Can We Verify Step by Step for Incorrect Answer Detection?

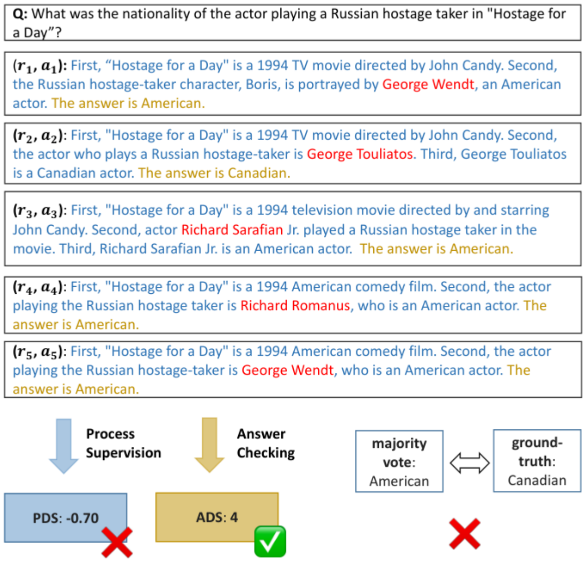
Xin Xu , **Shizhe Diao** , **Can Yang**[*] , **Yang Wang**
The Hong Kong University of Science and Technology
xxuca@connect.ust.hk, {sdiaoaa, macyang, yangwang}@ust.hk

Abstract

Chain-of-Thought (CoT) prompting has marked a significant advancement in enhancing the reasoning capabilities of large language models (LLMs). Previous studies have developed various extensions of CoT, which focus primarily on enhancing end-task performance. In addition, there has been research on assessing the quality of reasoning chains in CoT. This raises an intriguing question: Is it possible to predict the accuracy of LLM outputs by scrutinizing the reasoning chains they generate? To answer this research question, we introduce a benchmark, R2PE, designed specifically to explore the relationship between reasoning chains and performance in various reasoning tasks spanning five different domains. This benchmark aims to measure the falsehood of the final output of LLMs based on the reasoning steps. To make full use of information in multiple reasoning chains, we propose the process discernibility score (PDS) framework that beats the answer-checking baseline by a large margin. Concretely, this resulted in an average of 5.1% increase in the F1 score and 2.97% improvement in AUC-PR across all 45 subsets within R2PE. We further demonstrate our PDS's efficacy in advancing open-domain QA accuracy. Codes and data are available at https://github.com/XinXU-USTC/R2PE.git. For further details on the appendix, please refer to https://arxiv.org/abs/2402.10528.

1 Introduction

Recent development in large language models (LLMs) [OpenAI, 2023; Bubeck *et al.*, 2023; Yang *et al.*, 2024a] has showcased their remarkable aptitude for tackling diverse downstream tasks. Given several demonstrations with reasoning steps, LLMs exhibit a formidable capability to address reasoning tasks, commonly referred to as chain-of-thought (CoT) [Wei *et al.*, 2022]. There have been significant advancements in improving the reasoning abilities of LLMs in terms of end-task performance, such as intricate math word reasoning [Shum *et al.*, 2023; Diao *et al.*, 2023; Zheng *et al.*, 2023; Yu *et al.*, 2023; Xu *et al.*, 2024; Yang *et al.*, 2024b], and tasks that involve extensive search and tactical planning [Yao *et al.*, 2023]. Another line of research lies in the analysis of the reasoning steps themselves with or without human-annotated ones [Clinciu *et al.*, 2021; Prasad *et al.*, 2023; Xia *et al.*, 2024].



Q: What was the nationality of the actor playing a Russian hostage taker in "Hostage for a Day"?

(r_1, a_1): First, "Hostage for a Day" is a 1994 TV movie directed by John Candy. Second, the Russian hostage-taker character, Boris, is portrayed by George Wendt, an American actor. The answer is American.

(r_2, a_2): First, "Hostage for a Day" is a 1994 TV movie directed by John Candy. Second, the actor who plays a Russian hostage-taker is George Touliatos. Third, George Touliatos is a Canadian actor. The answer is Canadian.

(r_3, a_3): First, "Hostage for a Day" is a 1994 television movie directed by and starring John Candy. Second, actor Richard Sarafian Jr. played a Russian hostage taker in the movie. Third, Richard Sarafian Jr. is an American actor. The answer is American.

