LLMs can Find Mathematical Reasoning Mistakes by Pedagogical Chain-of-Thought

Zhuoxuan Jiang\textsuperscript{1,2}, Haoyuan Peng\textsuperscript{2}, Shanshan Feng\textsuperscript{3,4}, Fan Li\textsuperscript{5} and Dongsheng Li\textsuperscript{6}

\textsuperscript{1}Shanghai Business School, Shanghai, China
\textsuperscript{2}Learnable.AI Inc., Shanghai, China
\textsuperscript{3}Centre for Frontier AI Research, A*STAR, Singapore
\textsuperscript{4}Institute of High-Performance Computing, A*STAR, Singapore
\textsuperscript{5}The Hong Kong Polytechnic University, Hong Kong, China
\textsuperscript{6}Microsoft Research Asia, Shanghai, China

\texttt{jzx@sbs.edu.cn, penghy15@gmail.com, victor_fengss@foxmail.com, fan-5.li@polyu.edu.hk, dongshengli@fudan.edu.cn}

Abstract

Self-correction is emerging as a promising approach to mitigate the issue of hallucination in Large Language Models (LLMs). To facilitate effective self-correction, recent research has proposed mistake detection as its initial step. However, current literature suggests that LLMs often struggle with reliably identifying reasoning mistakes when using simplistic prompting strategies. To address this challenge, we introduce a unique prompting strategy, termed the Pedagogical Chain-of-Thought (PedCoT), which is specifically designed to guide the identification of reasoning mistakes, particularly mathematical reasoning mistakes. PedCoT consists of pedagogical principles for prompts (PPP) design, two-stage interaction process (TIP) and grounded PedCoT prompts, all inspired by the educational theory of the Bloom Cognitive Model (BCM). We evaluate our approach on two public datasets featuring math problems of varying difficulty levels. The experiments demonstrate that our zero-shot prompting strategy significantly outperforms strong baselines. The proposed method can achieve the goal of reliable mathematical mistake identification and provide a foundation for automatic math answer grading. The results underscore the significance of educational theory, serving as domain knowledge, in guiding prompting strategy design for addressing challenging tasks with LLMs effectively.

1 Introduction

In recent years, Large Language Models (LLMs) have emerged as the leading force in advancing various Natural Language Processing (NLP) tasks, consistently achieving state-of-the-art performance. Despite the remarkable improvement in general AI abilities, LLMs still suffer from some essential problems, e.g., hallucination, which suggests that current LLMs may generate contents that look reasonable but intrinsically illogical [Huang \textit{et al.}, 2023b].

To overcome hallucination, many efforts have been made, among which \textit{self-correction} is one of the most recently focused directions. It empowers LLMs to correct their outputs by zero- or few-shot prompting [Pan \textit{et al.}, 2023]. Some researchers treat self-correction as a single-process way [Chen \textit{et al.}, 2023; Madaan \textit{et al.}, 2023], while some researchers break down the task into two components, i.e., mistake finding and output correction, and achieve superior performance [Tyen \textit{et al.}, 2023]. Since finding mistakes is a fundamental and precedent skill for right correction, it is reasonable that the two-step self-correction is more reliable.

Mistake finding is a prerequisite of self-correction, while how to well achieve it is still an open challenge. Some studies show that current LLMs struggle to find mistakes reliably using basic prompting strategies [Huang \textit{et al.}, 2023a; Tyen \textit{et al.}, 2023]. Most previous benchmark studies mainly adopt simple and general prompting strategies, e.g., Zero-shot CoT and its variants\textsuperscript{1}. They do not fully leverage domain-related knowledge to design prompts. Domain-related knowledge is important because there are various kinds of reasoning problems, such as word sorting, tracking shuffled objects, logical deduction and multi-step arithmetic [Srivastava \textit{et al.}, 2023], which require different kinds of fine-grained reasoning skills. For example, word sorting needs the skills of text and order understanding while multi-step arithmetic needs the skills of calculation and deduction. Hence, without domain-related knowledge, the potential of LLMs may be underestimated due to inadequately designed prompts.

In this paper, we investigate the challenge of creating efficient prompts for a specific domain to identify reasoning errors utilizing LLMs. As a starting point, we choose the mathematical domain, a basic subject in education. On the one hand, mathematical questions queries necessitate rigorous reasoning and logic, an area where current LLMs fall

\textsuperscript{1}They are Direct trace-level prompting, Direct step-level prompting, and CoT step-level prompting. In this paper, we follow their task settings. The details of benchmark experiments with more prompting strategies will be introduced in Section 5.
short. On the other hand, finding reasoning mistakes in mathematics can provide a foundation for automatic answer grading in terms of pedagogy. To this end, we conduct interdisciplinary research on LLMs prompting strategies by adopting pedagogical knowledge within this work to achieve reliable mathematical mistake finding.

