Exploring Deductive and Inductive Reasoning Capabilities of Large Language Models in Procedural Planning

Jiabao Kang[1✉], **Xinye Li**[1], **Liyan Xu**[2], **Qinbin Liu**[2], **Xi Chen**[2],
Zhiying Tu[1], **Dianhui Chu**[1], **Dianbo Sui**[1✉*]
[1]Harbin Institute of Technology [2]Tencent
23B903065@stu.hit.edu.cn, suidianbo@hit.edu.cn

Abstract

Deductive and inductive reasoning are fundamental components of human cognition, and in daily life, people often apply these types of reasoning unconsciously. While previous studies have extensively examined the deductive and inductive reasoning abilities of Large Language Models (LLMs) in rule-based and math-related tasks, little attention has been given to their role in procedural planning——an area that holds considerable relevance for real-world applications. To fill this gap, we present DIRPP (Deductive and Inductive Reasoning in Procedural Planning) in this paper, a benchmark designed to assess the deductive and inductive reasoning abilities of various LLMs within the context of procedural planning. Based on the benchmark, we initially observe that LLMs demonstrate excellent deductive reasoning capabilities in procedural planning but show suboptimal performance in inductive reasoning. To enhance their inductive reasoning abilities, we further propose a novel and effective method called IMSE (Induction through Multiple Similar Examples), which enables LLMs to generate multiple similar procedural plans and then perform inductive reasoning based on these examples. Through various experiments, we find that the proposed method can significantly improve the inductive reasoning capabilities of LLMs.[1]

1 Introduction

In recent years, advances in Large Language Models (LLMs), such as GPT-4 (OpenAI, 2024) and DeepSeek (DeepSeek-AI et al., 2024), have completely revolutionized the field of natural language processing. LLMs perform well on a wide variety of reasoning tasks (Lanham et al., 2023; Yao et al., 2023), including logical reasoning tasks (Pan et al., 2023; Lam et al., 2024).

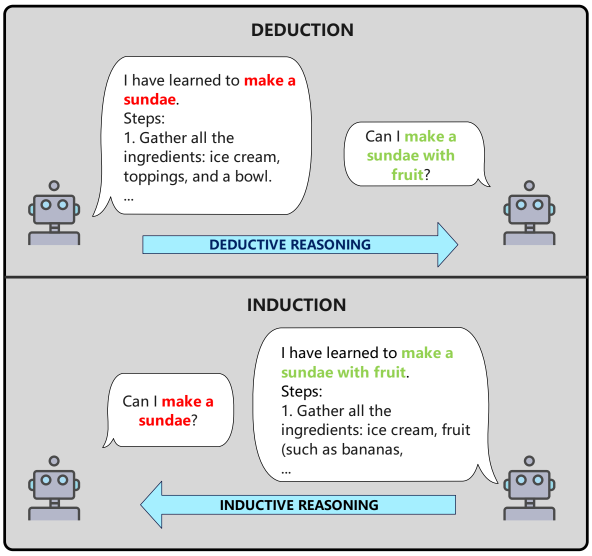
DEDUCTION
I have learned to make a sundae.
Steps:
1. Gather all the ingredients: ice cream, toppings, and a bowl.
...
Can I make a sundae with fruit?
DEDUCTIVE REASONING
INDUCTION
Can I make a sundae?
I have learned to make a sundae with fruit.
Steps:
1. Gather all the ingredients: ice cream, fruit (such as bananas,
...
INDUCTIVE REASONING

Figure 1: An example of inductive and deductive reasoning in procedural planning.

Deductive reasoning and inductive reasoning are the basic components of logical reasoning. People in daily life always use these two types of reasoning unconsciously. Deductive reasoning involves drawing specific conclusions from general principles under certain conditions. In contrast, inductive reasoning moves in the opposite direction. Inferences from the observed to the unobserved, or to general laws, are known as inductive inferences (Henderson, 2024). Deductive reasoning and inductive reasoning are considered crucial for achieving artificial intelligence (Lake et al., 2017; Chollet, 2019). Some research (Xu et al., 2024; Shao et al., 2024; Cheng et al., 2024;) has suggested that mixing deductive and inductive reasoning is not conducive to effective analysis. As a result, they have studied these two types of reasoning separately. For example, Xu et al. (2024) synthesizes 15 typical reasoning datasets and evaluates a wide variety of LLMs across inductive, deductive, abductive, and mixed-form reasoning settings. Shao et al. (2024)

[*] Dianbo Sui and Dianhui Chu are the corresponding authors.

[1] Resources of this paper can be found at https://github.com/KangJiabaohit/DIRPP.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 16320–16341
November 4-9, 2025 ©2025 Association for Computational Linguistics

examines the inductive and deductive capabilities of LLMs in the context of programming. Cheng et al. (2024) separates inductive and deductive reasoning to investigate which one is more important for the reasoning ability of LLMs.

It is worth noting that much of the recent work (Seals and Shalin, 2024; Sun et al., 2024; Mitchell et al., 2023; Mirchandani et al., 2023) on inductive and deductive reasoning abilities of LLMs is confined to rule-based or mathematically oriented tasks, as these tasks facilitate the separation of inductive and deductive reasoning, enabling more focused studies. However, exploring and probing the inductive and deductive reasoning abilities of LLMs in procedural planning—a field closely tied to real-life applications (Lu et al., 2022; Huang et al., 2022; Ahn et al., 2022; Zhao et al., 2023)—has received relatively little attention.

Procedural planning (Schank and Abelson, 1975; Pearson and Laird, 2005) entails breaking down a high-level goal into a series of coherent, logical, and goal-directed steps (*e.g., "Taking a shower"* → *"1. Prepare the bathroom; 2. Set the water temperature; 3. Undress; ..."*). It represents a form of structured general knowledge commonly used in daily life, with significant implications for both smarter AI systems and executable robotic systems (Kovalchuk et al., 2021; Huang et al., 2022). It is important to note that both inductive and deductive reasoning play a crucial role in enhancing the effectiveness of procedural planning. Specifically, inductive reasoning enables the system to generalize from observed patterns and past experiences (Heit, 2000; Hayes et al., 2010), allowing it to predict the most likely sequence of actions for new, unseen goals. This capability is vital for adapting to diverse tasks and improving planning efficiency. In contrast, deductive reasoning ensures the logical consistency and correctness of the planning process by enabling the system to deduce necessary steps based on predefined rules or knowledge (Johnson-Laird, 1999, 2008). This guarantees that the generated plans will achieve the specific goals without unnecessary steps or contradictions. Figure 1 illustrates an example that demonstrates both deductive and inductive reasoning in procedural planning.

In this paper, we explore the deductive and inductive capabilities of LLMs in procedural planning. To achieve this, we firstly propose a benchmark called DIRPP. Specifically, each example in DIRPP includes an abstract goal and an abstract procedural plan to achieve it, along with a specific goal and its corresponding specific procedural plan. Based on goals from CoScript (Yuan et al., 2023), we leverage GPT-4o-mini to complete the construction of our dataset. Next, we further introduce two metrics (the achievement rate and preference index) for DIRPP to quantitatively assess the performance of LLMs. Through pilot experimental results, we find that all LLMs demonstrate strong deductive abilities, while their inductive capabilities are comparatively weaker. To address this, we then propose a novel approach aimed at enhancing the inductive abilities of LLMs. Specifically, we first ask GPT-4o-mini to generate several related goals similar to the specific goal. Then, we instruct the evaluation model to generate procedural plans for these related goals. Finally, we enable the model to generalize from these multiple similar procedural plans, rather than relying on a single plan. Via various experiment, we find that our proposed method is effective.

To sum up, our contributions are as follows:

- To the best of our knowledge, this is the first study to investigate the deductive and inductive capabilities of LLMs in procedural planning.
- We propose a benchmark for evaluating the inductive and deductive reasoning abilities of LLMs.
- We introduce an effective method to enhance the inductive reasoning capabilities of LLMs in procedural planning.

2 Related Work

Deductive and Inductive Reasoning. Cognitive science holds that deductive and inductive reasoning are fundamental concepts for understanding human thought processes (Cai et al., 2024). In common cognitive models, these two types of reasoning are considered complementary: inductive reasoning generates hypotheses from observations, while deductive reasoning tests them (Wason, 1960). With LLMs making significant progress in a wide range of reasoning tasks (Bang et al., 2023; Bian et al., 2024; Imani et al., 2023), there has been growing interest in their underlying reasoning capabilities. Extensive research has focused on the logical reasoning abilities of LLMs. For example, Cai et al. (2024) simulate human thought processes by enabling LLMs to first summarize and then deduce, enhancing their reasoning abilities. Gendron et al. (2024) highlight that guiding models to follow causal reasoning paths improves their inductive reasoning capabilities. Yang et al. (2024) introduce a new task where natural language rules are hidden within facts, rather than explicitly provided to the models, to explore their inductive reasoning abilities. However, all the tasks explored in the above studies are rule-based or mathematically oriented, creating a gap between these studies and real-world applications. Therefore, we shift our focus to procedural planning tasks, which are more closely related to practical life.