(r_4, a_4): First, "Hostage for a Day" is a 1994 American comedy film. Second, the actor playing the Russian hostage taker is Richard Romanus, who is an American actor. The answer is American.

(r_5, a_5): First, "Hostage for a Day" is a 1994 American comedy film. Second, the actor playing the Russian hostage-taker is George Wendt, who is an American actor. The answer is American.

Figure 1: An example from HotpotQA that GPT-4 outputs highly consistent but wrong answers. In this particular example, our PDS can detect conflicting information about the actor (colored by red) and predicts that the answer will be incorrect because it is less than zero, while ADS predicts the answer to be correct because it is greater than 2.5.

[Wei *et al.*, 2022; Ye and Durrett, 2022] have manually inspected whether reasoning steps align with the correctness of ultimate answers in both correct and incorrect answer groups through case studies. Moreover, [Prasad *et al.*, 2023; He *et al.*, 2023] find that elevating the quality of the rationales could potentially enhance task performance. Since aggregate task performance is a summation of the accuracy of final answers across individual instances, these observations offer an initial qualitative scrutiny of the reasoning chains and their impact on the accuracy of the final predictions. Despite these insights, there remains a shortage of quantitative proof to sub-

[*] Corresponding author.

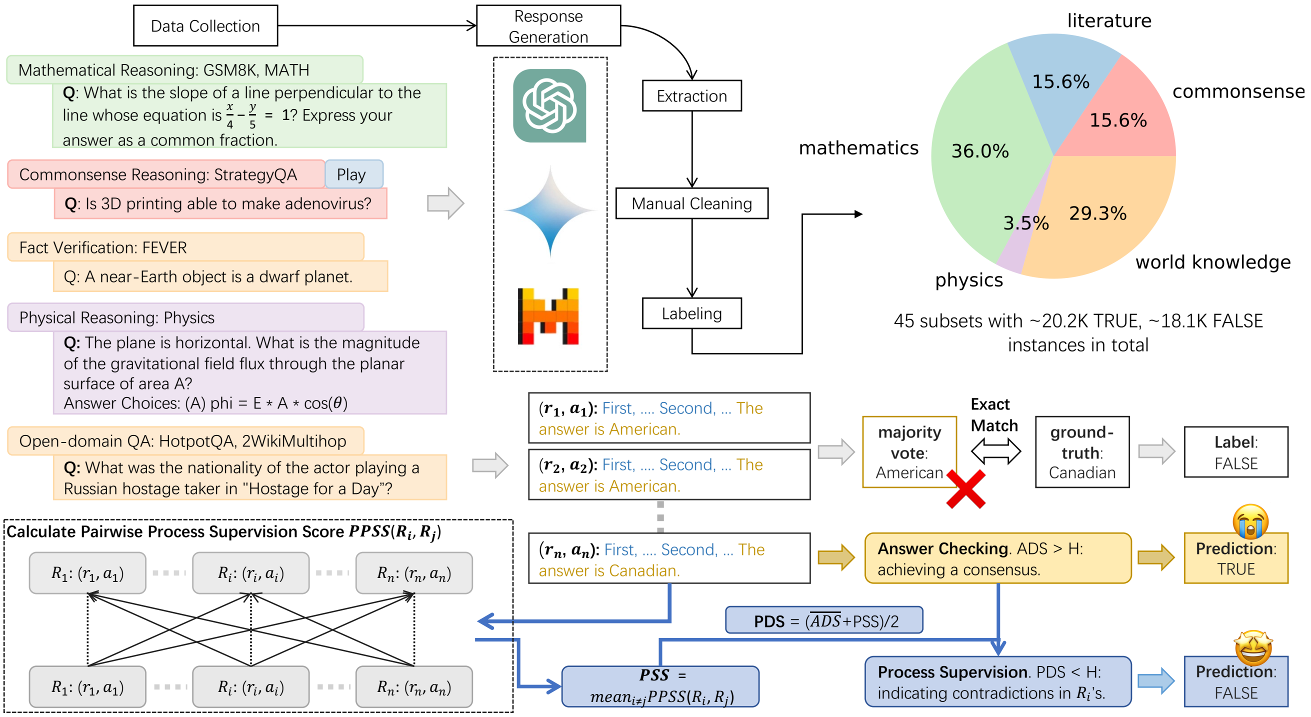

Data Collection
Response Generation
Extraction
Manual Cleaning
Labeling
Mathematical Reasoning: GSM8K, MATH
Q: What is the slope of a line perpendicular to the line whose equation is $\frac{x}{4} - \frac{y}{5} = 1$? Express your answer as a common fraction.
Commonsense Reasoning: StrategyQA
Play
Q: Is 3D printing able to make adenovirus?
Fact Verification: FEVER
Q: A near-Earth object is a dwarf planet.
Physical Reasoning: Physics
Q: The plane is horizontal. What is the magnitude of the gravitational field flux through the planar surface of area A?
Answer Choices: (A) phi = E * A * cos(θ)
Open-domain QA: HotpotQA, 2WikiMultihop
Q: What was the nationality of the actor playing a Russian hostage taker in "Hostage for a Day"?
literature
commonsense
mathematics
physics
world knowledge
15.6%
15.6%
36.0%
3.5%
29.3%
45 subsets with ~20.2K TRUE, ~18.1K FALSE instances in total
(r_1, a_1): First, ... Second, ... The answer is American.
(r_2, a_2): First, ... Second, ... The answer is American.
(r_n, a_n): First, ... Second, ... The answer is Canadian.
majority vote: American
Exact Match
ground-truth: Canadian
Label: FALSE
Answer Checking. ADS > H: achieving a consensus.
Prediction: TRUE
Calculate Pairwise Process Supervision Score $PPSS(R_i, R_j)$
$R_1: (r_1, a_1)$
$R_i: (r_i, a_i)$
$R_n: (r_n, a_n)$
$R_1: (r_1, a_1)$
$R_i: (r_i, a_i)$
$R_n: (r_n, a_n)$
PDS = $(\overline{ADS}$+PSS)/2
$PSS = mean_{i \neq j} PPSS(R_i, R_j)$
