Wrong reasoning** – The remaining incorrect conclusions that did not fit in the above categories are classified under this label.

3.4 Automated Evaluation

Manual analysis of reasoning chains provides a detailed categorization of errors; however, it is tedious and, therefore, challenging to scale for the entire dataset. However, analyzing the distribution of errors from our proposed taxonomy on the whole dataset is also crucial in understanding the shortcomings of LLMs' reasoning ability. Thus we develop an LLM-based auto-evaluator to automate the process of error evaluation. To this end, we prompt the GPT-4o model to identify and categorize errors in the given reasoning chain. Our prompt consists of a system instruction followed by a user prompt containing the reasoning chain to be evaluated along with the original puzzle and its gold solution. The system prompt can be further dissected into 3 key components: the instructions, the knowledge, and an exemplar. The **instruction** contains all the rules that the GPT-4o needs to follow to conduct accurate evaluation and error categorization of the reasoning chains. It incorporates similar sequential steps used during the manual evaluation of reasoning chains along with the required output format. The **knowledge** has a detailed description of our error taxonomy including the broad and sub-categories. We also provide a preference order for selecting categories along with the description to minimize any ambiguity in the evaluation process. Lastly, the **exemplar** consists of a puzzle, its correct solution, the original model-produced reasoning chain, and the manually evaluated reasoning chain with our error categories. We termed this LLM-based evaluator as "Auto-evaluator". Appendix B provides the structure of the Auto-evaluator prompt.

Using the Auto-evaluator, we evaluated a total of 1,370 reasoning chains generated by five different LLMs for solving 274 puzzles. The application of our Auto-evaluator to this large dataset allowed us to analyze the distribution of error categories on a broader scale. To validate the accuracy of the evaluations performed by the Auto-evaluator, we randomly sampled 20 reasoning chains from the manually evaluated set. The authors then compared their error category assignments to those given by the Auto-evaluator. The agreement score for the total number of reasoning steps between the manual evaluation and the GPT-4o evaluation is $\sim 86\%$. Table 3 shows the example of reasoning steps evaluated by GPT-4o.

4 Experimental Setup

4.1 Experiments

We evaluate a range of closed-source LLMs including GPT-4-Turbo, Claude-3-Opus, and Gemini-Pro, and open-source models Llama-2-13B, and Mistral-7B-Instruct on *GridPuzzle* in the Zero-shot-CoT setting (Kojima et al., 2022). We also conducted a scaling experiment on Llama-2-70B and the results are given in the Appendix F. Our *GridPuzzle* dataset consists of a set of instances denoted as $\mathcal{P} = < p_n^{i \times j}, a_n >$, where $p_n^{i \times j}$ is n^{th} puzzle instance with grid size of $i \times j$ and a_n as a gold answer. We prompt each LLM to generate a reasoning chain before predicting answer $\hat{a}$. To evaluate each model in the Zero-shot-CoT setting, we provide $< I, p_n^{i \times j} >$ as input to the model and predict an answer $\hat{a}$ where I is a natural language instruction. The evaluation is conducted on the OpenAI, Google, and Anthropic model versions released in April 2024 with temperature setting 0 for deterministic predictions. NVIDIA A100 GPUs are used for conducting the inference of open-source models with a batch size of 4. The example prompts used for these experiments are provided in Appendix A.

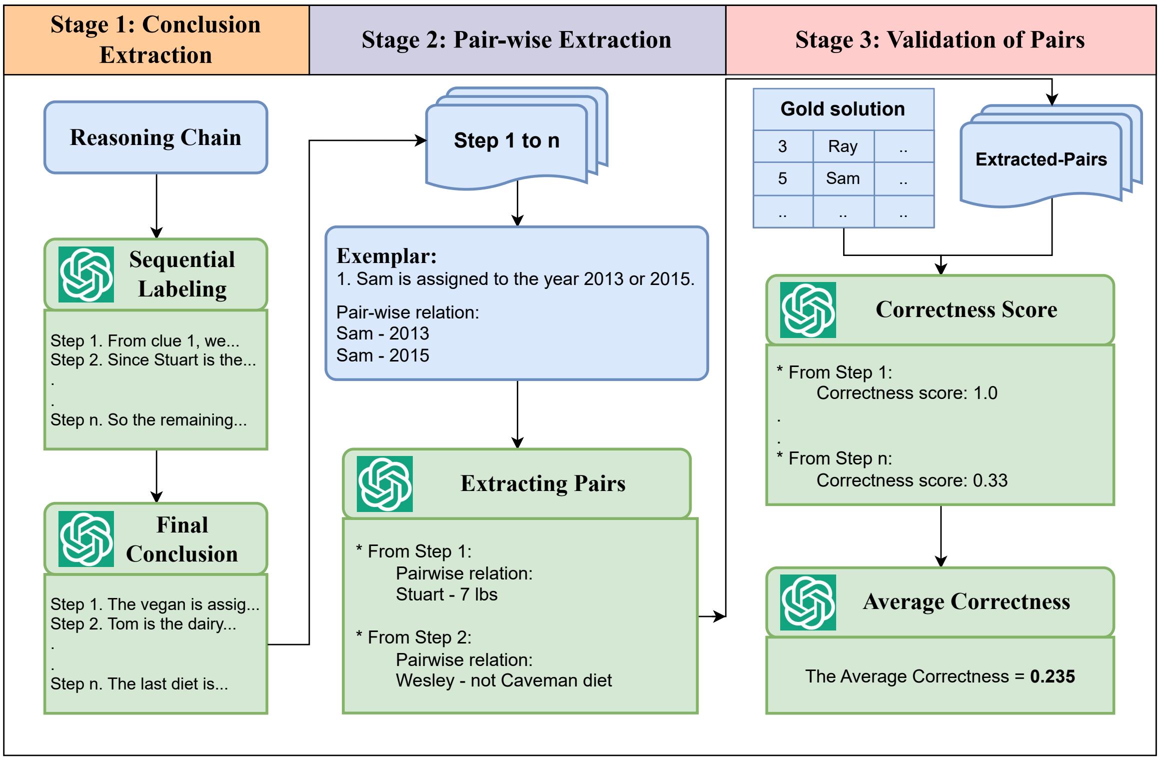

Figure 2: The process of calculating *PuzzleEval* metrics is described above. The reasoning chains are produced by our five LLMs and the gold solution is taken from our *GridPuzzle* dataset.

4.2 Metrics

Accuracy We use accuracy to demonstrate the capability of LLMs in solving grid-based puzzles based on their ability to predict the final answer. To calculate this metric, we use the LLM-generated final answers and compare them with the available gold solution. The predicted answers and the gold solution are in the form of tables with the number of rows and columns equal to the grid size of the puzzle. We perform an Exact Match (EM) to compare the two tables and mark them as correct only when all the entries of the tables match. See the example of the final answer table in Appendix C.