Procedural Planning. Procedural planning is a goal-oriented type of script. A script is a structured knowledge that achieves a goal through a series of steps (Schank and Abelson, 1975). Procedural planning generation is a standard problem in nature language process (Chambers, 2017; Ostermann, 2020). Recent research has focused on leveraging LLMs for procedural planning generation (Sakaguchi et al., 2021; Sancheti and Rudinger, 2022), or on solving restricted procedural planning problems (Yuan et al., 2023; Brahman et al., 2024). Some studies also explore applying procedural planning to robots in real-world environments, with the goal of enabling them to perform specific actions (Huang et al., 2022; Wu et al., 2022; Guan et al., 2023). Unlike existing studies, this paper evaluates the deductive and inductive reasoning abilities of LLMs from the perspective of procedural planning, aiming to explore whether LLMs can replicate human cognitive abilities in real-world applications.

3 Task Definitions

In this section, we formalize the tasks of deductive and inductive reasoning in procedural planning to help clarify the subsequent content.

Procedural Planning. A procedural plan is a sequence of steps ($\mathcal{S} = \{s_1, s_2, \ldots, s_{|\mathcal{S}|}\}$, *e.g., "{ Gather ingredients, Preheat oven, ...}"*) designed to achieve a goal ($\mathcal{G}$) (Schank and Abelson, 1975; Yuan et al., 2023), *e.g.,"Make a cake"*. The procedural planning generation task is defined as $\mathcal{M} : \mathcal{G} \rightarrow \mathcal{S}$, where $\mathcal{M}$ represents a language model.

Deductive Reasoning in Procedural Planning. A deductive reasoning task involves applying general principles to derive results under specific conditions. In this paper, we refer to an abstract goal ($\mathcal{G}_a$) (*e.g., "Make a sundae"*) and an abstract procedural plan ($\mathcal{S} = \{s_1, s_2, \ldots, s_{|\mathcal{S}|}\}$) to achieve the abstract goal ($\mathcal{G}_a$) as a general principle (*i.e.*, $\mathcal{P} = \{\mathcal{G}_a;\ s_1, s_2, \ldots, s_{|\mathcal{S}|}\}$). A specific condition is represented by a more specific goal ($\mathcal{G}_s$) (*e.g., "Make a sundae with fruit"*). Suppose $\mathcal{S}^{'} = \{s^{'}_1, s^{'}_2, \ldots, s^{'}_{|\mathcal{S}^{'}|}\}$ is a specific procedural plan to achieve the specific goal. Thus, the deductive reasoning task in procedural planning can be defined as $\mathcal{M} : \{\mathcal{P}; \mathcal{G}_s\} \rightarrow \mathcal{S}^{'}$. We evaluate the generated result based on whether it achieves the specific goal. If $\mathcal{S}^{'}$ successfully achieves $\mathcal{G}_s$, the result is considered acceptable, and vice versa.

Inductive Reasoning in Procedural Planning. Inductive reasoning refers to inferences from the observed to the unobserved, or to general laws. In this paper, we use a specific goal ($\mathcal{G}_s$, *e.g., "Make a sundae with fruit"*) and a specific procedural plan ($\mathcal{S}^{'} = \{s^{'}_1, s^{'}_2, \ldots, s^{'}_{|\mathcal{S}^{'}|}\}$) to achieve the specific goal ($\mathcal{G}_s$) as an example observed (*i.e.*, $\mathcal{E} = \{\mathcal{G}_s;\ s^{'}_1, s^{'}_2, \ldots, s^{'}_{|\mathcal{S}^{'}|}\}$). An abstract goal ($\mathcal{G}_a$, *e.g., "Make a sundae ~~with fruit~~"*) is the object about which conclusions are drawn. Suppose $\mathcal{S} = \{s_1, s_2, \ldots, s_{|\mathcal{S}|}\}$ is an abstract procedural plan to achieve the abstract goal. So the inductive reasoning task can be defined as $\mathcal{M} : \{\mathcal{E}; \mathcal{G}_a\} \rightarrow \mathcal{S}$. In Appendix A, we further explain the rationale behind the inductive reasoning setup. Similarly, we can evaluate the generated result based on whether it achieves the abstract goal. However, this criterion has significant flaws. Even if the LLM does nothing but copy the specific procedural plan to achieve the specific goal, the result may still meet the abstract goal (*e.g., "A procedural plan for making a fruit sundae is also a procedural plan for making a sundae"*). Therefore, we further propose using the achievement of the specific goal as the evaluation criterion to determine whether the model is merely copying the example, since the abstract procedural plan that achieves the abstract goal often fails to achieve the specific goal.

4 Deductive and Inductive Reasoning in Procedural Planning

In this section, we present our complete benchmark. We begin by outlining the construction process of our dataset, followed by a detailed explanation of the metrics used for evaluating deductive and inductive reasoning tasks. Finally, we assess a range of LLMs, leveraging their few-shot in-context learning ability.

4.1 DIRPP Dataset

Each example in the dataset includes an abstract goal and an abstract procedural plan to achieve it, along with a specific goal and a specific procedural plan to achieve that goal. A representative example is shown in Appendix Table 19.

Dataset Construction. The dataset construction process consists of two main parts: defining the goals and generating the procedural plans to achieve them. For goal construction, we use the goals from CoScript (Yuan et al., 2023). Each example in CoScript includes an abstract goal and a specific goal, where abstract goals are sourced from wikiHow (Koupaee and Wang, 2018) and specific goals are generated by carefully crafting prompts and using InstructGPT (Ouyang et al., 2022) to obtain results. Once the goals (both abstract and specific) are established, we leverage the few-shot in-context learning ability of GPT-4o-mini to generate procedural plans for both abstract and specific goals. The prompt used in this process is shown in Appendix Table 9. After that, to ensure the quality of the generated dataset, we further conduct a manual evaluation of the generated procedural plans by randomly selecting 500 samples. Three volunteers are tasked with determining whether each generated procedural plan can successfully achieve its goal. The inter-rater agreement reaches Fleiss's $\kappa = 0.86$. Besides, the achievement rate for the abstract goal is 97.4%, while for the specific goal, it is 90.2%. These results demonstrate the reliability of the procedural planning generated by GPT-4o-mini. Besides, we compare the quality of data generated by DeepSeek-V3 and GPT-4o-mini in Appendix B.

Dataset Filtering. To perform the inductive reasoning task, we need to filter the dataset. As mentioned earlier, evaluating the achievement rate of abstract goals alone is insufficient, as the procedural plan that achieves the specific goal may also achieve the abstract goal. Therefore, if the abstract and specific goals are too similar (e.g., *"Making a sundea"* and *"Making a sundea with ice cream"*), the accuracy of evaluation is affected. To address this, we utilize GPT-4o-mini to determine whether abstract procedural plans in the dataset can achieve specific goals. If an abstract plan achieves a specific goal, it indicates that the abstract and specific goals are too close, and we discard the sample. The prompt used to instruct GPT-4o-mini for these judgments is shown in Appendix C.

Dataset Statistics We use the first 15,000 samples in CoScript as data sources to build our benchmark. After filtering out samples with abstract goals that overlapped with specific goals, we obtained a final dataset including 11,580 entries, with their goals covering a variety of categories, including hobbies, food, education, sports, and more.

4.2 Evaluation Metrics

For inductive and deductive reasoning tasks, we evaluate performance using automated metrics, including BLEU, ROUGE, and BERTScore, as set out in Brahman et al. (2024).

In addition, for the deductive reasoning task, we define the achievement rate of specific goals (AR_s) as a metric to evaluate the model's deductive reasoning capability. It is calculated as follows:

$$AR_s = \frac{AN_s}{N} \tag{1}$$

where AN_s denotes the number of generated procedural plans that successfully achieve specific goals, and N is the total number of tested examples.