Process Supervision. PDS < H: indicating contradictions in R_i's.
Prediction: FALSE

Figure 2: An overview of R2PE benchmark and PDS framework. The construction of R2PE includes 5 stages: data collection, response generation, extraction, manual cleaning, and labeling. Sourced from 8 distinct datasets covering a variety of task types and derived from 6 different LLMs, R2PE comprises 45 subsets, featuring approximately 20.2K TRUE and 18.1K FALSE instances across 5 domains. The objective is to establish a discernibility score that accurately indicates the veracity of answers. PDS adopts answer checking and process supervision to detect all potential discrepancies among different rationales, which beats ADS that focuses merely on the answer consensus.

stantiate whether the evaluation of reasoning chains can reliably affirm the validity of the final outcomes. To bridge this gap, we introduce the **R2PE** (**R**elationship of **R**ationales to **P**erformance **E**valuation) benchmark, a test bed designed to quantitatively investigate this question.

The susceptibility of LLMs to generate incorrect information has been underscored by [OpenAI-Blog, 2022; Zhao *et al.*, 2023]. Meanwhile, [Ye and Durrett, 2022] associate false predictions with nonfactual explanations. Furthermore, [Wei *et al.*, 2022] have demonstrated that, even in arithmetic reasoning tasks, incorrect rationales can occasionally yield correct outcomes, and a variety of errors may occur at intermediate reasoning steps. To perform a qualitative evaluation of rationales to validate final predictions, the R2PE benchmark is established to integrate a diverse spectrum of reasoning tasks, covering mathematical, commonsense, physical, and textual reasoning (including fact verification and open-domain question answering). These tasks extend across domains such as mathematics, common knowledge, physics, literature, and general world knowledge. Responses are collected from six distinct LLMs to promote a broad generality of the findings. The characteristics and the creation steps of the R2PE benchmark are illustrated in Figure 2.

Each evaluated question or claim begins with the generation of multiple reasoning chains, leading to an answer aggregation based on majority voting [Wang *et al.*, 2022] after extraction and manual cleaning. Once the final outcomes are labeled as true or false using an exact match, our goal is to derive a discernibility score (DS), which is intended as an indicator of the credibility of the final answer. To fully exploit the information contained in all reasoning chains, we present a process discernibility score (PDS) that substantially exceeds the answer discernibility score (ADS) baseline that counts the number of the same answers. As shown in Figure 1, PDS can detect conflicting information about the actor and predict the incorrectness of the final answer. Figure 2 presents a succinct overview of both our benchmark and our method.