PuzzleEval We developed this LLM-based metric to assess step-by-step reasoning chains and provide a *correctness score* for each step, as well as the Average Correctness Score (ACS) for the entire chain. *PuzzleEval* is a reference-free metric specifically designed for assessing reasoning chains generated for grid-based puzzle tasks. It evaluates the correctness of each step in the reasoning chain and reports the score using only the final answer table provided as the gold solution, without requiring any comparison to a gold reasoning chain.

As shown in Figure 2, *PuzzleEval* consists of a three-stage pipeline to evaluate any reasoning chain.

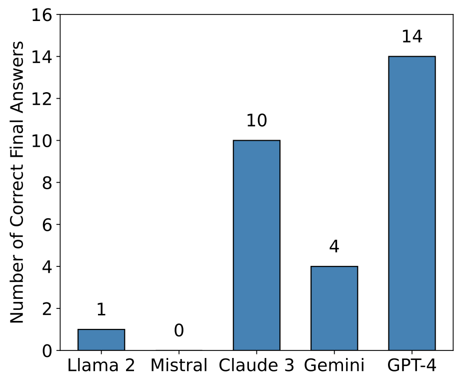

Figure 3: Performance of five different LLMs in terms of accuracy on the *GridPuzzle* dataset.

First, we prompt GPT-4o to label all the steps sequentially to account for any discrepancies in the different formats of reasoning chains produced by various models, and to extract only the final conclusions from each step. This stage is crucial as it removes the portion of a step where the models just reiterated clues or previous conclusions. Second, we instruct the model to extract the pair-wise relation of elements from the puzzle that have been either accepted or rejected in the extracted final conclusions. If the extracted conclusion is *"Sam is assigned to the year 2015 but not 2014."*, these pairs are of the form “Sam – 2015” or “Sam – not 2014”. Third, we provide the gold solution table and ask the model to check if these accepted or rejected pairs match the given information. As per the validation, the pairs extracted from every step are marked as correct or incorrect. After obtaining this information for each step the *correctness score* is calculated by adding up all the correct and incorrect steps (correct pairs are marked 1 and incorrect pairs are marked 0) divided by the total number of pairs in each step. Finally, the ACS is determined by adding up all the *correctness scores* from each step and dividing by the number of steps to capture the overall quality of the reasoning chain. Hence, *PuzzleEval* provides ACS for each reasoning chain in range of 0 to 1.

Model	3 x 4	3 x 5	4 x 4	4 x 5	4 x 6	Avg
Llama	0.45	0.46	0.46	0.42	0.28	**0.41**
Mistral	0.29	0.26	0.27	0.26	0.27	**0.27**
Claude	0.60	0.56	0.52	0.55	0.46	**0.54**
Gemini	0.60	0.64	0.54	0.52	0.62	**0.58**
GPT-4	0.61	0.62	0.56	0.54	0.60	**0.59**

Table 4: The results for *PuzzleEval* on the different grid sizes available in *GridPuzzle* dataset in terms of ACS.

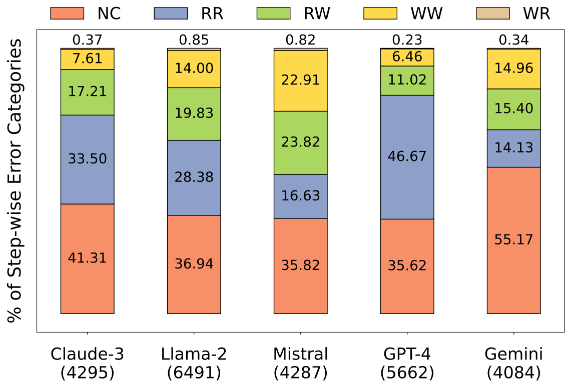

Figure 4: The percentage distribution of the broad error categories across the combined reasoning steps of all five LLMs. The total number of steps generated by each model is provided inside the round brackets below the model names.

5 Results and Analysis

5.1 Objective Evaluation

To evaluate the performance of LLMs when solving grid-based puzzles, we assess the outputs of 5 LLMs using the accuracy and *PuzzleEval*. As shown in Figure 3, we found that all the models have low performance on the *GridPuzzle* dataset in terms of accuracy. The smaller open-source LLMs completely failed at the puzzle-solving task, with Llama-2 solving only one puzzle correctly. Close-source models with significantly larger parameter sizes also exhibited poor performance. GPT-4 had the highest accuracy at only 5.11% (14 puzzles out of 274). Despite the overall low performance of all LLMs, the closed-source models perform marginally better. We evaluate the quality of the reasoning chains using *PuzzleEval*. Table 4 provides the ACS for each grid size available in the *GridPuzzle*. Surprisingly, compared to the accuracy, the performance of the models with *PuzzleEval* was significantly better as shown in Table 4. The ACS lie in the range of 0.26 to 0.64 across all grid sizes. This higher score can be attributed to the partial correctness of reasoning chains when solving the grid-puzzle task. The disparity between metrics shows that evaluating only final answers doesn’t fully capture LLMs’ effectiveness in complex logical tasks like grid puzzles.

With the increase in the sizes of the grids, the complexity of the puzzles also rises, leading to a depreciating performance by the LLMs with larger grids. Overall the performance of larger LLMs was much better than the small open-source models. Mistral-7B performed the worst in *PuzzleEval* which is in accordance with its low accuracy score. GPT-4 and Gemini models surprisingly have similar *PuzzleEval* scores (0.59 and 0.58 respectively) despite their large difference in accuracy. This difference in *PuzzleEval* could be attributed to the relatively shorter reasoning chains (fewer reasoning steps) produced by Gemini (an average of 14.91 steps) compared to GPT-4 (an average of 20.66 steps). Shorter reasoning chains may reduce the number of errors that occur while solving the puzzle. It is interesting to note that the smaller LLMs have consistently low performance with the increase in the grid size of the puzzles but the larger LLMs have mixed performance.