Similarly, for the inductive reasoning task, we can use the achievement rate of abstract goals (AR_a) defined analogously to AR_s as a performance measure. However, this metric alone is insufficient because, in inductive reasoning, specific procedural plans can often achieve abstract goals without modification, leading to AR_a values close to 1 and thus rendering the metric less meaningful. To address this limitation, we additionally measure the achievement rate of specific goals (AR_s) for the generated procedural plans in the inductive reasoning task. We can assess the model's plagiarism using AR_s to determine whether the model is performing inductive reasoning or simply plagiarizing examples. Furthermore, to better evaluate the model's inductive reasoning ability, we introduce a preference index, which provides a more nuanced assessment of performance.

$$PI_a = \frac{PN_a}{N} \tag{2}$$

where PN_a represents the preferred number of inductively generated procedural plans compared to the abstracted procedural plans in the dataset, and N is the total number of tested samples. This indicator is specifically discussed in the context of inductive reasoning tasks and serves as a complement to the achievement rate of specific goals. The implication of this metric is to measure how much better the generated procedural plan is in the inductive reasoning task, relative to the data in the dataset.

Model	$\mathbf{AR_s}$ ↑	Model	$\mathbf{AR_s}$ ↑
Llama-3-8B	87.61	Mistral	86.83
OLMo-7B	86.51	OLMo-13B	88.98
Qwen2.5-7B	88.84	Qwen2.5-14B	90.47
Qwen2.5-32B	90.55	Claude-3	89.66
GPT-3.5-turbo	90.19	GPT-4o-mini	**91.08**

Table 1: The achievement rate of specific goals of each model in deductive reasoning (evaluated by GPT-4o-mini). Note that the data in the table are all percentages.

If the generated procedural plan is more inductive, logically consistent, applicable, and concise compared to the dataset sample, it can be inferred that the generated plan is preferred.

4.3 Pilot Experiments

In this section, we use the DIRPP dataset to evaluate the inductive and deductive reasoning capabilities of a variety of LLMs. These LLMs include both open-source models and closed-source models. Closed-source models include Claude-3 (claude-3-haiku-20240307), GPT-3.5-turbo (Brown et al., 2020), and GPT-4o-mini. Open-source models range in size from 7B to 32B parameters and include Llama-3-8B (Llama-3.1-8B-Instruct), Mistral (Mistral-7B-Instruct-v0.3), OLMo family (OLMo-2-1124-7B-Instruct, OLMo-2-1124-13B-Instruct), and Qwen family (Qwen2.5-7B-Instruct, Qwen2.5-14B-Instruct, Qwen2.5-32B-Instruct). We report results in terms of both automated evaluation and human evaluation. The prompts for conducting inductive and deductive reasoning are presented in Tables 12 and 13.

4.3.1 Automated Evaluation

Implementation Details. We leverage GPT-4o-mini's few-shot ability to train it to assess whether a generated procedural plan can achieve its goal. Additionally, through carefully designed prompts, GPT-4o-mini is tasked with making a preference decision between the generated procedural plan and the sample in the dataset. In this manner, we obtain the evaluation results provided by GPT-4o-mini. The prompt used is included in the Appendix D. The results are as follows.

Deductive Reasoning. Table 1 presents the achievement rate of specific goals across various models in the deductive reasoning task. Results for other metrics, such as ROUGE, BLEU, and BERTScore, are provided in the Appendix Table 20. It is not difficult to find that, among all models, GPT-4o-mini has the best performance, with an RA_s of 91.08%, and OLMO-7B has the worst performance, with an RA_s of 86.51%. Additionally, within models of the same family (OLMo family and Qwen family), performance improves as the number of parameters increases. In general, closed-source models outperform open-source models. Notably, the Qwen family models perform among the best for models with comparable parameter sizes, with Qwen2.5-32B's performance even approaching that of closed-source models. **In conclusion**, these results suggest that the performance of tested LLMs is sufficiently strong in the deductive reasoning task, indicating that the deductive reasoning abilities of LLMs in procedural planning are acceptable.

Inductive Reasoning. The achievement rate of abstract goals, the achievement rate of specific goals, and the preference index of inductive reasoning are presented in Table 2. ROUGE, BLEU and BERTScore automatic metrics are reported in the Appendix Table 21. First, as expected, for all models, their AR_a values are close to 100%. This suggests that, for the inductive reasoning task, a LLM's reasoning ability cannot be solely evaluated by the achievement rate of abstract goals, which contrasts with the evaluation approach used in the deductive reasoning task. Second, for the AR_s evaluation metric, GPT-3.5-turbo performs the best, with an AR_s value of 16.62%, while Qwen2.5-7B performs the worst, with an AR_s value of 45.34%. Other models exhibit AR_s values in between, with the smaller model Mistral attaining a relatively good AR_s value of 22.92%. Third, when examining the PI_a index, we find that Qwen2.5-32B achieves the highest PI_a value of 74.81%, while Mistral records the lowest PI_a value of 43.95%. The performance of other models lies between these two values. Finally, considering both AR_s and PI_a together, the model with the strongest inductive reasoning ability is Qwen2.5-32B, which boasts both the highest PI_a value and a strong AR_s. This is followed by several closed-source models, including Claude-3, GPT-3.5-turbo, and GPT-4o-mini. Conversely, models with fewer parameters, such as Llama-3-8B, Mistral, OLMo-7B, and Qwen2.5-7B, exhibit the weakest inductive reasoning abilities. These models either have the lowest AR_s or the lowest PI_a, with the other metric being slightly better. Overall, their inductive reasoning abilities are the weakest among the models com-

Model	$\mathbf{AR_a} \uparrow$	$\mathbf{AR_s} \downarrow$	$\mathbf{PI_a} \uparrow$
Llama-3-8B	97.36	38.92	44.33
Mistral	97.32	22.92	43.95
OLMo-7B	96.73	45.21	59.82
OLMo-13B	97.73	27.20	46.73
Qwen2.5-7B	96.85	45.34	53.78
Qwen2.5-14B	97.61	29.09	67.25
Qwen2.5-32B	97.98	19.14	**74.81**
Claude-3	97.48	25.44	70.15
GPT-3.5-turbo	**98.11**	**16.62**	65.37
GPT-4o-mini	97.48	24.18	70.28

Table 2: The achievement rate of abstract goals, the achievement rate of specific goals and the preference index of each model in inductive reasoning (evaluated by GPT-4o-mini).

pared. It is noteworthy that, despite the increase in parameters, the PI_a of OLMo-13B is lower than that of OLMo-7B, suggesting that OLMo-13B's inductive reasoning ability is also at a lower level. Nevertheless, even when considering the best AR_s and PI_a (16.62% and 74.81%, respectively) values across all models, the result indicates that the model's inductive reasoning ability remains a gap to the oracle. We also evaluate the performance of DeepSeek on a small subset of the dataset, as detailed in Appendix D.3. **In conclusion**, the results suggest that the inductive reasoning abilities of LLMs in procedural planning are suboptimal and still have room for improvement.

4.3.2 Human Evaluation

Implementation Details We randomly select 100 samples from the results generated by each model and recruit five additional volunteers to perform the labeling task. The labeling criteria are consistent with those used in the previous experiment. Specifically, the volunteers are provided with the same prompt and instructed to complete the annotations accordingly. The results of the manual evaluation are presented as follows.

Deductive Reasoning. Table 3 presents the achievement rate of specific goals as evaluated by human assessors. The results of human evaluations show some differences from those of GPT-4o-mini, though the overall discrepancy is minimal. This may be due to the small sample size. Moreover, even the lowest-performing model, Qwen2.5-7B, achieved an AR_s of 87.00%, while most models exceeded an AR_s of 90.00%. This further supports our previous argument that LLMs exhibit excellent deductive reasoning abilities in procedural planning.

Model	$\mathbf{AR_s} \uparrow$	**Model**	$\mathbf{AR_s} \uparrow$
Llama-3-8B	90.00	Mistral	93.00
OLMo-7B	88.00	OLMo-13B	91.00
Qwen2.5-7B	87.00	Qwen2.5-14B	90.00
Qwen2.5-32B	93.00	Claude-3	**94.00**
GPT-3.5-turbo	93.00	GPT-4o-mini	**94.00**

Table 3: The achievement rate of specific goals of each model in deductive reasoning (evaluated by humans).

Model	$\mathbf{AR_a} \uparrow$	$\mathbf{AR_s} \downarrow$	$\mathbf{PI_a} \uparrow$
Llama-3-8B	91.00	58.00	56.00
Mistral	92.00	47.00	57.00
OLMo-7B	92.00	73.00	63.00
OLMo-13B	94.00	54.00	58.00
Qwen2.5-7B	95.00	69.00	60.00
Qwen2.5-14B	96.00	51.00	67.00
Qwen2.5-32B	**98.00**	45.00	72.00
Claude-3	96.00	53.00	76.00
GPT-3.5-turbo	96.00	**41.00**	**78.00**
GPT-4o-mini	96.00	56.00	73.00

Table 4: The achievement rate of abstract goals, the achievement rate of specific goals and the preference index of each model in inductive reasoning (evaluated by humans).