In the field of pedagogy, the teaching and learning objectives are theoretically consistent with the Bloom Cognitive Model (BCM)\(^2\) [Bloom et al., 1984], which breaks the student abilities into six levels from a cognitive perspective, i.e., Remember, Understand, Apply, Analyze, Evaluate, and Create (as shown in Figure 1) [Anderson and Krathwohl, 2001]. According to this theory, the first three foundational levels are associated with learning ability, signifying that these skills are relatively easier to acquire, whereas the upper three levels involve planning ability, which are more complex to nurture and develop. We focus on the learning abilities in this work and leave the exploration of planning abilities as future work.

To align with BCM and bridge the gap in design prompting strategies for LLMs, we introduce a novel method named Pedagogical Chain-of-Thought (abbr. PedCoT). More specifically, the PedCoT strategy embodies two parts. First, following the BCM and the levels of learning ability, we propose the pedagogical principles of prompt (PPP) design for LLMs. The detailed content of PPP is consistent with learning ability, as shown in Figure 1. Second, concerning the content of prompts to be grounded when interacting with LLMs, we formulate the two-stage interaction process (TIP) and grounded prompts, consistent with the three principles.

To assess the efficacy of our approach, we gather two publicly available datasets, each containing mathematical problems of varying degrees of complexity, i.e., multi-step arithmetic and multi-step word problems. The experimental results, compared against strong baselines, consistently reveal a noteworthy contrast to what most existing studies have asserted. Specifically, we observe that current LLMs, such as GPT-4 Turbo, can effectively find mathematical reasoning mistakes through our proposed PedCoT. The result highlights the importance and value of domain knowledge in prompting the reasoning abilities of LLMs.

\(^2\)Also called Bloom’s Taxonomy of Educational Objectives in Cognitive Domain.

Figure 1: We develop the principles for prompt design for LLMs by leveraging the educational Bloom Cognitive Model and we focus on the learning ability. The bold parts are keywords used in prompts.

The main contributions of this paper include:

- We conduct an interdisciplinary study and investigate a new problem on how to leverage domain knowledge to guide prompt design for LLMs, i.e., leveraging pedagogical theories to find mathematical mistakes.
- We develop a novel zero-shot prompting strategy named PedCoT to bridge the gap between educational theory and prompts for LLMs, which is featured by (1) pedagogical principles for prompt (PPP) design, (2) two-stage interaction process (TIP) and (3) PedCoT prompts.
- Experiments on two public datasets with various complexity of math problems demonstrate that contrary to what most existing studies claim, current LLMs actually can find mathematical reasoning mistakes by using our PedCoT equipped with pedagogical theory.

2 Related Work

2.1 Automatic Answer Grading

The task of finding mistakes among mathematical reasoning steps is analogous to the task of automatic answer grading (ASG) from the perspective of pedagogy. The ASG has garnered significant attention within the educational technology community. The methods for ASG are manifold, which depend on specific subjects and/or question types, e.g., automatic scoring for essay [Xie et al., 2022], reading comprehension [Fernandez et al., 2022], open math response [Erickson et al., 2020] and short math answering grading [Zhang et al., 2022a]. The majority of these studies focus on improving semantic modeling by incorporating the most recent language models from the Natural Language Processing (NLP) community. As to the mathematical grading task, besides capturing the semantics, evaluating the rationale (intermediate reasoning steps) on the word problem step-by-step is still much more challenging [Cobbe et al., 2021; Welleck et al., 2022]. The emergence of LLMs significantly improves the level of machine intelligence and provides a potential direction for finding mistakes. However, the difference between ASG and mistake finding is that the former can utilize the standard answer information while the latter cannot. To the best of our knowledge, this paper represents the first attempt to leverage the pedagogical theories from step-by-step grading of mathematical answers for LLMs.

2.2 Chain-of-Thought Prompting

LLMs indeed shown great potential in various tasks, but prompting them efficiently can be challenging. Recently, the concept of Chain-of-Thought (CoT) has been introduced [Wei et al., 2022]. Then various zero- and few-shot CoT prompting methods are studied to demonstrate the ability of LLMs to solve challenging tasks [Wang et al., 2022]. For example, in terms of zero-shot prompting, Zero-shot CoT simply adds ‘Let’s think step by step’ after each question, which enables LLMs to generate step-by-step rationales and achieve better final answers [Kojima et al., 2022]. Later, Plan-and-Solve Prompting instructs LLMs to devise a plan by breaking down the entire task into smaller ones and then executing each subtask to accomplish the plan, addressing the step missing and calculation issues which Zero-shot CoT may
2.3 Detecting Reasoning Errors with LLMs