In summary, our contributions are as follows:

- We propose R2PE, the first benchmark that quantitatively assesses the relationship between reasoning chains and end-task performance across a spectrum of reasoning tasks, multiple domains, and an array of LLMs.
- We introduce a process discernibility score (PDS) framework that aggregates the information in different reasoning chains for CoT verification.
- Comprehensive experiments of PDS reveal its superiority over the answer discernibility score (ADS) in predicting final answer correctness, leading to consistent increases in the F1 score and AUC-PR.
- Further experiments show the effectiveness of our approach in improving open domain question answering

performance when combined with verify-and-edit [Zhao *et al.*, 2023].

2 R2PE Benchmark

R2PE serves as a comprehensive platform for verifying LLM-generated answers in CoT reasoning, which is meticulously constructed with the following critical characteristics: (*i*) Assessability: every instance within R2PE can be verifiable as true or false based on a certain criterion. (*ii*) Task Diversity: The benchmark should encompass a wide range of reasoning datasets, featuring various answer formats, and spanning different task categories across multiple domains. (*iii*) Generalizability across LLMs: Responses should be elicited using different LLMs to ensure broad applicability. (*iv*) High Quality: To minimize instances where correct answers are inaccurately labeled as false due to extraction failures, it is crucial that answers are precisely extracted from responses.

2.1 Construction Process

As shown in Figure 2, the creation of R2PE consists of the following five steps:

Data Collection. we utilize a total of eight datasets including benchmarks for mathematical reasoning such as GSM8K [Cobbe *et al.*, 2021] and MATH [Hendrycks *et al.*, 2021], common sense reasoning tasks like StrategyQA [Geva *et al.*, 2021] and Play dialogue from BIG-bench collections [Srivastava *et al.*, 2022], physical reasoning tasks (Physics from BIG-bench collections [Srivastava *et al.*, 2022]); fact verification (FEVER [Thorne *et al.*, 2018]) and open-domain question answering (HotpotQA [Yang *et al.*, 2018], 2WikiMultihop [Ho *et al.*, 2020]). Each selected dataset meets the criterion of assessability through verifiable answers. These tasks not only vary in answer formats, including numerical, yes/no, multiple choice, and free form, but also span an array of domains such as mathematics, commonsense, literature, physics, and general world knowledge (see Appendix A.1).

Response Generation. For each question (or claim in the FEVER dataset) Q, the LLM is prompted using CoT to produce n responses $(R_1, R_2, ..., R_n)$, respectively. The full prompts are given in Appendix A.3. The responses are aggregated from six distinct LLMs to ensure the generalizability: text-davinci-003, GPT-3.5-turbo [OpenAI-Blog, 2022], its instruct-trained variant GPT-3.5-turbo-instruct, Gemini Pro, Mixtral-8x7b, and mistral-medium. A concise exposition along with detailed settings, is deferred to Appendix A.2.

Extraction. Rationale r_i and the corresponding anwser a_i need to be isolated from each response R_i. During the pilot extraction period, we use the prompt cue ”The answer is” as a delimiter to segregate responses. We find that almost all LLMs yielded some outputs that did not conform to the expected answer format. Given the varying responses styles across different LLMs, the answer trigger words are identified for different LLMs and datasets after observing the original responses to facilitate extraction (see Appendix A.3). This strategy effectively segments most of the responses.

Manual Cleaning. To maintain a high-quality benchmark, manual inspection and cleaning are performed to handle unusual cases. For atypical responses that deviate from the recognized patterns, we either manually separate the response to derive the answer or assign a special marker when separation is not feasible. A detailed description of the extraction and manual cleaning procedure is provided in Appendix A.3.

Labeling. Upon completion of extraction and manual cleaning,$R = \{R_1 = (r_1, a_1), ..., R_n = (r_n, a_n)\}$ are collected for each question Q. Then we can aggregate the outputs to obtain the final answer by the majority votes: $a = argmax_{a_i} freq(a_i)$, where $freq(a_i)$ denotes the frequency in which a_i appears. The final result a of each question Q is then compared with its ground truth; A match results in a ”TRUE” label, while a mismatch is assigned as ”FALSE”.

2.2 R2PE Overview

Our R2PE benchmark comprises a diverse collection of data derived from six LLMs on eight reasoning tasks. These tasks encompass a variety of types, answer formats, and domains, making the benchmark rich and comprehensive. In total, it includes approximately 38.3K instances with around 20.2K labeled as TRUE and 18.1K as FALSE. The structure of R2PE allows for the organization of instances into 45 distinct subsets based on the original dataset and the LLM used to generate responses. Hereinafter, subsets are denoted as (dataset name from LLM name), with detailed statistics and concrete examples available in Appendix B.