Inductive Reasoning. Table 4 presents the results of human evaluation. The AR_a and PI_a of each model show some variation, though the changes are relatively minor. Specifically, the AR_a of the models decreased slightly, while their PI_a increased. Overall, the trends in these two metrics are align with those observed in GPT-4o-mini's evaluation. However, all models exhibit a substantial increase in AR_s. This may be due to humans being more sensitive to the finer details compared to GPT-4o-mini, allowing them to better assess whether a procedural plan can achieve a specific goal, resulting in a large increase in AR_s. Nevertheless, the human evaluation results also suggest that there is still substantial room for improvement in the model's inductive reasoning ability.

5 Induction through Multiple Similar Examples

Results in the pilot experiment show that LLMs' deductive reasoning abilities in procedural planning have reached an excellent level, while their

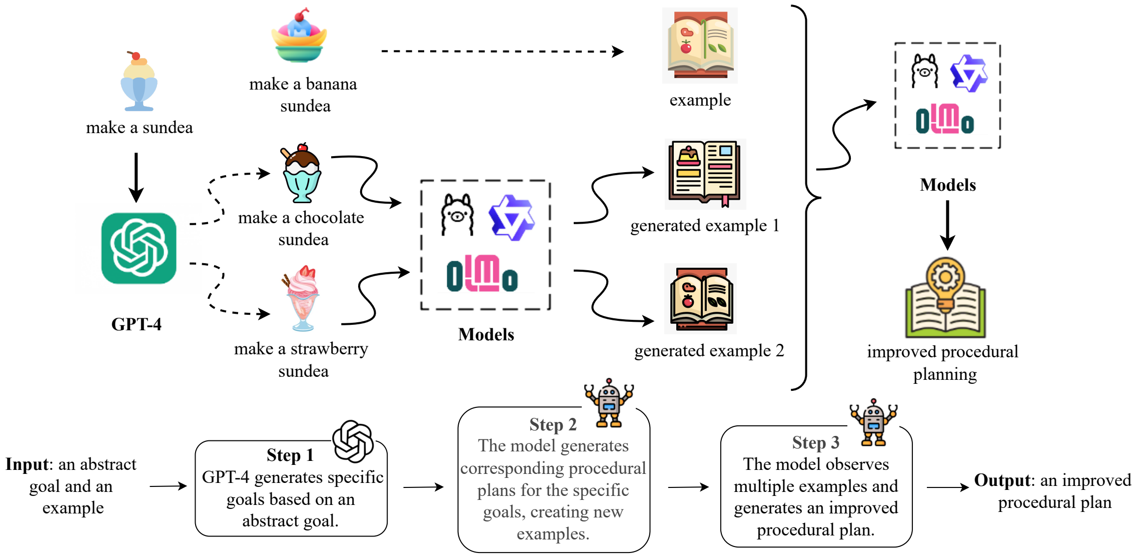
make a sundea
make a banana sundea
example
GPT-4
make a chocolate sundea
make a strawberry sundea
Models
generated example 1
generated example 2
Models
improved procedural planning
Input: an abstract goal and an example
Step 1 GPT-4 generates specific goals based on an abstract goal.
Step 2 The model generates corresponding procedural plans for the specific goals, creating new examples.
Step 3 The model observes multiple examples and generates an improved procedural plan.
Output: an improved procedural plan

Figure 2: Illustration of our proposed method, IMSE.

inductive reasoning abilities remain sub-optimal. In this section, we introduce a novel and effective approach to enhance the inductive reasoning capabilities of LLMs.

5.1 Methodology

In our inductive reasoning task setup, the model is asked to observe a single example (the specific goal and corresponding procedural plan) and use its internal knowledge to derive a general principle (the abstract procedural plan for achieving the abstract goal). This mirrors human learning, where individuals are taught to achieve a specific goal and then use their experience to formulate a procedural plan for an abstract goal. For example, after learning how to make a sundae with fruit, a person can easily summarize the general steps for making a sundae. OLMo et al. (2025) enhance model performance by generating multiple outputs and selecting the best ones. An immediate idea is to apply this method directly to enhance the model's inductive reasoning capability. However, due to the nature of the inductive reasoning task, we do not directly ask the model to generate multiple outputs. Instead, we could first ask the model to generate a variety of similar examples, and then have it summarize based on these examples.

Figure 2 illustrates the entire flow of our approach. To generate a variety of similar examples, we first need to obtain multiple other specific goals similar to the specific goal. Here, we use GPT-4o-mini's few-shot in-context learning ability to generate K^2 similar specific goals. Next, the model generates specific procedural plans for these goals, providing us with multiple similar examples. Finally, we follow the same process as in the inductive reasoning task, with the only difference being that the model observes multiple examples instead of just one. By doing so, the model can identify the common elements across examples and eliminate overly detailed aspects of each, resulting in a more refined abstract procedural plan for the abstract goal. The prompts used in each step are provided in the Appendix E.

5.2 Results

Our experimental setup follows the same procedure as described in Section 4.3. Meanwhile, we apply the ISME method and report the results from both automated and manual evaluations. In the Appendix F, we present the experimental results of allowing the model to improve itself.

5.2.1 Automated Evaluation

Table 5 presents the improved results (AR_a, AR_s, and PI_a). Results for other automatic metrics (ROUGE, BLEU, and BERTScore) are provided in the Appendix Table 22. First, for each improved model, the AR_a value, already close to 100% before the improvement, is further enhanced, with the proposed method resulting in an average increase of 1.15%. This demonstrates that observing multiple similar examples and generalizing their

[2] In the experiment, the value of K is set to 2.

Model	$\mathbf{AR_a}\uparrow$	$\mathbf{AR_s}\downarrow$	$\mathbf{PI_a}\uparrow$
Llama-3-8B	98.99	13.85	89.80
Mistral	98.11	13.22	89.04
OLMo-7B	97.86	12.59	94.58
OLMo-13B	98.74	11.59	88.91
Qwen2.5-7B	98.87	13.48	94.58
Qwen2.5-14B	**99.24**	11.71	95.47
Qwen2.5-32B	99.11	**9.44**	96.22
Claude-3	97.86	12.34	95.97
GPT-3.5-turbo	98.87	10.45	92.95
GPT-4o-mini	98.49	9.57	**96.98**

Table 5: The achievement rate of abstract goal, the achievement rate of specific goal and the preference degree of each improved model in inductive reasoning (evaluated by GPT-4o-mini).

Model	$\mathbf{AR_a}\uparrow$	$\mathbf{AR_s}\downarrow$	$\mathbf{PI_a}\uparrow$
Llama-3-8B	95.00	15.00	86.00
Mistral	96.00	17.00	90.00
OLMo-7B	96.00	14.00	92.00
OLMo-13B	97.00	12.00	86.00
Qwen2.5-7B	96.00	14.00	92.00
Qwen2.5-14B	96.00	13.00	95.00
Qwen2.5-32B	97.00	10.00	97.00
Claude-3	**99.00**	12.00	**98.00**
GPT-3.5-turbo	98.00	10.00	97.00
GPT-4o-mini	**99.00**	**9.00**	**98.00**

Table 6: The achievement rate of abstract goal, the achievement rate of specific goal and the preference degree of each improved model in inductive reasoning (evaluated by humans).

common features to produce abstract procedural plans helps better achieve the abstract goals. Second, after applying the proposed method, the AR_s value of each model is reduced to different degrees. The OLMo-7B model shows the largest decrease, from 45.21% to 12.59% (a reduction of 32.62%), followed by Qwen2.5-7B, which drops 31.86%, from 45.34% to 13.48%. The smallest decrease is observed in GPT-3.5-turbo, with a reduction of 6.17%, from 16.62% to 10.45%. After the improvement, Qwen2.5-32B achieves the best AR_s value of 9.44%, while Llama-3-8B records the largest AR_s value of 13.85%. Other models exhibit AR_s values between these two extremes. Notably, even Llama-3-8B, which has the largest AR_s value (13.85%), outperforms GPT-3.5-turbo, the best model before the improvement, which has an AR_s value of 16.62%. This demonstrates the effectiveness of our method. By inducting from multiple examples rather than relying on a single one, we effectively reduce the models' dependency on any specific example during induction, leading to a significant reduction in the AR_s value. Similar to the AR_s value, the PI_a value is also greatly improved, with varying degrees of improvement across models. After the improvement, all models, except Llama-3-8B, Mistral, and OLMo-13B, achieve PI_a values greater than 90.00%. GPT-4o-mini achieves the highest PI_a value of 96.98%, while OLMo-13B has the lowest, at 88.91%. However, before the improvement, the best PI_a value is only 74.81%. This indicates that the improved models generate more inductive, logically consistent, applicable, and concise abstract procedural plans in the inductive reasoning task.