Self-correction is one of the recently focused topics among zero- and few-shot prompting methods to tackle hallucination problems [Saunders et al., 2022; Chen et al., 2023; Madaan et al., 2023]. It can be divided into two steps: mistake finding and output correction. The former step is regarded as a fundamental skill and normally addressed by using some few-shot prompting methods [Weng et al., 2023; Ling et al., 2023]. Nevertheless, recent studies demonstrate that the current LLMs still struggle to identify mistakes even in objective, unambiguous cases [Huang et al., 2023a; Tyen et al., 2023]. Hence, some studies indicate that even the state-of-the-art LLMs perform poorly in fixing their own reasoning errors without external feedback and LLMs are only able to improve their outputs in terms of style and quality [Kim et al., 2023; Shinn et al., 2023].

A recent study proposed SelfCheck [Miao et al., 2023], which promotes LLMs to detect reasoning mistakes in a zero-shot manner. SelfCheck can partially address the problem of detecting reasoning mistakes of LLMs, while it still has some problems: (1) SelfCheck may be faced with a bias when the regenerated and real current steps to be compared are consistently wrong. To deal with the bias, PedCoT considers more comprehensive and reliable factors, including pedagogical theories, various mistake types, and different mistake labels; (2) SelfCheck requires five LLM requests to complete a check, and it is inefficient; and (3) SelfCheck is evaluated by its performance in detecting reasoning mistakes without comparing with other related methods. We conduct more comprehensive experiments on various CoT prompting methods and achieve the state-of-the-art performance.

3 Problem Statement and Analysis

In this section, we will firstly provide a detailed definition of the research problem. Subsequently, we will undertake a comprehensive analysis of the research problem and introduce the proposed solution to address this problem.

3.1 Problem Definition

We propose a novel problem of finding mathematical reasoning mistakes by zero-shot prompting LLMs. Mathematical reasoning mistakes can be divided into three-hierarchy decision-making mistakes: namely principle mistake, step mistake, and trace mistake respectively.

Given a question (i.e., math problem) $Q$ and a trace of the top $i$ answer steps $A_i = \{a_1, \ldots, a_i\}$, as variables, they are integrated into predefined instruction templates $T = \{T_j\}$ and then the combined prompts can be represented by $P = \{P_j\}$, where $j$ means the $j$-th stage of request on a pre-selected LLM (e.g., GPT-4) and $k$ corresponds to the $k$-th pedagogical principle. In this work, we set the $J = 2$ and $K = 3$ since we only need two stages of requests on LLM to complete the mistake finding task and we leverage three pedagogical principles for the time being. Therefore, at Stage-1, the input prompt $P_1$ is combined with $Q$, $A_{i-1}$ and $T_1$ to request on the LLM to obtain the intermediate content $G^{(k)}$. The process is named as Regenerate and is introduced in the following section. At Stage-2, the input prompt $P_2$ is combined with $Q$, $A_i$, $T_2$ and $G$ to request on the LLM again to extract some content $E^{(k)}$ and compare the consistency between the regenerated content and extracted content. This process is called as Extract-Compare. The LLM can finally output a principle mistake label in terms of principles which is defined as $L_{principle}^{(k)} \in \{\text{correct-and-aligned, reasonable-but-incomplete, contradic-}

Figure 2: Diagram of two-stage interaction process (TIP) with LLMs for finding mistakes at the $i$-th step. The left and right parts are the exemplifying input and output contents. The detailed contents, as well as complete prompts, of the example can be referred to Appendix.
Table 1: The proposed PedCoT prompting method which evolved from left two columns of pedagogical ability levels and principles for prompt design. We design the instruction templates for PedCoT prompts that coordinate with the two-stage interaction process (TIP). The bold parts are keywords to embody the corresponding educational meanings. We refine the wording of PedCoT prompts to avoid misunderstanding by LLMs through numerous empirical attempts. The contents in [] are variables.