Each instance e within R2PE incorporates a series of structured data fields: question or claim Q, the associated dataset name, the queried LLM name, five responses alongside their rationales and answers $R = \{R_1 = (r_1, a_1), R_2 = (r_2, a_2), ..., R_5 = (r_5, a_5)\}$, the final output a, the ground-truth answer of question Q, and the label $L \in \{T, F\}$ to indicate whether the generated answer a matches the ground truth. Detailed explications of the constituent data fields and illustrative examples are presented in Appendix B.

We aim to predict the correctness of final outcomes based on the intermediate reasoning steps. The response set R serves as an input for predicting the label L for every instance e. We introduce a numerical criterion, the discernibility score (DS), to encapsulate the quality of R.A low DS might suggest a potential mismatch between the final output a and the ground-truth answer. Hence, we will classify the example as false: $\hat{L} = F$, if its DS falls below a certain threshold H.

3 Process Discernibility Score

3.1 The Fallacy of Answer Agreement

From [Wang *et al.*, 2022; Zhao *et al.*, 2023], the degree of consensus among answers, is posited as a diagnostic tool to gauge the instances in which LLMs may make wrong predictions. We refer to the answer agreement number as Answer Discernibility Score (ADS) hereinafter. A high ADS often suggests a high likelihood that the proposed answer is correct. If the ADS is below that midpoint (i.e., the threshold $H = n/2$), $\hat{L} = F$ is predicted.

However, LLMs may generate substantially consistent but incorrect answers in reasoning tasks. Figure 1 showcases one such example, where the LLM is queried about the nationality of an actor from ”Hostage for a Day”. The LLM gives the answer ”American” four times and ”Canadian” once. Here, the ADS stands at 4, which erroneously points to a seemingly

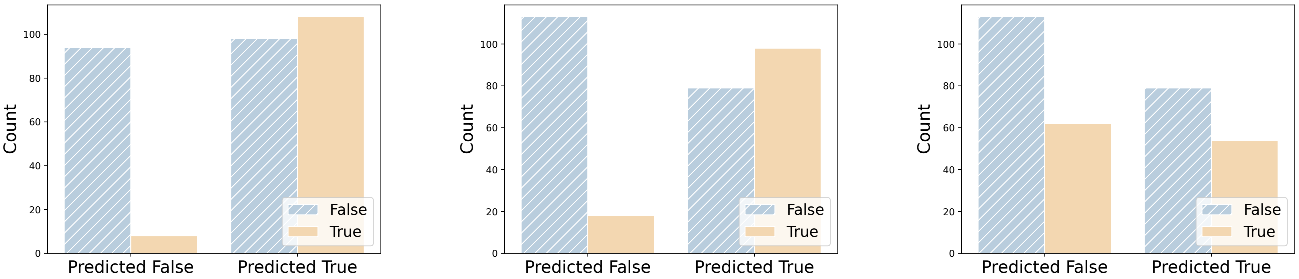


Figure 3: Results on the subset (HotpotQA from GPT-4): ADS (left), PDS (middle), and PDS-ADS (right). ADS has high precision: large answer agreement does not always guarantee accurate predictions. PDS yields desirable outcomes, while PDS-ADS has poor precision.

accurate but ultimately false consensus. This example serves to warn against overreliance on ADS as the sole indicator of the veracity of the final answer in reasoning tasks.

3.2 Process Discernibility Score

In our R2PE benchmark, each response R_i constitutes a rationale r_i paired with an answer a_i. The ADS, however, limits its analysis to the answers $(a_1, ..., a_n)$, which takes up only a small proportion of the information contained in R. Inspired by [Yoran *et al.*, 2023], a DS that makes use of rationales would be preferable.

To quantify the similarity between two responses, we employ Pairwise Process Supervision Score PPSS(R_k, R_l) to measure the degree to which the content of R_l is both included in and opposes that of R_k. PPSS is confined within the interval (-1, 1), where a positive score indicates a higher level of similarity relative to contradiction. For a set of n responses, we can calculate the average of PPSS across all ordered response pairs, culminating in the Process Supervision Score (PSS): PSS $=$ mean$_{i\neq j}$PPSS(R_i, R_j). This allows PSS to evaluate the information across all responses. To underscore the final answers, we get

$$\text{PDS} = (\overline{\text{ADS}} + \text{PSS})/2, \tag{1}$$

wherein $\overline{\text{ADS}} = 2(\text{ADS} - n/2)/n$ represents the normalized ADS, adjusted to align with PSS's range. The overview of PDS is shown in Figure 2.