5.2.2 Human Evaluation

The results of the human evaluation are summarized in Table 6. Overall, the results from manual evaluation are similar to those obtained from GPT-4o-mini evaluation. While the improved models show only minimal changes in AR_a values, with slight increases, both AR_s and PI_a values exhibit significant improvements. Specifically, Mistral achieves the highest AR_s value of 17.00%, while GPT-4o-mini shows the lowest at 9.00%. Prior to the improvement, GPT-3.5-turbo is the top performer, with an AR_s value of 41.00%. The proposed method effectively reduced the AR_s values. Regarding PI_a values, Llama-3-8B and OLMo-13B have the lowest scores, at 86.00%, while Claude-3 and GPT-4o-mini achieve the highest, with values of 98.00%. Before the improvement, even the best model, GPT-3.5-turbo, has a PI_a value of only 78.00%. These results further demonstrate the effectiveness and reliability of the proposed method.

5.3 Ablation Study

To demonstrate that our method effectively improves the models' reasoning abilities, rather than merely benefiting from the simpler nature of the few-shot setting, we conduct an ablation study. Specifically, we perform experiments using several representative models: Llama-3-8B, Qwen2.5-14B, Qwen2.5-32B, GPT-3.5-Turbo, and GPT-4o-mini. For simplicity, we select the first 500 examples from the dataset as our evaluation set.

Model	$\mathbf{AR_a} \uparrow$	$\mathbf{AR_s} \downarrow$	$\mathbf{PI_a} \uparrow$
Llama-3-8B	97.20	40.20	38.20
Qwen2.5-14B	97.20	29.00	73.00
Qwen2.5-32B	97.80	17.60	83.00
GPT-3.5-turbo	97.40	20.80	69.80
GPT-4o-mini	96.20	24.60	71.00

Table 7: The achievement rate of abstract goal, the achievement rate of specific goal and the preference degree of each improved model in few-shot inductive reasoning (evaluated by GPT-4o-mini).

In the ablation study, we disable the model's ability to generate similar examples. Instead, we randomly select two specific goals and their corresponding procedural plans from the remaining dataset to replace the similar examples generated by our proposed method. This creates a few-shot setting, but the added examples are not semantically related to the query instance presented to the model. We only present the results of the automated evaluation. The detailed results of this ablation study are provided in Table 7.

Compared to the results in Table 2, it is evident that simply increasing the number of examples does not lead to substantial improvements in AR_s and PI_a, nor does it enhance the models' inductive reasoning abilities. Only Qwen2.5-32B shows a slight performance improvement, but this gain is significantly smaller than the one achieved by our proposed method. The performance of other models remains largely unchanged. Notably, for Llama-3-8B, the added examples act as noise, causing a degradation in performance. This underscores that the effectiveness of our method lies not in the mere addition of examples, but in the crucial step of generating similar examples.

6 Conclusion

In this work, we introduce a benchmark, DIRPP, designed to explore deductive and inductive reasoning in procedural planning for LLMs. Our findings indicate that while LLMs demonstrate strong deductive reasoning capabilities, their inductive reasoning abilities requires improvement. To address this, we propose a novel and effective method, IMSE, which enables the model to generate multiple similar examples and generalize based on these examples, thereby enhancing its inductive reasoning capability. We hope that our work will inspire future research into reasoning within the context of procedural planning.

Limitations

Our research is generally logical and well-founded, but it is not without limitations. The main issues are as follows:

- Although we evaluate a variety of LLMs, due to constraints in computational resources, the largest open-source model included in our exploration is limited to 32B parameters. Models with larger parameter sizes are not considered in the evaluation, which limits the generalizability of our conclusions.

- While our proposed method, IMSE, effectively enhances the inductive reasoning capabilities of LLMs in procedural planning, it necessitates the generation of multiple similar examples. This results in a significant increase in the number of outputs and a corresponding rise in computational costs. Future work should focus on exploring more cost-effective strategies for improvement.

- In our experiments, we rely on GPT-4o-mini as the evaluator. However, since GPT-4o-mini's judgment may differ from that of human evaluators, this introduces the potential for biases, leading to discrepancies between our findings and those that might arise from human judgment. Moving forward, it will be important to either identify more reliable evaluators or improve the evaluation metrics to mitigate this issue.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *Preprint*, arXiv:2204.01691.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.

Ning Bian, Xianpei Han, Le Sun, Hongyu Lin, Yaojie Lu, Ben He, Shanshan Jiang, and Bin Dong. 2024. ChatGPT is a knowledgeable but inexperienced solver: An investigation of commonsense problem in large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3098–3110, Torino, Italia. ELRA and ICCL.

Faeze Brahman, Chandra Bhagavatula, Valentina Pyatkin, Jena D. Hwang, Xiang Lorraine Li, Hirona J. Arai, Soumya Sanyal, Keisuke Sakaguchi, Xiang Ren, and Yejin Choi. 2024. Plasma: Making small language models better procedural knowledge models for (counterfactual) planning. *Preprint*, arXiv:2305.19472.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, et al. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Chengkun Cai, Xu Zhao, Haoliang Liu, Zhongyu Jiang, Tianfang Zhang, Zongkai Wu, Jenq-Neng Hwang, and Lei Li. 2024. The role of deductive and inductive reasoning in large language models. *Preprint*, arXiv:2410.02892.

Nathanael Chambers. 2017. Behind the scenes of an evolving event cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 41–45, Valencia, Spain. Association for Computational Linguistics.

Kewei Cheng, Jingfeng Yang, Haoming Jiang, Zhengyang Wang, Binxuan Huang, Ruirui Li, Shiyang Li, Zheng Li, Yifan Gao, Xian Li, Bing Yin, and Yizhou Sun. 2024. Inductive or deductive? rethinking the fundamental reasoning abilities of llms. *Preprint*, arXiv:2408.00114.

François Chollet. 2019. On the measure of intelligence. *Preprint*, arXiv:1911.01547.

DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. *Preprint*, arXiv:2401.02954.

Gaël Gendron, Qiming Bao, Michael Witbrock, and Gillian Dobbie. 2024. Large language models are not strong abstract reasoners. *Preprint*, arXiv:2305.19555.

Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Leveraging pretrained large language models to construct and utilize world models for model-based task planning. *Preprint*, arXiv:2305.14909.

Brett K Hayes, Evan Heit, and Haruka Swendsen. 2010. Inductive reasoning. *Wiley interdisciplinary reviews: Cognitive science*, 1(2):278–292.

Evan Heit. 2000. Properties of inductive reasoning. *Psychonomic bulletin & review*, 7:569–592.

Leah Henderson. 2024. The Problem of Induction. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Winter 2024 edition. Metaphysics Research Lab, Stanford University.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9118–9147. PMLR.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. MathPrompter: Mathematical reasoning using large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 37–42, Toronto, Canada. Association for Computational Linguistics.

Philip N Johnson-Laird. 1999. Deductive reasoning. *Annual review of psychology*, 50(1):109–135.

Philip N Johnson-Laird. 2008. Mental models and deductive reasoning. *Reasoning: studies in human inference and its foundations*, pages 206–222.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *Preprint*, arXiv:1810.09305.

Alexander Kovalchuk, Shashank Shekhar, and Ronen I Brafman. 2021. Verifying plans and scripts for robotics tasks using performance level profiles. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31, pages 673–681.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2017. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253.

Long Hei Matthew Lam, Ramya Keerthy Thatikonda, and Ehsan Shareghi. 2024. A closer look at logical reasoning with llms: The choice of tool matters. *Preprint*, arXiv:2406.00284.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. NeuroLogic a*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 780–799, Seattle, United States. Association for Computational Linguistics.

Suvir Mirchandani, Fei Xia, Pete Florence, Brian Ichter, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, and Andy Zeng. 2023. Large language models as general pattern machines. *Preprint*, arXiv:2307.04721.

Melanie Mitchell, Alessandro B. Palmarini, and Arseny Moskvichev. 2023. Comparing humans, gpt-4, and gpt-4v on abstraction and reasoning tasks. *Preprint*, arXiv:2311.09247.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, et al. 2025. 2 olmo 2 furious. *Preprint*, arXiv:2501.00656.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Simon Ostermann. 2020. Script knowledge for natural language understanding.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, et al. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *Preprint*, arXiv:2305.12295.