<table>
<thead>
<tr>
<th>Ability Levels</th>
<th>Principles</th>
<th>Instruction Templates for Two-stage PedCoT Prompts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remember</td>
<td>Whether the right mathematical concepts are involved.</td>
<td>You are given a math problem and several initial steps ... Question: [Q] Initial steps: [Previous Answer Steps (a_1, \ldots, a_{i-1})] Execute the following instructions sequentially. 1. List the mathematical concepts that ... in the next step ... 2. List the key analyses that ... in the next step ... 3. List the mathematical expressions that ... in the next step ...</td>
</tr>
<tr>
<td>Understand</td>
<td>Whether the right problem solving approaches are adopted.</td>
<td>You are given a math problem and several initial steps ... Question: [Q] Initial steps: [Previous Answer Steps (a_1, \ldots, a_{i-1})]</td>
</tr>
<tr>
<td>Apply</td>
<td>Whether the right calculations are made.</td>
<td></td>
</tr>
</tbody>
</table>

As shown in Figure 1, six ability levels are defined by the Bloom Cognitive Model which explains the teaching objectives from the cognitive perspective of pedagogy. From bottom (low) to up (high), they are Remember, Understand, Apply, Analyze, Evaluate, and Create. Along with the theory, the lower three levels are called learning ability which are easier to obtain, while the higher three levels, planning ability, are notably more complex to nurture and develop. This paper focuses on the lower three ability levels.

Pedagogical Principles for Prompt Design. To fill the gap between pedagogical theories and grounded prompts for LLMs to execute, we propose three pedagogical principles for prompt (PPP) design, corresponding to the three ability levels. The PPP can guide the later grounded prompt design on what aspects should be involved. The right part of Figure 1 shows the content of principles. Admittedly, based on the BCM, each ability level contains various aspects of fine-grained skills, and the proposed principles cannot fully reflect them. For example, remembering mathematical concepts is just a representative skill among 14 counterparts at the level of Remember. The detail of ability levels can be referred to the theory [Anderson and Krathwohl, 2001]. As a starting point, we propose to defer the exploration of more intricate principles to subsequent research endeavours.

4 Methodology

Contrary to existing methodologies that solely rely on the inherent capabilities of LLMs to identify errors - an approach that often proves ineffective - we instead take a pioneer study to utilize external domain knowledge on the targeted problem.

\[ L_{step}^{(i)} = \begin{cases} 1, & \text{if } \forall k \in \{1,2,3\}, \text{st.} L_{principle}^{(k)} \neq \text{contradiction-found}, \\ 0, & \text{if } \exists k \in \{1,2,3\}, \text{st.} L_{principle}^{(k)} = \text{contradiction-found}. \end{cases} \]

\[ L_{trace} = \begin{cases} 1, & \text{if } \forall i \in \{1, \ldots, n\}, \text{st.} L_{step}^{(i)} = 1, \\ 0, & \text{if } \exists i \in \{1, \ldots, n\}, \text{st.} L_{step}^{(i)} = 0, \end{cases} \]

where \( n \) is the number of steps in an answer trace. Actually, the answer steps are in a sequential order, and the step mistake labels are obtained orderly. Once the first step mistake is found, the trace is identified as mistake found and the iteration can stop. We believe that the above proposed principle mistake label schema is aligns well with educational objectives for answer grading. It provides explainable information for understanding the logic of mistake identification, as opposed to the binary judgment employed by most existing methods.

3.2 Analysis

The basic idea on addressing the problem of finding mathematical reasoning mistakes is to leverage pedagogical theories and bridge the gap between pedagogy and LLMs. Pedagogical theories are essential in the field of education as they provide a framework for understanding how and why students learn. They are widely used to assist teachers in evaluating the students’ learning outcomes. Hence, it is believed that the pedagogical theories can be transferred to instruct LLMs in mistake finding and self-correction. Following the framework of pedagogical theories, we first develop the principles for prompt design, and then the grounded prompts can be designed in a systematical manner. In this paper, we develop three pedagogical principles for prompt (PPP) design according to ability levels that are defined by Bloom Cognitive Model. The detail analysis of the developed PPP is discussed as below:

**Ability Level.** As shown in Figure 1, six ability levels are defined by the Bloom Cognitive Model which explains the teaching objectives from the cognitive perspective of pedagogy. From bottom (low) to up (high), they are Remember, Understand, Apply, Analyze, Evaluate, and Create. Along with the theory, the lower three levels are called learning ability which are easier to obtain, while the higher three levels, planning ability, are notably more complex to nurture and develop. This paper focuses on the lower three ability levels.

**Table 1** The proposed PedCoT prompting method which evolved from left two columns of pedagogical ability levels and principles for prompt design. We design the instruction templates for PedCoT prompts that coordinate with the two-stage interaction process (TIP). The bold parts are keywords to embody the corresponding educational meanings. We refine the wording of PedCoT prompts to avoid misunderstanding by LLMs through numerous empirical attempts. The contents in [] are variables.
In this paper, we design the novel PedCoT prompts for mathematical reasoning mistake finding based on the previously developed PPP design. The PedCoT prompts are also coordinated with the newly-proposed two-stage interaction process (TIP) on LLMs. We introduce the TIP first.