5.2 Reasoning Chain Evaluation

The relative distribution of the broad error categories over the collective reasoning steps for each model is given in Figure 4. It is important to note that, despite using the same zero-shot-CoT setting, the GPT-4 and Llama-2 used significantly more reasoning steps ($> 5.5k$ steps) to solve the 274 puzzles compared to the other three models ($\sim 4k$ steps). The distribution of error sub-categories for each model is presented as heatmaps in the first five

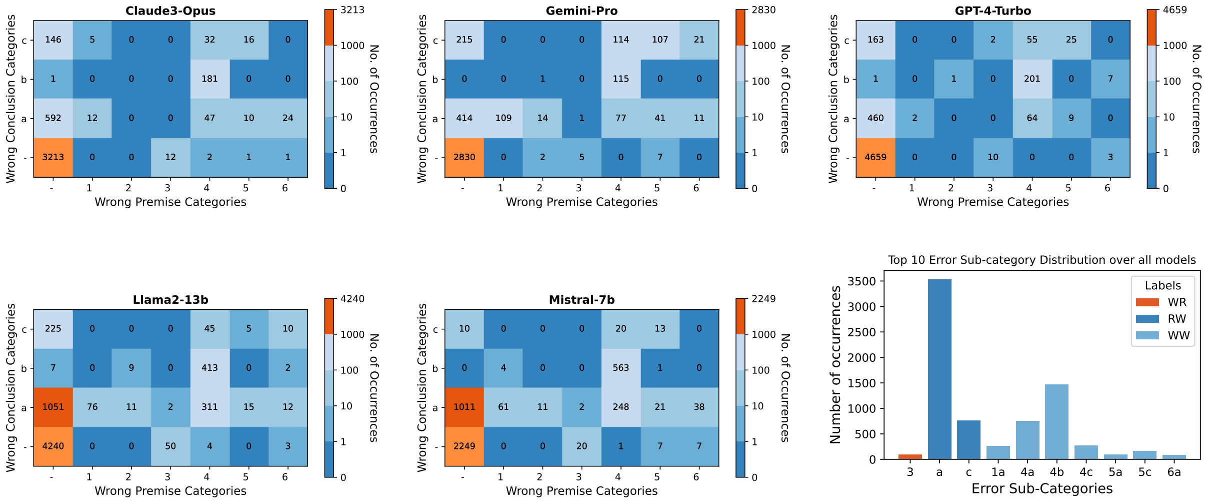

Figure 5: The first five sub-figures in the above section show the error Sub-category distribution over five LLMS. The last sub-figure denotes the top 10 error Sub category distribution across all model reasoning steps.

sub-figures in Figure 5. Here, we present several findings based on the evaluation of different error category distributions across *GridPuzzle*.

Majority of reasoning steps are error-free. Figure 4 shows that most reasoning steps for each model fall into the "NC" error category, indicating that many steps reiterate the facts or clues from the initial puzzle rather than focusing on reasoning. Over 55% of Gemini-Pro's reasoning steps fall into this category, the most among all models, suggesting that Gemini spends the fewest steps on actual reasoning. The "RR" category comprises over 46% of GPT-4's reasoning steps, highlighting its strong reasoning ability. This higher number of correct reasoning steps correlates with GPT-4's higher *PuzzleEval* score, reflecting its overall effectiveness.

The accuracy is low despite the reasoning chains being mostly error-free. The disparity between accuracy and *PuzzleEval* arises from the relative location of errors within the reasoning chains. It has been observed that "RR" category reasoning steps mainly occur in the initial half of the chain, leading to a high overall *PuzzleEval* score. Conversely, errors in the "RW", "WR", and "WW" categories typically occur in the latter half, resulting in incorrect final answers and lower accuracy scores. Based on error taxonomy, "RW", "WR", and "WW" broad error categories have been further dissected into 6×3 error sub-categories, with their distribution across reasoning steps shown in Figure 5.

Dominant broad categories: RW and WW. The most common error sub-category across all heatmaps appears to be the "-", the absence of errors. All the reasoning steps with "NC" and "RR" classifications fall in this category. To observe the actual overall trend across all 5 LLMs, the top 10 most common error sub-categories have been listed in the last sub-figure of Figure 5. The top categories 'a' and 'c' refer to the **Wrong Reasoning** and the **Wrong Elimination** sub-categories under the "RW" category. These errors arise when the premise is correct but the LLMs fail to make accurate deductions from it. A number of the top 10 sub-error categories ('1a', '4a', '4b', '4c', '5a', '5c', and '6a') emerge from the "WW" category.

For the categories, '4a', '4b', and, '4c' the errors in the premise are propagated from errors in previous reasoning steps showing how initially occurring errors in the chain can lead to more dependent errors. The '4b' error category is the one where this behavior is maximized as here both the premise and conclusions were wrong because of previously propagated errors. The '5a' and '5c' errors occurred due to the incompleteness or lack of information in the premise and wrong reasoning or elimination in the corresponding conclusions. The '1a' kind of error occurred when the premise consisted of hallucinated information. The only sub-category from the "WR" category making it in the top 10 is the '3' category which is caused due to wrong assumptions in the premise. It can be noted here that the reasoning steps of the "WR" category do not deteriorate either of the evaluation metrics, as the conclusions ended up being correct, but rather indicate the inconsistency of the LLMs in reasoning over puzzle-solving.

Mitigation Strategy	Accuracy	*PuzzleEval*
Baseline	12	0.61
Plan-and-Solve	9	0.62
Self-correct	10	0.59
Self-discover	**13**	**0.65**
Feedback-Learning	10	0.59
Program-of-Thought	10	-

Table 5: The results for accuracy and *PuzzleEval* using GPT-4-Turbo, with and without mitigation strategies for the 60 samples of 3×4 grid-size.

Proprietary LLMs are way better at GridPuzzle than Open-Source LLMs. From the results of objective and subjective metrics, it is evident that the open-source models have lower performance on the grid-puzzle-solving task than the proprietary models. The Llama-2 and Mistral models have the lowest accuracy values and their low performance on the *PuzzleEval* consistently degrades with the increase in the size and complexity of the grids. The Claude-3, Gemini, and GPT-4 models have higher values of accuracy but their performance across the grid sizes in the *PuzzleEval* is inconsistent. The disparity in the performance of both kinds of models can be attributed to the difference in their parameter sizes and the low instruction following capabilities of small open-source models.

Popular reasoning error mitigation strategies do not improve LLMs on GridPuzzle. We conduct a case study on a subset of *GridPuzzle*, focusing on a 3x4 grid size, utilizing commonly employed prompting techniques to enhance the reasoning capabilities of LLMs. In particular, we use five strategies: (1) Plan-and-Solve (Wang et al., 2023), (2) Self-correct (Zhang et al., 2024), (3) Self-discover (Zhou et al., 2024), (4) Feedback-Learning, and (5) Program of Thought prompting (Chen et al., 2023). We updated the prompts corresponding to these techniques to include some of our major findings from the reasoning chain evaluations and error categorization analysis as precautionary instructions.