Douglas J Pearson and John E Laird. 2005. Incremental learning of procedural planning knowledge in challenging environments. *Computational Intelligence*, 21(4):414–439.

Keisuke Sakaguchi, Chandra Bhagavatula, Ronan Le Bras, Niket Tandon, Peter Clark, and Yejin Choi. 2021. proScript: Partially ordered scripts generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2138–2149, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Abhilasha Sancheti and Rachel Rudinger. 2022. What do large language models learn about scripts? In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 1–11, Seattle, Washington. Association for Computational Linguistics.

Roger C. Schank and Robert P. Abelson. 1975. Scripts, plans and knowledge. In *International Joint Conference on Artificial Intelligence*.

Spencer M. Seals and Valerie L. Shalin. 2024. Evaluating the deductive competence of large language models. *Preprint*, arXiv:2309.05452.

Yunfan Shao, Linyang Li, Yichuan Ma, Peiji Li, Demin Song, Qinyuan Cheng, Shimin Li, Xiaonan Li, Pengyu Wang, Qipeng Guo, Hang Yan, Xipeng Qiu, Xuanjing Huang, and Dahua Lin. 2024. Case2code: Learning inductive reasoning with synthetic data. *Preprint*, arXiv:2407.12504.

Wangtao Sun, Haotian Xu, Xuanqing Yu, Pei Chen, Shizhu He, Jun Zhao, and Kang Liu. 2024. Itd: Large language models can teach themselves induction through deduction. *Preprint*, arXiv:2403.05789.

Peter C Wason. 1960. On the failure to eliminate hypotheses in a conceptual task. *Quarterly journal of experimental psychology*, 12(3):129–140.

Te-Lin Wu, Alex Spangher, Pegah Alipoormolabashi, Marjorie Freedman, Ralph Weischedel, and Nanyun Peng. 2022. Understanding multimodal procedural knowledge by sequencing multimodal instructional manuals. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4525–4542, Dublin, Ireland. Association for Computational Linguistics.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2024. Are large language models really good logical reasoners? a comprehensive evaluation and beyond. *Preprint*, arXiv:2306.09841.

Zonglin Yang, Li Dong, Xinya Du, Hao Cheng, Erik Cambria, Xiaodong Liu, Jianfeng Gao, and Furu Wei. 2024. Language models as inductive reasoners. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 209–225, St. Julian's, Malta. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Siyu Yuan, Jiangjie Chen, Ziquan Fu, Xuyang Ge, Soham Shah, Charles Jankowski, Yanghua Xiao, and Deqing Yang. 2023. Distilling script knowledge from large language models for constrained language planning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4303–4325, Toronto, Canada. Association for Computational Linguistics.

Zirui Zhao, Wee Sun Lee, and David Hsu. 2023. Large language models as commonsense knowledge for large-scale task planning. *Preprint*, arXiv:2305.14078.

A Inductive Reasoning

Inferences from the observed to the unobserved, or to general laws, are known as inductive inferences. In our experimental setup, the large language model is presented with a single example before being prompted to draw a conclusion. This setup is considered reasonable for the following reason. Imagine a person who has never made a sundae before, and we teach him how to make a banana sundae. When we then ask him to make a strawberry sundae, it becomes easy for him because he has implicitly inductively learned the general steps for making a sundae from the banana sundae process. He then deduces how to apply these steps to make a strawberry sundae. We are curious about whether large language models possess similar abilities in inductive and deductive reasoning, or how strong these reasoning abilities are. This is the inspiration behind our setup.

B Comparison of Data Quality

To further validate the quality of the data generated by GPT-4o-mini, we regenerate 500 samples using DeepSeek-V3 and use DeepSeek-V3 to evaluate both the previously constructed dataset and the newly generated data. The version of DeepSeek we used is DeepSeek-V3-0324. Experimental results indicate that the data generated by DeepSeek-V3 and GPT-4o-mini are of comparable quality, with only minor differences. Therefore, we chose the more cost-effective GPT-4o-mini to construct the dataset. The experimental results are shown in Table 8.

C Filtering Similar Examples

For the inductive reasoning task, the dataset is filtered to ensure the reliability of the evaluation results. The primary objective is to remove samples where the abstract goal and the specific goal are too similar. Specifically, we designed prompts to enable GPT-4o-mini to determine whether the abstract procedural plan achieves the specific goal. If the abstract procedural plan successfully achieves the specific goal, it indicates that the abstract and specific goals are too similar, and such samples are discarded. Table 14 shows an example of the prompt that we use to filter similar examples with GPT-4o-mini.

D Evaluation with GPT-4o-mini

D.1 Deductive Reasoning

In evaluating the deductive reasoning abilities of each model, we require GPT-4o-mini to assess whether the generated procedural plan can achieve its corresponding specific goal. We enable this capability in GPT-4o-mini through contextual learning. Table 15 provides a concrete example.

D.2 Inductive Reasoning

For the inductive reasoning task, we need to compute AR_s, AR_a, and PI_a for each model. Similarly, we enable GPT-4o-mini to acquire the ability to perform evaluations through its few-shot learning capability. Specifically, GPT-4o-mini needs to accomplish the following three tasks. First, GPT-4o-mini is required to assess whether the generated procedural plan can achieve the abstract goal. Second, GPT-4o-mini is used to determine whether the generated procedural plan can achieve the specific goal. Third, the generated procedural plan is compared with the abstract procedural plan in the dataset, and GPT-4o-mini is utilized to make a preference decision. Tables 16, 14, and 17 present the prompts used (the same prompt employed for data filtering is used when determining whether the generated procedural plan achieves the specific goal).

D.3 Performance of DeepSeek

We select the first 500 samples from the dataset and perform inductive reasoning tasks using DeepSeek, specifically the DeepSeek-V3-0324 version, which contains 671 billion parameters. The detailed experimental results are as follows: the AR_a value is 98%, the AR_s is 9.6%, and the PI_a is 89%. These results show that DeepSeek outperforms smaller-scale models, supporting the hypothesis that inductive reasoning capabilities improve as the model size increases. While DeepSeek's performance is promising, it also reveals notable gaps in inductive reasoning, as reflected by the non-negligible AR_s and PI_a values. Therefore, although larger models like DeepSeek exhibit enhanced inductive reasoning abilities, the core challenge identified in our research—that inductive reasoning in procedural planning remains a difficulty for LLMs—persists, albeit to a lesser extent in more advanced models.

Model	**AR$_a$ ↑**	**AR$_s$ ↑**
DeepSeek	**98.80**	**96.40**
GPT-4o-mini	97.80	93.40

Table 8: Achievement Rates: AR_a (Abstract Goal) and AR_s (Specific Goal) for generated plans (Evaluated by DeepSeek-V3).

Procedural Planning Generation

/*Task prompt*/

Please follow the example below to generate the output for me. Generate only output, do not repeat the question.

/*Examples*/

Goal 1: List the steps of baking a cake.
Steps:
{Specific Procedural Planning}
Goal 2: List the steps of borrowing a book from the library.
Steps:
{Specific Procedural Planning}
Goal 3: List the steps of taking a shower
Steps:
{Specific Procedural Planning}

/*Completion*/

Goal: List the steps of {Goal}
Steps: Generated Procedural Planning

Table 9: An example of prompt for GPT-4o-mini for procedural planning generation via in-context learning. Generated texts are highlighted. {Specific Procedural Planning} represents a procedural plan to achieve the corresponding goal. {Goal} will be replaced with specific content.

E Improvement of the Model

Initially, we train GPT-4o-mini to generate specific goals by leveraging its few-shot learning capability. To achieve this, we carefully design prompts, with an example provided in Table 18. Subsequently, we train the model to generate corresponding procedural plans based on these specific goals. At this stage, the prompt used is identical to that employed during the dataset construction phase, as shown in Table 19. Through this process, we obtain multiple similar examples. We then proceed similarly to inductive reasoning, with the key distinction being that the model is tasked with observing multiple examples, rather than a single one. Table 10 illustrates the prompt used, which enables the model to generate improved procedural plans.

/*Task Description*/

Please synthesize a unified and flexible script based on the following three scripts.

Abstract Goal: {Abstract Goal}
Script A: {Specific Script 1}
Script B: {Specific Script 2}
Script C: {Specific Script 3}

/*Requirements*/:

1.Create a clear, concise, and easy-to-follow script.
2.Retain the necessary steps and key points.
3.Ensure the script is flexible and applicable to various situations.