4.1 Two-Stage Interaction Process

Assume to find mistake at the $i$-th answer step. Figure 2 shows the two-stage interaction process with LLMs, including Stage-1 Regenerate and Stage-2 Extract-Compare.

**Stage-1: Regenerate.** Given the prompt $P_1$ for Stage-1, which is a combination of a math question $Q$, the previous answer steps $A_{i-1} = \{a_1, ..., a_{i-1}\}$ and the instruction template $T_1$, LLMs regenerate the potential mathematical concepts, key analysis (i.e., problem-solving approaches), and calculation results (i.e., calculations) about the expected current step, denoted as $G \in \{G^{(k)}\}$. We use regular expressions to split the output contents $G$ into three segments to obtain each $G^{(k)}$. Unlike the majority of existing approaches that necessitate LLMs to explicitly regenerate the current step and compare it with the actual current step, our method can generate a pedagogical analysis of the step while keeping the actual current step, $a_i$, unseen to LLMs for the sake of objectivity.

**Stage-2: Extract-Compare.** Given the prompt $P_2$ for Stage-2, which is a combination of a math question $Q$, the current answer steps $A_i = \{a_1, ..., a_i\}$, the outputs $G$ from Stage-1 Regenerate and the instruction template $T_2$, LLMs firstly extract the real mathematical concepts, key analysis (i.e., problem-solving approaches), and calculation results (i.e., calculations) within the current step $a_i$, denoted as $E \in \{E^{(k)}\}$. Then, LLMs compare the extracted contents $E$ with regenerated contents $G$ at Stage-1 along with the respective principle. At last, LLMs make the conclusions on whether the mathematical concepts, problem-solving approaches, and/or calculations are rightly involved in the current step $a_i$, i.e., giving the principle mistake label $L^{(k)}_{\text{principle}}$.

Then the step mistake label $L^{(i)}_{\text{step}}$ can be acquired.

To find mistake at any step, our method only requires two requests. To the best of our knowledge, no existing other empirical attempts.

4.2 Pedagogical Chain-of-Thought Prompts

Based on pedagogical principles mentioned in Section 3.2, we develop the Pedagogical Chain-of-Thought (PedCoT) prompts for each stage of TIP. Each prompt consists of the instruction templates (i.e., $T_1$ or $T_2$) and the variables (i.e., $Q$, $A$ and/or $G$). As shown in Table 1, the right two columns list the grounded contents of developed prompts for each stage. The bold parts are keywords to embody the corresponding educational meanings. We refine the wording of grounded prompts to avoid misunderstanding by LLMs through numerous empirical attempts.

By utilizing the proposed PedCoT prompts, the LLMs can perform a binary assessment of the current step’s accuracy. Additionally, they can provide explanatory feedback for educational purposes by generating a textual principle mistake label for each ability level. Thus, our method can bridge the gap between pedagogical theories and prompt design for LLMs, and fulfill the task of step-by-step reasoning mistake finding in the mathematical domain.

5 Experiment

5.1 Datasets

To evaluate our proposed PedCoT, we collect two public datasets containing step-level correctness labels for mathematical problems with different difficulties. These datasets contains step-level correctness labels for mathematical problems with different difficulties.

**BIG-Bench Mistake** [Tyen et al., 2023]: The original dataset consists of 2,186 reasoning traces generated by PALM-2 [Anil et al., 2023], encompassing five reasoning tasks derived from diverse domains. With an emphasis on empowering LLMs to identify reasoning errors within the realm of mathematics, we chose samples exclusively related to this field. In total, these samples encompass 1,506 reasoning steps across 300 answer traces. The type of mathematical problem consistently involves multi-step arithmetic calculations. These are relatively straightforward to solve and necessitate basic mathematical concepts and problem-solving strategies, such as the fundamental rules of arithmetic. 62 correct traces are involved to avoid the bias that LLMs may treat every trace as faulty.

**PRM800K** [Lightman et al., 2023]: The original dataset contains about 800,000 step-level labels over 75,000 solutions for math word problems in the MATH dataset [Hendrycks et al., 2021], which consists of 12,500 problems from high-school math competitions. The problems necessitate a comprehensive understanding of intricate mathematical concepts and problem-solving methodologies. We adopt this challenging dataset to verify the effectiveness and robustness of our method. We randomly select a set of 300 pairs of problems and their affiliated reasoning answer traces, containing 85 correct traces and 3,736 reasoning steps in total. The steps in the PRM800K dataset are categorized as positive, negative, and neutral. To use the proposed method, we adjust the labels to align with our label schema. Specifically, the positive or neutral steps are treated as right (i.e., $L_{\text{step}} = 1$) while the negative steps are labeled as wrong (i.e., $L_{\text{step}} = 0$) in our experiments.