The first strategy Plan-and-Solve prompts the model to first generate a plan to solve the given problem and then follow those steps. The second strategy is inspired by the Self-correct method which uses a combination of self-verification and self-refine to improve reasoning. Next, we used the Self-discover technique which is a 2-step structured reasoning process. We created our prompting technique called "Feedback-Learning" by providing specific feedback system instructions to the LLM based on our error taxonomy. Lastly, we also implemented a code-style prompting technique that implements a code to solve the puzzle but does not give a reasoning chain. The detailed prompt structure is described in Appendix E and the results of these strategies are in Table 5. It is evident from the results that prompting-based strategies are not sufficient to significantly improve the LLM reasoning on the grid-puzzle-solving task. Compared to the rest of the strategies, Self-Discover marginally improves the performance on both accuracy and *PuzzleEval*. These results indicate the need to develop techniques beyond prompting by having deeper insights from LLMs' reasoning chains.

6 Conclusion

In this work, we evaluated the logical reasoning abilities of LLMs through the lens of a grid-based puzzle-solving task. We introduced *GridPuzzle*, an evaluation dataset of 274 puzzles with various grid sizes. From a manual evaluation of reasoning chains generated by five different LLMs on *GridPuzzle*, we developed a fine-grained error taxonomy with five broad categories and nine subcategories. We then created an Auto-evaluator to automate the identification of error categories, providing broader insights into error distributions across the dataset. Additionally, we proposed *PuzzleEval*, a reference-free metric to objectively evaluate the correctness of reasoning chains for grid-based puzzles. Our analysis of error distributions in *GridPuzzle* revealed several interesting findings and insights into the logical reasoning abilities of different LLMs. We further evaluated existing reasoning-specific prompting methods, such as self-discover and self-correct, finding that they do not improve results on *GridPuzzle*. We believe our work offers a challenging dataset, highlights where these LLMs make mistakes, and provides insights to develop better logical reasoning systems for complex tasks such as grid puzzle-solving.

Limitations

While *GridPuzzle* facilitates the evaluation of LLMs' logical reasoning abilities, the complexity of the puzzles can be enhanced by incorporating further complex grid sizes beyond 4x6. Additionally, this study can be extended to different types of puzzles, such as Sudoku, Game of 24, and commonsense puzzles. Though our study provides fine-grained error categories, it can be further refined by mapping to formal logic to identify more detailed and atomic errors, offering a deeper understanding of LLMs' reasoning failures. Although we propose an effective automatic method for error identification to reduce manual analysis, exploring other automated methods using smaller-scale supervised learning could be a promising future research direction. We also note that this research is currently limited to the English language and can be extended to multilingual scenarios to evaluate the logical reasoning abilities of LLMs.

Ethics Statement

The dataset, GridPuzzle, used for this study is based on 274 puzzles from the open-source platform (more details in section 3.1). No personal information from data creators has been collected during the creation of the dataset. The data collection process strictly adheres to the terms of use and privacy policies of the platform. Furthermore, the use of proprietary LLMs such as GPT-4, Gemini, and Claude-3 in this study adheres to their policies of usage. We have used AI assistants (Grammarly and ChatGPT) to address the grammatical errors and rephrase the sentences.

Acknowledgement

We thank the anonymous reviewers for their constructive suggestions. We extend our gratitude to the Research Computing (RC), and Enterprise Technology at ASU for providing computing resources, and access to the ChatGPT enterprise version for experiments. We acknowledge support by a 2023 Spring Amazon Research Award (ARA). This material is also based upon work supported by the Engineering Research and Development Center - Information Technology Laboratory (ERDC-ITL) under Contract No. W912HZ24C0022.

References

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7358–7370, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mingkai Deng, Bowen Tan, Zhengzhong Liu, Eric Xing, and Zhiting Hu. 2021. Compression, transduction, and creation: A unified framework for evaluating natural language generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7580–7605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality.

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

Panagiotis Giadikiaroglou, Maria Lymperaiou, Giorgos Filandrianos, and Giorgos Stamou. 2024. Puzzle solving using reasoning of large language models: A survey. *arXiv preprint arXiv:2402.11291*.

Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2022. Roscoe: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Zhouhong Gu, Zihan Li, Lin Zhang, Zhuozhi Xiong, Sihang Jiang, Xiaoxuan Zhu, Shusen Wang, Zili Wang, Jianchen Wang, Haoning Ye, et al. 2023. Beyond the obvious: Evaluating the reasoning ability in real-life scenarios of language models on life scapes reasoning benchmark~(lsr-benchmark). *arXiv preprint arXiv:2307.05113*.

Akshat Gupta. 2023. Are chatgpt and gpt-4 good poker players?–a pre-flop analysis. *arXiv preprint arXiv:2308.12466*.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, et al. 2022. Folio: Natural language reasoning with first-order logic. *arXiv preprint arXiv:2209.00840*.

Shibo Hao, Yi Gu, Haotian Luo, Tianyang Liu, Xiyan Shao, Xinyuan Wang, Shuhua Xie, Haodi Ma, Adithya Samavedhi, Qiyue Gao, et al. 2024. Llm reasoners: New evaluation, library, and analysis of step-by-step reasoning with large language models. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Shulin Huang, Shirong Ma, Yinghui Li, Mengzuo Huang, Wuhe Zou, Weidong Zhang, and Hai-Tao Zheng. 2024. Lateval: An interactive llms evaluation benchmark with incomplete information from lateral thinking puzzles.

Adam Ishay, Zhun Yang, and Joohyung Lee. 2023. Leveraging large language models to generate answer set programs. In *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning*, pages 374–383.

Elgun Jabrayilzade and Selma Tekir. 2020. LGPSolver - solving logic grid puzzles automatically. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1118–1123, Online. Association for Computational Linguistics.

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

Mehran Kazemi, Quan Yuan, Deepti Bhatia, Najoung Kim, Xin Xu, Vaiva Imbrasaite, and Deepak Ramachandran. 2024. Boardgameqa: A dataset for natural language reasoning with contradictory information. *Advances in Neural Information Processing Systems*, 36.

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

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. RiddleSense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1504–1515, Online. Association for Computational Linguistics.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2021. Logiqa: a challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3622–3628.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Arindam Mitra and Chitta Baral. 2015. Learning to automatically solve logic grid puzzles. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1023–1033, Lisbon, Portugal. Association for Computational Linguistics.

David A. Noever and Ryerson Burdick. 2021. Puzzle solving without search or human knowledge: An unnatural language approach. *ArXiv*, abs/2109.02797.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. *In the proceeding of the Association for Computational Linguistics (ACL) 2024*.