/*Completion*/

Please consolidate and optimize the scripts according to the above requirements, ensuring clarity, efficiency, and practicality. Output only the integrated script.
Generated Abstract Procedural Planning

Table 10: An example of prompt for improving the model. Generated texts are highlighted. {Abstract Goal}, {Specific Script 1}, {Specific Script 2}, and {Specific Script 3} will be replaced with specific content.

Model	**AR$_a$ ↑**	**AR$_s$ ↓**	**PI$_a$ ↑**
Mistral	97.70	15.40	81.70
OLMo-13B	98.50	13.20	86.80
Qwen2.5-32B	99.20	**8.10**	**98.00**
GPT-3.5-turbo	**99.30**	10.90	96.60

Table 11: The achievement rate of abstract goal, the achievement rate of specific goal and the preference degree of each self-improved model in inductive reasoning (evaluated by GPT-4o-mini).

F Self-improvement of the Model

In the original approach, we utilized GPT-4o-mini to generate specific goals. Here, we explore the experimental results of allowing the model to generate specific goals on its own. We selected four representative models (Mistral, OLMo-13B, Qwen2.5-32B, and GPT-3.5-Turbo) and randomly sample 1,000 examples from the dataset for experimentation. The models are tasked with generating specific goals on their own, and we then re-run the experiments. The experimental results are presented in Table 11. From the experimental results, it is evident that allowing the models to generate specific goals on their own leads to varying effects—some models exhibit higher scores, while others show lower scores. However, the overall change is not substantial enough to be unacceptable. Allowing the model to generate specific goals on its own is also a promising improvement approach.

G Results

Brahman et al. (2024) indicate that the correlation between the automated metric scores and human scores is weak. Therefore, we only present the experimental results of ROUGE, BLEU, and BERTScore for each task, without further detailed analysis. Table 20 presents the BLEU, ROUGE, and BERTScore for each model in the deductive reasoning task. Table 21 provides the corresponding results for each model in the inductive reasoning task. Table 22 reports the performance of the improved models in the inductive reasoning task.

/*Task prompt*/

Please follow the example below to generate the output for me. Generate only output, do not repeat the question.

/*Examples*/

Abstract Goal: List the steps of saving money.

Steps:

1. Set a financial goal for how much you want to save.
2. Review your income and expenses to understand your current financial situation.
3. Create a budget that allocates a portion of your income for savings.
4. Open a savings account, if you don't already have one.

...

Specific Goal: List the steps of saving money as a kid.

Steps:

1. Set a small savings goal, like saving for a toy or video game.
2. Ask your parents for a piggy bank or a special jar to keep your money safe.
3. Collect your allowance or any money you receive from chores, gifts, or special occasions.
4. Decide to save a portion of your money instead of spending it all.

...

Abstract Goal: List the steps of organizing a party.

Steps:

1. Decide on the date and time for the party.
2. Choose a theme or type of party (optional).
3. Create a guest list.
4. Send out invitations to your guests.

...

Specific Goal: List the steps of organizing a birthday party.

Steps:

1. Decide on a date and time for the birthday party.
2. Choose a theme (optional).
3. Create a guest list.
4. Send out invitations.

...

/*Completion*/

Abstract Goal: List the steps of {Abstract Goal}

Steps:

{Abstract Procedural Planning}

Specific Goal: List the steps of {Specific Goal}.

Steps: answer

Table 12: An example of prompt for models to perform deductive reasoning. {Abstract Procedural Planning}, {Abstract Goal}, and {Specific Goal} will be replaced with specific content from the dataset. Generated texts are highlighted.

/*Task prompt*/

Please follow the example below to generate the output for me. Generate only output, do not repeat the question.

/*Examples*/

Specific Goal: List the steps of saving money as a kid.

Steps:

1. Set a small savings goal, like saving for a toy or video game.
2. Ask your parents for a piggy bank or a special jar to keep your money safe.
3. Collect your allowance or any money you receive from chores, gifts, or special occasions.
4. Decide to save a portion of your money instead of spending it all.

...

Abstract Goal: List the steps of saving money.

Steps:

1. Set a financial goal for how much you want to save.
2. Review your income and expenses to understand your current financial situation.
3. Create a budget that allocates a portion of your income for savings.
4. Open a savings account, if you don't already have one.

...

Specific Goal: List the steps of organizing a birthday party.

Steps:

1. Decide on a date and time for the birthday party.
2. Choose a theme (optional).
3. Create a guest list.
4. Send out invitations.

...

Abstract Goal: List the steps of organizing a party.

Steps:

1. Decide on the date and time for the party.
2. Choose a theme or type of party (optional).
3. Create a guest list.
4. Send out invitations to your guests.

...

/*Completion*/

Specific Goal: List the steps of {Specific Goal}

Steps:

{Specific Procedural Planning}

Abstract Goal: List the steps of {Abstract Goal}.

Steps: answer

Table 13: An example of prompt for models to perform inductive reasoning. {Abstract Procedural Planning}, {Abstract Goal}, and {Specific Goal} will be replaced with specific content from the dataset. Generated texts are highlighted.

/*Task prompt*/ Please follow the example below to generate the output for me. Output only yes or no.
/*Examples*/ **Procedural Planning**: 1. Set a financial goal for how much you want to save. 2. Review your income and expenses to understand your current financial situation. 3. Create a budget that allocates a portion of your income for savings. ... **Question**: This is the procedural plan of saving money, but is this the procedural plan of saving money as a kid? **Answer**: no || **Procedural Planning**: 1. Set a small savings goal, like saving for a toy or video game. 2. Ask your parents for a piggy bank or a special jar to keep your money safe. 3. Collect your allowance or any money you receive from chores, gifts, or special occasions. ... **Question**: This is the procedural plan of saving money, but is this the procedural plan of saving money as a kid? **Answer**: yes || **Procedural Planning**: 1. Decide on the date and time for the party. 2. Choose a theme or type of party (optional). 3. Create a guest list. ... **Question**: This is the procedural plan of organizing a party, but is this the procedural plan of organizing a birthday party? **Answer**: no || **Procedural Planning**: 1. Decide on a date and time for the birthday party. 2. Choose a theme (optional). 3. Create a guest list. ... **Question**: This is the procedural plan of organizing a party, but is this the procedural plan of organizing a birthday party? **Answer**: yes ||
/*Completion*/ **Procedural Planning**: {Abstract Procedural Planning} **Question**: This is the procedural plan of {Abstract Goal}, but is this the procedural plan of {Specific Goal}? **Answer**: answer

Table 14: An example of prompt for GPT-4o-mini to determine whether an abstract procedural plan in the dataset can achieve a specific goal. {Abstract Procedural Planning}, {Abstract Goal}, and {Specific Goal} will be replaced with specific content from the dataset. Generated texts are highlighted. The result is either yes or no.

/*Task prompt*/

Please follow the example below to generate the output for me. Output only yes or no.

/*Examples*/

Procedural Planning:

1. Set a financial goal for how much you want to save.
2. Review your income and expenses to understand your current financial situation.
3. Create a budget that allocates a portion of your income for savings.
4. Open a savings account, if you don't already have one.

...

Question: Can this procedural plan achieve the goal of saving money as a kid?

Answer: no ||

Procedural Planning:

1. Read the recipe.
2. Get the ingredients and materials you need.
3. Measure each ingredient according to the recipe.
4. Preheat the oven.

...

Question: Can this procedural planning achieve the goal of baking a cake?

Answer: yes ||

Procedural Planning:

1. Decide on the date and time for the party.
2. Choose a theme or type of party (optional).
3. Create a guest list.
4. Send out invitations to your guests.

...

Question: Can this procedural plan achieve the goal of organizing a birthday party?

Answer: no ||

Procedural Planning:

1. Walk into library.
2. Find book on shelf.
3. Walk to check out desk.
4. Hand book to librarian.

...

Question: Can this procedural plan achieve the goal of borrowing a book from the library?

Answer: yes ||

/*Completion*/

Procedural Planning:

{Specific Procedural Planning}

Question: Can this procedural plan achieve the goal of {Specific Goal}?

Answer: answer

Table 15: An example of prompt for GPT-4o-mini to determine whether a generated procedural plan can achieve a specific goal. {Specific Procedural Planning}, {Abstract Goal}, and {Specific Goal} will be replaced with specific content. Generated texts are highlighted. The result is either yes or no.

/*Task prompt*/

Please follow the example below to generate the output for me. Output only yes or no.

/*Examples*/

Procedural Planning:

1. Walk into library.
2. Find book on shelf.
3. Walk to check out desk.
4. Hand book to librarian.

...

Question: Can this procedural planning achieve the goal of saving money?

Answer: no ||

Procedural Planning:

1. Read the recipe.
2. Get the ingredients and materials you need.
3. Measure each ingredient according to the recipe.
4. Preheat the oven.