Subsequently, we classify instances where the PDS falls below the threshold H as $\hat{L} = F$, whereas all remaining instances are designated as $\hat{L} = T$.

Compared to ADS, our PDS is capable of overseeing reasoning processes as well as doing implicit answer checking ($\overline{\text{ADS}}$ term in Equation (1)). As depicted in Figure 1, the PDS can detect contradictive information about the name of the actor playing a Russian hostage taker in "Hostage for a Day" among multiple reasoning chains (highlighted by red) and give a correct label $\hat{L} = F$, while the ADS only assesses the agreements among the answers.

3.3 Experimental Setup

We will detail the experimental setup in this section.

Dataset. All subsets of the R2PE benchmark.

Baselines. From [Wang *et al.*, 2022; Zhao *et al.*, 2023], ADS can act as the baseline in our setting. We also include SelfCheckGPT [Manakul *et al.*, 2023] and HaloCheck [Elaraby *et al.*, 2023] as our baselines.

Metrics. We will employ the F1 score and AUC-PR as our evaluation metrics. As we aim to detect potential incorrect answers, False labels are treated as positive. Therefore, true positives are examples with $L = F$ and $\hat{L} = F$. We do not compare with SelfCheckGPT and HaloCheck in terms of F1 score, as they do not provide thresholds for prediction.

PDS Implementation. For PPSS$(\cdot, \cdot)$, we adopt the SAUMMAC zero-shot model [Laban *et al.*, 2022],which is a consistency detection method for text summarization that leverages the out-of-the-box natural language inference model to compute pairwise entailment score. As suggested by [Laban *et al.*, 2022], probabilities of sentence-level entailment and contradiction are used. To detect all potential inconsistencies in any sentence, we use min operation to aggregate across sentence pairs:

$$\text{PPSS}(R_k, R_l) = \min_{j} \max_{i} \big(ent(a_i, b_j) \quad - con(a_i, b_j)\big), \tag{2}$$

where a_i, b_j are sentences of two responses R_k, R_j correspondingly, ent represents the probability of entailment, con is the probability of contradiction, the min is taken over all sentences of R_j, and max is taken over all sentences of R_k. Note that the PPSS metric lacks symmetry; however, this is not a problem in our implementation because the PSS calculation averages across all possible ordered response pairs.

Threshold Selection. As ADS $\in \{0, 1, 2, ..., 5\}$, threshold H is set to be the midpoint $H = 2.5$. Another reason for this selection is that ADS > 2.5 means the majority agrees, which aligns the choice in [Zhao *et al.*, 2023]. The range of this PPSS spans in $(-1, 1)$. A negative value implies the potential presence of conflicting information. Consequently, it is reasonable to set the threshold H for the PDS to be 0. Following [Li *et al.*, 2023], the threshold of ADS is raised to 4.5 for discriminative tasks (yes/no, multiple-choice questions), and the PDS threshold is adjusted to 0.4 according to Equation (1). Our choice of threshold H is not task-specific. We regard this as a benefit because held-out data for threshold tuning may not always be available in the real scenario. A detailed discussion is given in Appendix C.3.

Dataset	Method	GPT3	GPT-instruct	GPT-3.5	Gemini	Mixtral	mistral	avg
GSM8K	ADS	66.59	53.63	42.83	52.05	58.63	50.14	-
	PDS	**69.30**	**56.65**	**52.50**	**55.04**	**62.50**	**53.26**	+4.23
MATH	ADS	81.33	76.72	70.40	77.92	81.19	77.56	-
	PDS	**86.55**	**80.38**	**74.93**	**78.95**	**83.85**	**81.33**	+3.48
StrategyQA	ADS	36.79	59.06	54.28	52.54	60.39	62.59	-
	PDS	**52.21**	**59.14**	**57.51**	**56.76**	**63.14**	**63.83**	+4.49
Play	ADS	50.22	56.64	54.28	53.07	71.12	56.32	-
	PDS	**55.59**	**59.00**	**56.16**	**54.09**	**72.02**	**58.78**	+2.30
Physics	ADS	48.70	52.46	56.52	65.86	61.54	63.33	-
	PDS	**52.31**	**55.56**	**58.65**	**88.44**	**66.90**	**66.01**	+6.58
FEVER	ADS	50.92	55.97	49.14	63.56	64.58	-	-
	PDS	**59.64**	**60.68**	**53.01**	**63.64**	**65.72**	-	+3.70
HotpotQA	ADS	72.78	71.52	63.95	81.46	74.24	-	-
	PDS	**85.71**	**78.84**	**70.25**	**85.65**	**78.66**	-	+7.00
2WikiMultihop	ADS	69.26	69.75	42.67	62.83	69.36	-	-
	PDS	**78.65**	**76.51**	**57.14**	**71.76**	**70.81**	-	+9.80
avg	-	+7.92	+3.88	+5.76	+5.63	+2.82	+2.65	+5.10