5.2 Models, Baselines, and Metrics

**Models.** Our experiments are conducted with two state-of-the-art LLMs: GPT-4 [OpenAI, 2023] and its latest generation GPT-4 Turbo. The temperature for generation is consistently set to 0 for both models to minimize the diversity of model outputs. We do not employ any ensemble strategies, such as self-consistency [Wang et al., 2022], to maintain a pure comparison with different CoT baseline methods.

**Baselines.** We compare our proposed PedCoT with the following zero-shot prompting methods: (1) Zero-shot CoT [Kojima et al., 2022]; (2) Plan-and-Solve Prompting [Wang et al., 2023]; (3) SelfCheck [Miao et al., 2023]. We re-implement these baselines by prompting LLMs with

---

3Accessible on github.com/HaoyuanPeng/PedCoT-IJCAI24/
4We used gpt-4-1106-preview, the latest version then.
the same prompts and multi-stage pipelines. To better evaluate the abilities of LLMs on detecting reasoning tasks with CoT prompting, we also consider Direct-Step Prompting [Tyen et al., 2023] and Vanilla Two-stage Prompting as our baselines. Direct-Step Prompting instructs LLMs to generate a single word yes for each step if it is correct, or no otherwise. Vanilla Two-Stage Prompting first prompts LLMs to analyze each step’s correctness without specific instructions. Subsequently, LLMs are prompted again to generate a label word based on their analyses.

**Metrics.** Similar to the previous work that evaluates the performance of mistake finding [Tyen et al., 2023], we also utilize the mistake finding accuracy (abbr. MF. Acc.) and weighted average F1 (abbr. Avg. F1) scores as our metrics. Specifically, the MF. Acc. is strict for step-level accuracy which means mistakes should be found at the right step. Only the first mistake among a trace is concerned. The average F1 score provides a lenient measure of trace-level accuracy, indicating that mistakes within a trace scope can be identified. However, it does not account for whether these errors are pinpointed at the correct step. In addition, the binary classification accuracy (abbr. Cls. Acc.), as well as the precision ($P_+$ and $P_-$), recall ($R_+$ and $R_-$) and F1 ($F_+$ and $F_-$) for each label, are reported in the experiment.

### 5.3 Results and Analysis

Table 2 reports the comparison among different zero-shot CoT methods applied with GPT-4 and GPT-4 Turbo on the two datasets. Consistent with previous research, directly prompting the LLMs to output a single word for the correctness of each reasoning step (i.e., Direct-Step) is ineffective even with the state-of-the-art LLMs. Vanilla Two-stage Prompting allows LLMs to generate several analyses before being prompted again for the final label word, leading to a slightly better performance than Direct-Step. Zero-shot CoT and Plan-and-Solve Prompting explicitly instruct LLMs to generate intermediate analyses, and show improvements compared to Vanilla Two-stage Prompting. The above results indicate that the more analysis contents before giving the final judgment are generated by prompting LLMs, the better performance is obtained, especially with GPT-4 Turbo.

Our PedCoT consistently outperforms all the other baselines by a remarkable margin on the two datasets and two LLMs. Particularly, on the BIG-Bench Mistake dataset, PedCoT surpasses SelfCheck across both Acc. and Avg. F1 metrics. On the PRM800K dataset, PedCoT achieves a higher MF. Acc. score ranging from 4.00% to 7.33% compared to two-stage CoT baselines when using GPT-4. Additionally, its Avg. F1 and Cls. Acc. scores surpass these baselines by about 4.93% to 5.33%. Compared to SelfCheck, PedCoT exhibits a 5.33% leading in MF. Acc., and a considerable leading of 8.03% and 9.00% in Avg. F1 and Cls. Acc. scores respectively. When using GPT-4 Turbo, PedCoT exhibits a MF. Acc. advantage of 5.66% compared to the two-stage CoT baselines, along with improvements in Avg. F1 and Cls. Acc. In comparison to SelfCheck, PedCoT achieves a higher MF. Acc. by 2.66%, and a leading of 13.24% and 14.66% in Avg. F1 and Cls. Acc. respectively. All of the above results demonstrate the effectiveness of leveraging pedagogical knowledge in our proposed methods.