...

Question: Can this procedural planning achieve the goal of baking a cake?

Answer: yes ||

Procedural Planning:

1. Go to the bathroom.
2. Get undressed.
3. Start the shower.
4. Use any soap, shampoo etc.

...

Question: Can this procedural planning achieve the goal of organizing a party?

Answer: no ||

Procedural Planning:

1. Walk into library.
2. Find book on shelf.
3. Walk to check out desk.
4. Hand book to librarian.

...

Question: Can this procedural plan achieve the goal of borrowing a book from the library?

Answer: yes ||

/*Completion*/

Procedural Planning:

{Abstract Procedural Planning}

Question: Can this procedural plan achieve the goal of {Abstract Goal}?

Answer: answer

Table 16: An example of prompt for GPT-4o-mini to determine whether a generated procedural plan can achieve an abstract goal. {Abstract Procedural Planning}, {Abstract Goal}, and {Specific Goal} will be replaced with specific content. Generated texts are highlighted. The result is either yes or no.

/*Task Description*/

You are tasked with comparing two abstract procedural plans (**Abstract Procedural Planning A ** and **Abstract Procedural Planning B**) based on their ability to generalize from the specific procedural plan. Specifically, you need to determine which abstract procedural plan captures the essential steps, logic, and general principles of the **specific procedural planning**, while maintaining the ability to be applied to similar tasks or scenarios. Your evaluation should focus on how well each abstract plan can extrapolate the process described in the **specific procedural planning** and apply it to a broader range of contexts. Please evaluate both abstract procedural plans based on the following criteria:

/*Evaluation Criteria*/

1. **Generality and Inductive Ability**:

- Which abstract procedural plan (**A** or **B**) is better at capturing the core logic and generalizable steps of the **specific procedural planning**?
- Which one can be applied to more diverse tasks, scenarios, or variations while preserving the overall logical structure from the original procedure?
- Does **Abstract Procedural Planning A** or **B** demonstrate a stronger ability to extend to new or unforeseen situations beyond the given task?

2. **Logical Consistency and Coherence**:

- Which abstract procedural plan maintains a more consistent, logical sequence of steps?
- Which one organizes the steps in a way that is clear and easy to follow, while still being applicable to other similar tasks or variations?
- Which script better preserves the integrity of the original **specific procedural planning** logic and stepwise structure?

3. **Adaptability**:

- Which abstract procedural plan can more easily accommodate variations, such as different ingredients, methods, or tools, without needing significant modifications to the structure?
- Consider how each abstract plan allows for flexibility. For example, can **Abstract Procedural Planning A** be applied to different types of tasks, such as recipes with other ingredients or different procedures, without major adjustments?
- Does **Abstract Procedural Planning B** offer more adaptability for future task variations?

4. **Simplicity and Clarity**:

- Which abstract procedural plan is simpler, clearer, and easier to follow?
- Does one of the abstract plans break down the steps into more understandable or actionable components?
- Is one of the abstract plans more intuitive and user-friendly for someone unfamiliar with the **{Abstract Goal}**?

/*Procedural Planning to Compare*/

****Specific Procedural Planning:**** {Specific Procedural Planning}
****Abstract Procedural Planning A:**** {Procedural Planning in the dataset}
****Abstract Procedural Planning B:**** {Generated Procedural Planning}

/*Questions*/

Based on the above evaluation criteria, determine which abstract procedural plan (**A** or **B **) better generalizes from the **specific procedural planning** and captures the essential steps of **{Abstract Goal}** in a way that can be more broadly applied to a variety of tasks, scenarios, or modifications. Output only **Abstract Procedural Planning A** or **Abstract Procedural Planning B**.

Answer: answer

Table 17: An example of prompt for GPT-4o-mini to determine whether a generated procedural plan is better than an abstract procedural plan in the dataset. {Abstract Goal}, {Procedural Planning in the dataset}, and {Generated Procedural Planning} will be replaced with specific content. Generated texts are highlighted. The result is either **Abstract Procedural Planning A** or **Abstract Procedural Planning B**.

/*Task Description*/ Given an abstract goal, generate two specific and concise goals related to it. Each goal should be as brief and straightforward as possible while adding relevant restrictions. Ensure that the specific goals differ from the example goal provided (i.e., {Specific Goal}) and focus on different aspects of the goal.
/*Examples*/ **Abstract goal:** making a memory board **Specific goal 1:** making a memory board with notes **Specific goal 2:** make a memory board with photos **Abstract goal:** making photo blocks **Specific goal 1:** making photo blocks with friends **Specific goal 2:** making photo blocks with family **Abstract goal:** eating dragon fruit **Specific goal 1:** eating dragon fruit with a spoon **Specific goal 2:** eating dragon fruit topped with yogurt
/*Completion*/ **Abstract Goal:** {Abstract Goal} Generated Specific Goals

Table 18: An example of prompt for GPT-4o-mini for specific goals generation via in-context learning. Generated texts are highlighted. {Specific Goal} and {Abstract Goal} will be replaced with specific content.

Abstract Goal : Making a Sundae
1. Gather all the ingredients: ice cream, toppings, and a bowl.
2. Choose your favorite flavor of ice cream.
3. Scoop the ice cream into the bowl.
4. Add your desired toppings, such as chocolate syrup, sprinkles, or nuts.
5. Optionally, add whipped cream on top.
6. Place a cherry on top if desired.
7. Grab a spoon and enjoy your sundae.
Specific Goal : Making a Sundae with fruit
1. Gather all the ingredients: ice cream, fruit (such as bananas, strawberries, or cherries), whipped cream, and any toppings (like nuts or chocolate syrup).
2. Choose a bowl or glass to serve the sundae.
3. Scoop the desired amount of ice cream into the bowl.
4. Slice the fruit into bite-sized pieces.
5. Arrange the sliced fruit on top of the ice cream.
6. Add whipped cream on top of the fruit.
7. Drizzle chocolate syrup or any other topping over the whipped cream.
8. Sprinkle nuts or other toppings if desired.
9. Serve immediately with a spoon.

Table 19: Dataset Example: Abstract and Specific Goals with Corresponding Procedural Plans.

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
Llama-3-8B	27.46	59.94	30.20	41.24	77.62
Mistral	30.15	61.98	32.17	43.61	78.90
OLMo-7B	19.58	53.22	24.64	34.98	73.57
OLMo-13B	24.59	59.05	26.84	39.62	77.56
Qwen2.5-7B	30.45	62.37	32.47	43.97	78.77
Qwen2.5-14B	26.32	60.28	29.00	41.00	77.77
Qwen2.5-32B	23.36	58.52	26.93	39.15	76.79
Claude-3	28.81	61.92	31.81	43.22	78.32
GPT-3.5-turbo	39.57	64.64	40.61	52.55	80.89
GPT-4o-mini	32.78	65.07	36.12	47.04	80.13

Table 20: The BLEU, ROUGE, and BERTScore of each model in the deductive reasoning task.

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
Llama-3-8B	29.02	59.92	30.36	42.12	77.93
Mistral	28.90	60.61	30.49	43.30	78.85
OLMo-7B	19.41	52.97	23.05	34.75	74.43
OLMo-13B	20.12	55.22	22.66	37.06	76.96
Qwen2.5-7B	25.45	58.31	26.43	40.12	77.27
Qwen2.5-14B	20.95	56.44	22.45	37.23	76.75
Qwen2.5-32B	21.27	57.43	23.64	37.93	76.70
Claude-3	29.23	61.09	30.87	43.12	78.42
GPT-3.5-turbo	32.73	62.41	34.76	48.54	79.77
GPT-4o-mini	27.32	60.77	28.44	42.10	78.58

Table 21: The BLEU, ROUGE, and BERTScore of each model in the inductive reasoning task.

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
Llama-3-8B	15.94	42.77	27.22	33.36	70.58
Mistral	29.73	58.34	36.75	45.14	77.45
OLMo-7B	8.19	39.35	18.20	25.87	64.62
OLMo-13B	11.37	47.47	20.40	30.34	67.33
Qwen2.5-7B	12.96	46.28	25.73	34.65	68.45
Qwen2.5-14B	9.93	43.46	20.38	30.12	67.63
Qwen2.5-32B	11.72	45.83	21.16	31.25	68.88
Claude-3	21.32	49.89	28.78	37.24	73.06
GPT-3.5-turbo	21.16	52.45	26.16	36.71	74.03
GPT-4o-mini	16.78	52.51	28.66	38.81	70.88

Table 22: The BLEU, ROUGE, and BERTScore of each improved model in the inductive reasoning task.