Table 1: PDS consistently outperforms ADS across all subsets in our R2PE benchmark in F1 scores (in %). The abbreviations GPT-3, GPT-instruct, GPT-3.5, Gemini, Mixtral, mistral correspond respectively to the LLMs text-davinci-003, GPT-3.5-turbo-instruct, GPT-3.5-turbo, Gemini Pro, Mixtral-8x7b, and mistral-medium. Results on FEVER, HotpotQA, and 2WikiMultihop from mistral-medium are not reported, as discussed in Section 2. The last row and column are the average improvement of PDS over ADS across datasets and LLMs respectively. The results of HotpotQA and 2WikiMultihop from GPT-3.5 are replaced with those from GPT-4 (see Appendix A.2).

Dataset	Method	GPT3	GPT-instruct	GPT-3.5	Gemini	Mixtral	mistral	avg
GSM8K	SelfCheckGPT	53.87	35.67	30.35	27.07	40.30	29.29	36.09
	HaloCheck	47.89	34.52	28.45	21.65	39.61	25.03	32.86
	ADS	70.79	55.83	54.58	52.59	63.18	48.15	57.52
	PDS	**74.14**	**61.02**	**55.37**	**56.23**	**68.37**	**51.86**	**61.17**
MATH	SelfCheckGPT	89.04	74.55	55.60	73.46	85.51	82.96	76.85
	HaloCheck	88.11	76.46	58.26	80.08	82.57	81.77	77.87
	ADS	92.82	85.50	78.28	87.81	90.13	88.05	87.10
	PDS	**93.83**	**86.77**	**80.84**	**89.49**	**91.82**	**89.66**	**88.73**
StrategyQA	SelfCheckGPT	50.86	41.74	42.72	47.37	61.67	58.73	50.52
	HaloCheck	37.02	40.94	42.48	35.93	59.80	41.20	42.90
	ADS	49.56	51.42	49.36	49.85	59.55	56.27	52.87
	PDS	**49.97**	**54.49**	**52.87**	**51.18**	**62.08**	**67.22**	**56.30**
Play	SelfCheckGPT	40.63	45.60	45.37	37.22	63.70	45.37	46.33
	HaloCheck	37.02	40.94	42.48	35.93	59.80	41.20	42.90
	ADS	44.37	48.93	49.90	46.00	72.68	50.44	52.05
	PDS	**46.14**	48.01	**51.31**	**47.66**	**72.84**	**51.86**	**52.97**
Physics	SelfCheckGPT	40.63	45.60	45.37	37.22	63.70	45.37	46.33
	HaloCheck	37.02	40.94	42.48	35.93	59.80	41.20	42.90
	ADS	39.94	40.60	48.63	79.10	53.05	52.58	52.32
	PDS	**46.85**	**48.58**	**61.31**	78.50	**56.99**	**60.66**	**58.82**
FEVER	SelfCheckGPT	55.08	49.36	44.86	51.63	62.28	-	52.64
	HaloCheck	55.44	47.00	50.17	47.56	60.36	-	52.11
	ADS	57.02	49.72	48.57	54.02	67.31	-	55.33
	PDS	**62.19**	**51.41**	**49.00**	**55.43**	66.91	-	**57.03**
HotpotQA	SelfCheckGPT	85.69	80.20	71.37	84.48	52.60	-	74.87
	HaloCheck	88.06	82.18	85.20	82.51	48.20	-	77.23
	ADS	89.71	79.14	83.99	90.15	74.89	-	83.58
	PDS	**92.61**	**83.21**	**86.21**	**91.47**	**79.41**	-	**86.58**
2WikiMultihop	SelfCheckGPT	85.69	80.20	71.37	84.48	52.60	-	74.87
	HaloCheck	88.06	82.18	85.20	82.51	48.20	-	77.23
	ADS	83.26	75.52	48.12	79.37	59.51	-	69.16
	PDS	**86.72**	**79.80**	**54.37**	**80.20**	58.94	-	**72.01**

Table 2: PDS consistently outperforms ADS across almost all subsets in our R2PE benchmark in AUC-PR (in %).