Nisarg Patel, Mohith Kulkarni, Mihir Parmar, Aashna Budhiraja, Mutsumi Nakamura, Neeraj Varshney, and Chitta Baral. 2024. Multi-logieval: Towards evaluating multi-step logical reasoning ability of large language models. *arXiv preprint arXiv:2406.17169*.

Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. 2023. ReCEval: Evaluating reasoning chains via correctness and informativeness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10066–10086, Singapore. Association for Computational Linguistics.

Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations*.

Tal Schuster, Ashwin Kalyan, Oleksandr Polozov, and Adam Tauman Kalai. 2021. Programming puzzles. *arXiv preprint arXiv:2106.05784*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Gladys Tyen, Hassan Mansoor, Peter Chen, Tony Mak, and Victor Cărbune. 2023. Llms cannot find reasoning errors, but can correct them! *arXiv preprint arXiv:2311.08516*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Annual Meeting of the Association for Computational Linguistics*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Jaekyeom Kim, Moontae Lee, Honglak Lee, and Lu Wang. 2024. Small language models need strong verifiers to self-correct reasoning.

Jingmiao Zhao and Carolyn Jane Anderson. 2023. Solving and generating npr sunday puzzles with large language models. *arXiv preprint arXiv:2306.12255*.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V. Le, Ed H. Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. 2024. Self-discover: Large language models self-compose reasoning structures.

A *GridPuzzle* Dataset - Sample Puzzle

The *GridPuzzle* dataset contains 274 puzzles of various grid sizes and complexity. A sample puzzle from the dataset along with the Zero-shot-CoT prompt is described in Figure 6. All the puzzles in the dataset have a similar structure with varying numbers of clues.

Puzzle

Jamie never misses an opportunity to watch the night sky for falling stars. Using only the clues below, match the shooting stars to the options from dates and locations. Remember, as with all grid-based logic puzzles, no option in any category will ever be used more than once.

shooting stars: 2, 9, 16, 23.

dates: April 7, April 10, April 17, April 22.

locations: Eastbrook, Gilmore City, Isleton, Manchester.

Clues

1. The outing to Eastbrook logged 7 fewer shooting stars than the trip to Isleton.
2. The April 17 outing was in Gilmore City.
3. The outing where they saw 16 shooting stars was in Gilmore City.
4. The April 10 outing logged 7 more shooting stars than the April 22 outing.

Attached Instruction Prompt

While answering use the following format:
Step-by-step solution: [Your steps show how you are solving the puzzle]
Final Answer:
Fill the following table to show your final answer.
2 | correct option from dates | correct option from locations
9 | correct option from dates | correct option from locations
16 | correct option from dates | correct option from locations
23 | correct option from dates | correct option from locations

Gold Answer

2 | April 22 | Eastbrook
9 | April 10 | Isleton
16 | April 17 | Gilmore City
23 | April 7 | Manchester

Figure 6: The prompt structure of a 4 x 4 grid size puzzle from *GridPuzzle* dataset. Every Zero-shot-CoT prompt from the dataset consists of Puzzle, its corresponding Clues, the Instruction for solving the puzzle, along with the Gold solution of the Puzzle.

B Auto Evaluator: GPT-4o

To expand the reasoning chain evaluation process we prompt the GPT-4o model with a detailed system prompt. The structure of this system prompt is elaborated in Figure 7. The 3 main components of this system prompt are the **Instruction** - similar to the ones given to human evaluators, the **Knowledge** - obtained from the error taxonomy, and an **Exemplar** - consisting of a Puzzle, its Gold Solution, the LLM-generated Reasoning chain, and the evaluated Reasoning Chain.

Instruction

Task Definition:
Analyze and evaluate the given reasoning chains into specific reasoning error categories.

Evaluation Instructions:
Analyze each sentence by dividing it into a premise and conclusion, evaluate errors based on the correct solution, and classify according to specific error categories and subcategories.

Formatting Instructions:
Follow a JSON format for the output, detailing each sentence with its premise, conclusion, explanation, error category, and subcategory.

Knowledge

Error Taxonomy Description:
Detailed description of each of the error categories and their subcategories.
Main Categories:
Wrong Premise Wrong Conclusion (WW), Wrong Premise Right Conclusion (WR), Right Premise Wrong Conclusion (RW), Right Premise Right Conclusion (RR), and No Conclusion (NC).
Subcategories:
Hallucination, Incomplete Information, Assumptions, Error Propagation, Incomplete Information, Wrong Assumption, Wrong reasoning, wrong elimination.

Evaluation Example

Puzzle:
Nicholas and Edith are having a small dinner party...
Categories:
vintages : 1984, 1988, 1992, 1996.
wines : Annata Branco, Bianca Flaux, Ece Suss, Vendemmia.
types : gewurztraminer, merlot, pinot noir, riesling.
Clues:
1. The Ece Suss was bottled sometime after the Annata Branco.
2. The Bianca Flaux was bottled 4 years before the Vendemmia.
3. ...

Gold Solution:
1984 | Annata Branco | gewurztraminer
1988 | Ece Suss | pinot noir
1992 | Bianca Flaux | merlot
1996 | Vendemmia | riesling

LLM Reasoning Chain:
1. From clue 6, we know that the 1984 bottle is a gewurztraminer. So,...

Human-evaluated reasoning chain:


```
[
  {
    "Sentence": "From clue 6, we know that the 1984 bottle is a gewurztraminer.",
    "Premise": "clue 6",
    "Conclusion": "We know that the 1984 bottle is a gewurztraminer.",
    "Explanation": "Reiteration of the clue.",
    "Error Category": "NC",
    "Sub Category": "-"
  },...
```


Figure 7: The prompt structure of auto-evaluator. The prompt consists of instructions to evaluate reasoning chains, Knowledge of the error category descriptions, and a human-evaluated reasoning chain example. This system prompt is followed by a user prompt with the puzzle, its gold solution, and the reasoning chain to be evaluated.