Additionally, although SelfCheck outperforms other baselines on BIG-Bench Mistake dataset, as displayed in Table 2, its Avg. F1 and Cls. Acc. scores on PRM800K dataset fall behind, as shown in Table 2. The results may be caused by the following reasons: During the process of SelfCheck prompting, one of the LLM requests is to summarize the target of the current answer step, and then regenerate a new current step for subsequent comparison. For the BIG-Bench Mistake dataset, due to the easy arithmetic ability required to solve math problems, SelfCheck can perform well. However, the PRM800K dataset presents more complex math problems requiring advanced reasoning skills. In such cases, SelfCheck

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BIG-Bench Mistake</td>
<td>GPT-4</td>
<td>Direct-Step</td>
<td>42.67</td>
<td>78.67</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vanilla</td>
<td>46.67</td>
<td>75.19</td>
<td>76.00</td>
<td>40.38</td>
<td>33.87</td>
<td>36.84</td>
<td>83.47</td>
<td>86.97</td>
<td>85.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Zero-shot CoT</td>
<td>46.33</td>
<td>72.82</td>
<td>73.33</td>
<td>33.93</td>
<td>30.65</td>
<td>32.20</td>
<td>82.38</td>
<td>84.45</td>
<td>83.40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Plan-and-Solve</td>
<td>47.00</td>
<td>73.17</td>
<td>74.33</td>
<td>34.69</td>
<td>30.07</td>
<td>30.63</td>
<td>86.55</td>
<td>84.25</td>
<td>84.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SelfCheck</td>
<td>81.00</td>
<td>88.37</td>
<td>88.33</td>
<td>71.43</td>
<td>72.58</td>
<td>72.00</td>
<td>98.83</td>
<td>92.44</td>
<td>92.63</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PedCoT (ours)</td>
<td>83.00</td>
<td>88.81</td>
<td>89.33</td>
<td>81.25</td>
<td>62.90</td>
<td>70.91</td>
<td>90.87</td>
<td>96.22</td>
<td>93.47</td>
</tr>
<tr>
<td>PRM800K</td>
<td>GPT-4 Turbo</td>
<td>Direct-Step</td>
<td>43.33</td>
<td>86.24</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vanilla</td>
<td>64.67</td>
<td>85.84</td>
<td>87.33</td>
<td>85.29</td>
<td>46.77</td>
<td>60.42</td>
<td>87.59</td>
<td>97.90</td>
<td>92.46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Zero-shot CoT</td>
<td>70.00</td>
<td>88.73</td>
<td>89.67</td>
<td>89.74</td>
<td>56.45</td>
<td>69.31</td>
<td>89.66</td>
<td>98.32</td>
<td>93.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Plan-and-Solve</td>
<td>77.00</td>
<td>87.33</td>
<td>88.67</td>
<td>91.18</td>
<td>50.00</td>
<td>64.58</td>
<td>88.35</td>
<td>98.74</td>
<td>93.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SelfCheck</td>
<td>80.33</td>
<td>87.84</td>
<td>87.67</td>
<td>74.19</td>
<td>71.32</td>
<td>93.13</td>
<td>91.18</td>
<td>92.14</td>
<td>92.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PedCoT (ours)</td>
<td>85.33</td>
<td>89.33</td>
<td>90.00</td>
<td>86.36</td>
<td>61.29</td>
<td>71.70</td>
<td>90.62</td>
<td>97.48</td>
<td>93.93</td>
</tr>
</tbody>
</table>

Table 2: Comparison of various CoT methods on the BIG-Bench Mistake and PRM800K datasets. †The scores are from the original paper.
may summarize an incorrect target, leading to a incidental alignment with an incorrect actual current step. This bias in SelfCheck can result in erroneously classifying the real current step as correct, impacting the precision for correct traces ($P_+$) and reducing recall for incorrect traces ($R_-$). The reason of our method can surmount these challenges met with SelfCheck, by our analysis, is that the PedCoT is grounded in pedagogical principles and allows LLMs to remain blind to the current step. This approach enables the LLMs to render judgments from an impartial and unbiased perspective.

### 5.4 Ablation Study

We design ablation experiments to assess the contribution of each principle. Specifically, we deactivate the three principles one by one, and observe the performance change patterns of our proposed methods. In this study, deactivating a principle means that this principle’s corresponding prompt contents are removed. We select 25% of the test set for ablation experiments. The results are reported in Table 3.

**Table 3: Ablation experiment of PedCoT by ablating one principle.**

<table>
<thead>
<tr>
<th>LLMs</th>
<th>Methods</th>
<th>BIG-Bench Mistake</th>
<th>PRM800K</th>
</tr>
</thead>
<tbody>
<tr>
<td>PedCoT</td>
<td>90.67</td>
<td>91.78</td>
<td>92.00</td>
</tr>
<tr>
<td>GPT-4</td>
<td>Principle 1</td>
<td>89.33</td>
<td>90.54</td>
</tr>
<tr>
<td></td>
<td>Principle 2</td>
<td>89.33</td>
<td>90.54</td>
</tr>
<tr>
<td></td>
<td>Principle 3</td>
<td>28.00</td>
<td>37.44</td>
</tr>
<tr>
<td>PedCoT</td>
<td>92.00</td>
<td>91.31</td>
<td>92.00</td>
</tr>
<tr>
<td>GPT-4</td>
<td>Principle 1</td>
<td>90.67</td>
<td>91.18</td>
</tr>
<tr>
<td></td>
<td>Principle 2</td>
<td>90.67</td>
<td>92.00</td>
</tr>
<tr>
<td>Turbo</td>
<td>Principle 3</td>
<td>21.33</td>
<td>32.22</td>
</tr>
</tbody>
</table>