Prompt	Model Response
Puzzle: A group of friends has decided to try several different weight-loss diets and exercises to see who amongst them can lose the most weight in 3 months. Using only the clues below, match the pounds lost to the options from names and diets. Remember, as with all grid-based logic puzzles, no option in any category will ever be used more than once. **pounds lost :** 3, 5, 7, 9. **names :** Raymond, Stuart, Tom, Wesley. **diets :** caveman, dairy-free, gluten-free, vegan. **Clues:** 1. Stuart lost 4 more pounds than the person who used the vegan diet. 2. The dieter who used the gluten-free diet lost 2 more pounds than Raymond. 3. Stuart lost 7 lbs. 4. The dieter who lost 7 lbs is either the person who used the caveman diet or Raymond. 5. Wesley lost 5 lbs. While answering use the following format: **Step-by-step solution:** [Your steps show how you are solving the puzzle] **Final Answer:** Fill the following table to show your final answer. 3 | correct option from names | correct option from diets 5 | correct option from names | correct option from diets 7 | correct option from names | correct option from diets 9 | correct option from names | correct option from diets	**Step-by-step solution:** 1. From clue 3, we know Stuart lost 7 lbs. 2. From clue 1, since Stuart lost 7 lbs, the person on the vegan diet lost 3 lbs (7 - 4 = 3). 3. From clue 4, since Stuart lost 7 lbs, he is either on the caveman diet or he is Raymond. But we know Stuart's name, so he must be on the caveman diet. 4. From clue 2, the person on the gluten-free diet lost 2 more pounds than Raymond. Since we have the pounds lost as 3, 5, 7, and 9, and Stuart already lost 7 lbs, the only way for someone to lose 2 more pounds than another (excluding the 7 lbs already accounted for) is for the gluten-free dieter to have lost 9 lbs and Raymond to have lost 7 lbs. However, since Stuart lost 7 lbs and is not Raymond, this clue helps us deduce the sequence of weight loss but needs to be revisited for exact matching once we have more information. 5. From clue 5, Wesley lost 5 lbs. 6. Now, knowing Wesley lost 5 lbs, Stuart lost 7 lbs, and the vegan diet resulted in a 3 lbs loss, we can deduce that Tom, being the only person left, lost 9 lbs. 7. Since Tom lost 9 lbs and the gluten-free diet is associated with losing 2 more pounds than Raymond, and the only diet left for Wesley (who lost 5 lbs) is dairy-free, it fits that Tom is on the gluten-free diet. 8. This leaves the vegan diet for the person who lost 3 lbs, which hasn't been directly matched to a name yet. Since Stuart is on the caveman diet and lost 7 lbs, Wesley is left with dairy-free and lost 5 lbs, and Tom is on the gluten-free diet and lost 9 lbs, Raymond must be the one who lost 3 lbs and is on the vegan diet.
Gold Solution: 3 | Raymond | vegan 5 | Wesley | gluten-free 7 | Stuart | caveman 9 | Tom | dairy-free	**Final Answer:** 3 | Raymond | vegan 5 | Wesley | dairy-free 7 | Stuart | caveman 9 | Tom | gluten-free

Manually Evaluated Reasoning Steps	Error Categories
1. From clue 3, we know Stuart lost 7 lbs.	RR
2. From clue 1, since Stuart lost 7 lbs, the person on the vegan diet lost 3 lbs (7 - 4 = 3).	RR
From clue 4, since Stuart lost 7 lbs, he is either on the caveman diet or he is Raymond	RR
But we know Stuart's name, so he must be on the caveman diet.	RR
4. From clue 2, the person on the gluten-free diet lost 2 more pounds than Raymond.	RR
Since we have the pounds lost as 3, 5, 7, and 9, and Stuart already lost 7 lbs, the only way for someone to lose 2 more pounds than another (excluding the 7 lbs already accounted for) is for the gluten-free dieter to have lost 9 lbs and Raymond to have lost 7 lbs.	RW-C
However, since Stuart lost 7 lbs and is not Raymond, this clue helps us deduce the sequence of weight loss but needs to be revisited for exact matching once we have more information.	RR
5. From clue 5, Wesley lost 5 lbs.	RR
6. Now, knowing Wesley lost 5 lbs, Stuart lost 7 lbs, and the vegan diet resulted in a 3 lbs loss, we can deduce that Tom, being the only person left, lost 9 lbs.	WR-2
7. Since Tom lost 9 lbs and the gluten-free diet is associated with losing 2 more pounds than Raymond, and the only diet left for Wesley (who lost 5 lbs) is dairy-free, it fits that Tom is on the gluten-free diet.	RW-C
8. This leaves the vegan diet for the person who lost 3 lbs, which hasn't been directly matched to a name yet.	RR
Since Stuart is on the caveman diet and lost 7 lbs, Wesley is left with dairy-free and lost 5 lbs, and Tom is on the gluten-free diet and lost 9 lbs, Raymond must be the one who lost 3 lbs and is on the vegan diet.	WW-4B

Figure 8: The **top left** section of the figure consists of a 3x4 sample puzzle from the *GridPuzzle* dataset along with the Zero-shot-CoT prompt. Right below the prompt, we have the Gold solution for the corresponding puzzle. In the **top right** section of the figure, we have the Model-generated Reasoning chain to solve this puzzle along with the Final Answer. In this particular instance, the reasoning chain was generated by the GPT-4 model. In the **bottom half** of the figure, we have the manually evaluated reasoning chain, split into individual sentences. The corresponding error categories found in each step is given the **bottom right** column called ‘Error Categories.’

Prompt	Llama2-13B Response
Puzzle: Using only the clues below, match the days to the options from guests and rooms. Remember, as with all grid-based logic puzzles, no option in any category will ever be used more than once. **days :** 2, 3, 4, 5. **guests :** Owens, Parker, Underwood, Zamora. **rooms :** 114, 124, 314, 320. **Clues:** 1. Underwood is in room 114. 2. The guest in room 320 is staying 1 day fewer than Underwood. 3. Owens is staying for 3 days. 4. Parker is staying 1 day longer than the client in room 314. While answering use the following format: **Step-by-step solution:** Your steps showing how you are solving the puzzle **Final Answer:** Fill the following table to show your final answer. 2 | correct option from guests | correct option from rooms 3 | correct option from guests | correct option from rooms 4 | correct option from guests | correct option from rooms 5 | correct option from guests | correct option from rooms	Let's break down the clues and solve the puzzle step by step. Step 1: From clue 1, we know Underwood is in room 114. Since Underwood is staying for 2 days (as the days are 2, 3, 4, 5), we can conclude that Underwood is staying for 2 days. Step 2: From clue 2, we know the guest in room 320 is staying 1 day fewer than Underwood. Since Underwood is staying for 2 days, the guest in room 320 is staying for 1 day. This means the guest in room 320 is staying for day 2. Step 3: From clue 3, we know Owens is staying for 3 days. Since Owens is staying for 3 days, Owens must be staying for day 3. Step 4: From clue 4, we know Parker is staying 1 day longer than the client in room 314. Since the days are 2, 3, 4, 5, the client in room 314 must be staying for day 4. Parker is staying 1 day longer, so Parker is staying for day 5. Step 5: Now that we have the days for each guest, we can fill in the table: 2 | Zamora | 320 3 | Owens | (not mentioned) 4 | (not mentioned) | 314 5 | Parker | (not mentioned)
Gold Solution: 2 | Zamora | 124 3 | Owens | 314 4 | Parker | 320 5 | Underwood | 114	**Final Answer:** 2 | Zamora | 320 3 | Owens | 124 4 | Underwood | 314 5 | Parker | 114 Note: The rooms for days 3, 4, and 5 are not explicitly mentioned in the clues, but we can deduce them based on the information provided.

Auto-Evaluated Reasoning Steps	Error Categories	Sub Categories

{
"Sentence": "From clue 1, we know Underwood is in room 114.",
"Premise": "clue 1",
"Conclusion": "we know Underwood is in room 114.",
"Explanation": "Reiteration of the clue.",
"Error Category": "NC",
"Sub Category": "-"
}
	NC	-

{
"Sentence": "Since Underwood is staying for 2 days (as the days are 2, 3, 4, 5), we can conclude that Underwood is staying for 2 days.",
"Premise": "Underwood is in room 114.",
"Conclusion": "Underwood is staying for 2 days.",
"Explanation": "The conclusion is incorrect as it assumes Underwood is staying for 2 days without sufficient information.",
"Error Category": "RW",
"Sub Category": "a"
}
	RW	a

⋮	⋮	⋮

{
"Sentence": "Parker is staying 1 day longer, so Parker is staying for day 5.",
"Premise": "the client in room 314 must be staying for day 4.",
"Conclusion": "Parker is staying for day 5.",
"Explanation": "The conclusion is incorrect as it is based on the wrong premise that the client in room 314 is staying for day 4.",
"Error Category": "WW",
"Sub Category": "4b"
}
	WW	4b

Figure 9: The **top left** section of the figure consists of a 3x4 sample puzzle from the *GridPuzzle* dataset along with the Zero-shot-CoT prompt. Right below the prompt, we have the Gold solution for the corresponding puzzle. In the **top right** section of the figure, we have the Model-generated Reasoning chain to solve this puzzle along with the Final Answer. In this instance, the reasoning chain was generated by the Llama2-13b model. In the **bottom half** of the figure, we have the GPT-4o Auto-Evaluated Reasoning chain.The auto-evaluation is done sentence-wise and the output is in a JSON-structured format consisting of 5 components: the *Sentence*, the *Premise*, the *Conclusion*, the *Error category* and the *Sub-category*. The corresponding error categories found in each sentence are given in the **bottom right** columns called ‘Error Categories’ and ‘Sub Categories.’

C Evaluation of Reasoning Chains

In order to identify the error categories from the erroneous reasoning chains we conducted manual and auto-evaluation of the reasoning chains. The process of manual evaluation has been described in figure 8 and the process of auto-evaluation using GPT-4o has been described in Figure 9.

D Annotation Guideline

To conduct the manual analysis of the reasoning chains, the annotators were provided the guidelines described in figure 10. The same guideline was also used to create the system prompt for the GPT-4o Auto-evaluator. The annotation process was conducted by 5 annotators and the annotations were also cross-examined to resolve any discrepancies.

12-step Annotation Guideline

1. Take your time to refer to the annotation example provided.
2. Make sure to break every step into individual sentences before doing the annotations and categorizations.
3. Analyze every sentence separated by a full stop and divide it into premise and conclusion (the conclusion is the derived information). If a sentence starts with words like "so" or "therefore", the premise will be the previous conclusion.
4. Take your time to refer to the error categories described in the Error Categories knowledge section below and Classify the derived information into specific error categories.
5. When classifying into the main Error Categories, the premise and conclusion correctness are determined independently from each other and it is only based on the answer given in the correct solution.
6. Compare the premise and conclusions with the correct solution first to determine if they are wrong or right.
7. Do not use the sub-category explanations to justify your main category (WW, WR, RR, RW, NC) classifications. Only after the error categories have been assigned you will look into the sub categories.
8. The premise is declared wrong only when it violates the data given in the correct solution. The conclusion is declared wrong only when it violates the data given in the correct solution.
9. If the category has a wrong premise, classify it further into a subcategory for the wrong premise.
10. If the category has a wrong conclusion, classify it further into a subcategory for the wrong conclusion.
11. Check your work and see if the chosen category is indeed the most appropriate category.
12. You are not required to solve the entire puzzle or correct any mistakes. You can take the help from the correct solution to conduct your analysis.

Figure 10: The 12-step guideline provided to the annotators to conduct manual evaluation of the reasoning chains.

E Mitigation Strategy Prompts

We conducted a study on the 60, 3x4 puzzles present in *GridPuzzle* dataset to try and improve the reasoning abilities of LLMs when solving the grid-puzzle task. We used prompt-based methods, such as the Plan-and-Solve technique, which divides puzzle-solving into planning and solving steps. We also enhanced the solver with insights from our error taxonomy. The prompt structure for this technique is given in figure 11.

First Prompt

System Prompt:
You are a grid-puzzle solving assistant.

User Prompt:
Let's first understand the problem and devise a plan to solve the problem.

<Puzzle>

Plan:

Second Prompt

System Prompt:
Make sure you do not make these errors while solving the problem:

<Instructions based on Error Taxonomy>

User Prompt:
Let's carry out the plan and solve the problem step-by-step.

Plan:

<Response from First Prompt>

<Puzzle>

Solution:

Figure 11: The prompt structure for the Plan-and-Solve strategy which is split into two prompts one for planning and the other for solving the puzzle.

Next, we devised our own strategy to improve LLM reasoning by using the top error categories from our findings and teaching the LLM to rectify those mistakes. This strategy termed as Feedback-learning makes use of a detailed system prompt that acts as a feedback-providing unit followed by a basic user prompt to solve the puzzle. The prompt structure for this strategy is shown in figure 12.

We also implemented a code-based technique to sole GridPuzzle. We borrowed the PoT prompt from the original implementation to create a solver function to solve the puzzles. Next, we asked an LLM to implement this code and produce the Final Answer. Since the codes produced by the LLM may contain some errors we utilized the LLM's compiler to implement the code instead of a rigid Python compiler. The prompt structure is provided in figure 13. Next is the Self-correct strategy which merges Self-verify and Self-refine qualities to minimize LLM reasoning errors. It starts with solving