Since the datasets have varying complexities of mathematical problems, the performance change patterns are not always significant. Regardless the performance stays stable or shows a significant decrease, it can be concluded that each principle has a positive impact on the PedCoT method. This demonstrates the effectiveness of incorporating the pedagogical theory of each learning ability from BCM.

### 5.5 Study on One-Stage vs. Two-Stage Prompting

In this section, we investigate the necessity of two-stage prompting. A distinctive feature of our two-stage methodology is that, at Stage-1, the current step remains undisclosed to LLMs and thus they can make an unbiased judgement at Stage-2. To verify the necessity, we construct a version of one-stage prompt by combining our two-stage prompts.

We compare the performance between the two-stage PedCoT and its one-stage variant, utilizing two datasets with GPT-4. As presented in Table 4, the precision of correct traces ($P_+$) and the recall of incorrect traces ($R_-$) significantly degrade when employing one-stage prompting. This suggests that the LLMs would be misled by the real current step present in the one-stage prompt. Based on our analysis, it appears that when the real current step is shared with the LLMs, they may encounter difficulties in independently generating analyses. Instead, they seem to rely heavily on referencing the real current step. We call this phenomenon ‘lazy LLMs’ since LLMs are inclined to cater to the ideas present in prompts. Consequently, finding mistakes by giving the real current step becomes difficult to accomplish. This experiment demonstrates that our two-stage prompting is necessary.

### 6 Conclusion

In this paper, we investigate a novel problem of how to leverage domain knowledge of pedagogy to enhance the complex reasoning abilities of the zero-shot prompted LLMs. We learn from pedagogical theories to instruct LLMs and to guide the prompts design for LLMs. We propose a novel strategy, namely PedCoT method, which consists of three pedagogy principles for prompt design, two-stage interaction process, and pedagogical Chain-of-Thought prompts. We experiment two public datasets of math problems with various complexity levels. The state-of-the-art performance of our proposed method indicates that it can achieve efficient reasoning mistake finding with only two requests to LLMs. In addition, the ablation experiment and study on one-stage method validate the importance of the three pedagogical principles and the necessity of two-stage prompting. Overall, this finding indicates that LLMs actually can find mathematical reasoning mistakes by resorting to domain knowledge.

**Table 4: Two-stage PedCoT vs. its one-stage variant on GPT-4.**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Strategy</th>
<th>MF Acc.</th>
<th>Cls. Acc.</th>
<th>$P_+$</th>
<th>$R_-$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIG-Bench Mistake</td>
<td>PedCoT</td>
<td>83.00</td>
<td>89.33</td>
<td>81.25</td>
<td>96.22</td>
</tr>
<tr>
<td>One-Stage</td>
<td></td>
<td>74.33</td>
<td>81.00</td>
<td>53.33</td>
<td>85.29</td>
</tr>
<tr>
<td>PRM800K</td>
<td>PedCoT</td>
<td>45.33</td>
<td>79.33</td>
<td>61.39</td>
<td>81.86</td>
</tr>
<tr>
<td>One-Stage</td>
<td></td>
<td>35.00</td>
<td>61.00</td>
<td>38.41</td>
<td>60.47</td>
</tr>
</tbody>
</table>
Acknowledgments

This work was supported by the Humanities and Social Science Fund of Ministry of Education (No. 23YJC790187). We thank the anonymous reviewers for their insightful and constructive comments.

Contribution Statement

Authors: Zhuoxuan Jiang, Haoyuan Peng, Shanshan Feng, Fan Li and Dongsheng Li

Zhuoxuan Jiang: Conceptualized the study, designed experiments, wrote the manuscript, supervised the project and secured funding.

Haoyuan Peng: Contributed to data collection, conducted experiments, performed result analyses, crafted the prompts and critically revised the manuscript.

Shanshan Feng, Fan Li and Dongsheng Li: Provided critical feedback on the manuscript, and edited for clarity.

Equal Contribution: Zhuoxuan Jiang and Haoyuan Peng contributed equally to this work and listed in alphabetical order. Zhuoxuan Jiang is the corresponding author.

Order of Authors: Authors are listed in order of decreasing contribution.

References