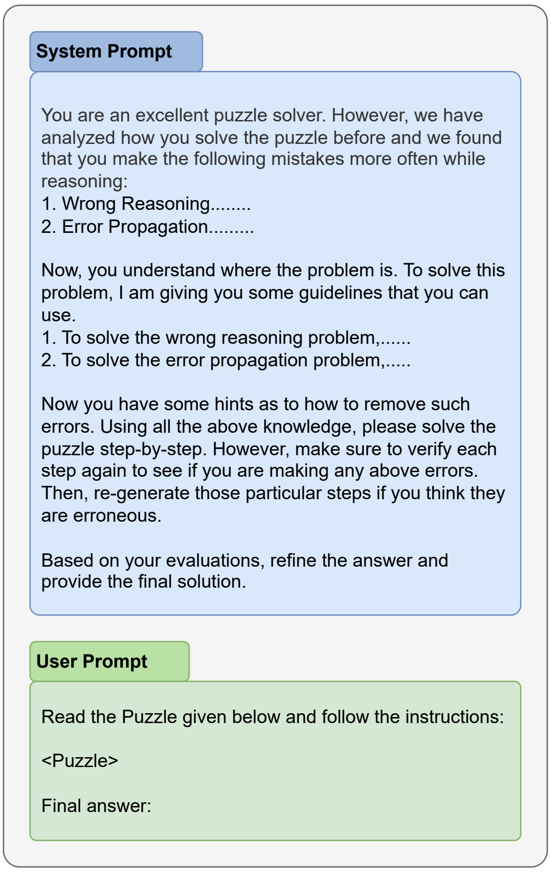

Figure 12: The prompt structure for the Feedback-learning strategy. The system prompts consist of instructions regarding the major errors as well as ways to rectify those errors.

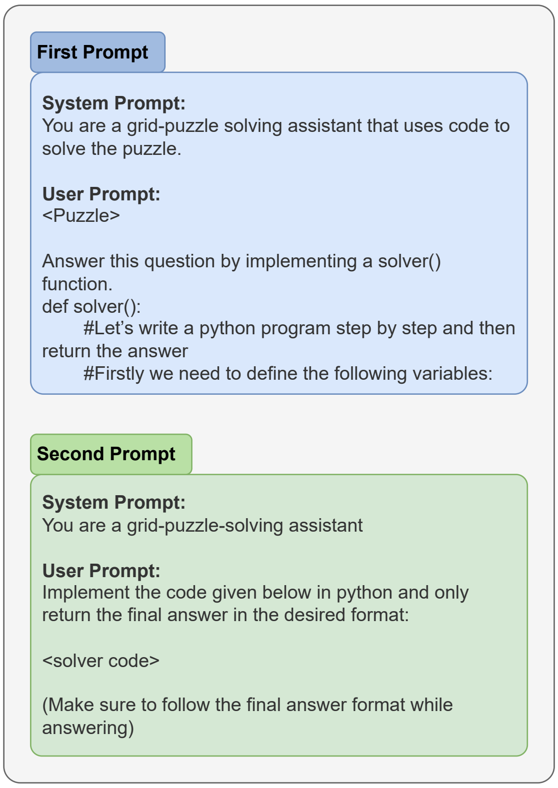

Figure 13: The prompt structure for the Program of Thought prompting technique. In the first part, we asked the LLM to generate a code to solve the given puzzle. In the second part, an LLM is prompted to implement the code produced in the first part to get the Final Answer table.

the puzzle using a Zero-shot-CoT prompt, followed by prompting the LLM to verify and refine the solution. Finally, it integrates the model's suggestions with insights from our error taxonomy to enhance the puzzle-solving response. The prompt structure for this strategy is shown in figure 14. Lastly, the Self-Discover strategy, depicted in figure 15, proved most effective in reducing LLM reasoning errors in puzzle-solving. This approach begins by having the model analyze the problem and potential errors, follows with a list of prescribed reasoning modules, prompts the LLM to select and apply the most suitable module, and concludes by using a structured prompt to solve the puzzle.

F Model Scaling Effect: Llama-70B

We conducted a case study on the Llama models to analyze their performance on GridPuzzle with increasing model parameter size. We repeated the same experiment in the Zeo-shot-CoT setting with the Llama-70B model. We found that the performance of the bigger model was marginally higher than the 13B model. The Accuracy went up from 1 correct final answer in the 13B model to 2 in the 70B model. The scores on PuzzleEval also went up 11% on average. However, despite the slight improvement, the Llama model's performance was still inferior to GPT-4, Gemini, and Claude. The experimental findings are presented in Table 6. We infer that even with the increasing model parameter size, the LLMs lack the intrinsic reasoning capabilities required to solve complex logic problems such as *GridPuzzle*.

Model	3 x 4	3 x 5	4 x 4	4 x 5	4 x 6	Avg
Llama-70B	0.51	0.51	0.52	0.58	0.42	**0.52**

Table 6: The results for *PuzzleEval* on the different grid sizes available in *GridPuzzle* dataset in terms of ACS for Llama-70B. The Accuracy of Llama-70B was 2/274 puzzles.

First Prompt

System Prompt:
You are a grid-puzzle solving assistant.

User Prompt:
<Zero-shot-CoT Puzzle Prompt>

Final Answer:

Second Prompt

System Prompt:
You are a grid-puzzle-solving assistant

User Prompt:
Let's verify the solution of the given puzzle below and suggest some corrections:

<Puzzle>

Solution:

<Previous Response>

Suggestion:

Third Prompt

System Prompt:
You are a grid-puzzle-solving assistant

User Prompt:
Now let's use the suggestions to solve the puzzle again:

<Previous Suggestion Response>

<Instructions based on Error Taxonomy>

<Puzzle>

Solution:

Figure 14: The prompt structure for the Self-Correct strategy is split into 3 parts. The first prompt solves the puzzle, the second prompt verifies the solution and gives suggestions to improve the solution, and the third prompt uses these suggestions along with error taxonomy-based instructions to refine the final solution.

First Prompt

Read the Puzzle given below and follow the instructions:
Following are the mistakes that you should be careful of before solving the puzzle:
<Instructions based on Error Taxonomy>

<Puzzle>

Before solving the puzzle, Select several reasoning modules that are crucial to utilize in order to solve the given task:
<Self-Discover Reasoning Modules>

Second Prompt

For the given puzzle:
<Puzzle>

Rephrase and specify each reasoning module so it better helps solve the task:
<Previous Response>

Third Prompt

For the given puzzle:
<Puzzle>

Operationalize the reasoning modules into a step-by-step reasoning plan in JSON format:
<Previous Response>

Fourth Prompt

For the given puzzle:
<Puzzle>

Follow the step-by-step reasoning plan in JSON to correctly solve the task. Fill in the values following the keys by reasoning specifically about the task. Do not simply rephrase the keys:
<Previous Response>

Figure 15: The prompt structure for the Self-Discover strategy. In the first part of this prompt the model is prompted to assess the problem and select the appropriate reasoning module to solve it. Then the module is modified to give a structured plan to solve the puzzle. In the second part, the model uses this structured plan along with instructions from our error taxonomy to solve the puzzle.