Method	GPT3	GPT-Instruct	GPT-4	Gemini	Mixtral
ADS	72.78	71.52	63.95	81.46	74.24
PDS	**85.71**	**78.84**	**70.25**	**85.65**	**78.66**
PDS - ADS	82.00	78.16	60.57	82.35	74.24
PDS w/o ans	78.44	**78.84**	63.41	84.79	74.19
PDS-avg	77.49	78.81	62.72	84.86	74.29
PDS-Halocheck	77.49	78.11	62.72	84.99	75.32
PDS-selfcheckNLI	45.77	51.01	33.62	66.87	60.25

Table 3: Results of PDS on HotpotQA. Metrics are F1 (%).

Method	GPT3	GPT-Instruct	GPT-4	Gemini	Mixtral
ADS	69.26	69.75	42.67	62.83	69.36
PDS	**78.65**	**76.51**	**57.14**	**71.76**	**70.81**
PDS - ADS	76.97	73.35	55.00	70.14	69.36
PDS w/o ans	76.12	75.56	50.19	67.38	68.44
PDS-avg	76.95	75.34	51.06	67.23	68.87
PDS-HaloCheck	76.66	75.08	52.30	68.39	68.78
PDS-SelfCheckNLI	52.61	47.65	25.64	42.89	55.70

Table 4: Results of PDS on 2WikiMultihop. Metrics are F1 (%).

4 Results and Analysis

4.1 PDS Substantially Outperforms All Baselines

As shown in Table 1, Across all 45 subsets in our R2PE, PDS consistently improves ADS performance, yielding an average improvement of 5.10% in terms of F1. Although ADS demonstrates greater precision, it suffers from significantly lower recall, indicating its propensity to overlook numerous examples with $L = F$. In contrast, PDS adeptly balances precision and recall, culminating in superior overall performance relative to ADS (see Table 1, 2, and Appendix C.2). From Table 2, PDS improves ADS in terms of AUC-PR in the majority of cases and yields comparable results (less than 1% decrease) in the remaining ones. On average, PDS results in 3.01% absolute improvement in AUC-PR. Notably, ADS is a fairly strong baseline compared with SelfCheckGPT and HaloCheck in our setting, and PDS can further enhance the performance of ADS. We believe that the performance of SelfCheckGPT [Manakul *et al.*, 2023] and HaloCheck [Elaraby *et al.*, 2023], which lags behind ADS, can be attributed to their operation in different settings and the lack of explicit verification of the majority voting proportion of the final answer, which highlights the importance of jointly considering both the reasoning process and the final answer to validate CoT, demonstrating the superiority of PDS. For discriminative tasks, ADS has relatively low F1 scores and AUC-PR. This may be attributed to the fixed set of answers (e.g., yes or no) in these tasks, which inherently exhibit higher consistency, making verification based solely on answer agreement challenging. In contrast, PDS can relieve this issue by evaluating reasoning processes.

PDS demonstrates a stronger capability to verify the accuracy of the predictions for free-form questions than ADS, with notable average improvements of 7.00% in HotpotQA and 9.80% in 2WikiMultihop. This superior performance can be attributed to the process-oriented verification approach employed by PDS. By addressing the challenges associated with indirectly assessing answer consistency in free-form questions, PDS proves to be more effective in ensuring accuracy. It is important to note that the average improvement metrics provided in Table 1 serve primarily as an indicator of the extent to which PDS enhances performance over ADS. However, a direct comparison of F1 scores and AUC-PR between different subsets within R2PE is not recommended. This is because the efficacy of the F1 score as a comparative metric is limited by variations in the ratio of TRUE and FALSE instances across different subsets.

To meticulously evaluate the efficacy of the PDS, we delve into the (HotpotQA from GPT-4) subset, partitioning it into two sets: one where answers fail to reach a consensus greater than half, and its counterpart. We then label the examples with an ADS less than the threshold H as $\hat{L} = F$, and all the others as $\hat{L} = T$. As shown in Figure 3 (left), while the ADS proficiently segregates the examples with $L = T$, its discernment for the ”False” category is less acute. On the contrary, the PDS (Figure 3 middle) strikes a better balance between precision and recall. Eliminating the implicit answer verification component in Equation (1) results in diminished precision (Figure 3 right, discussed in Section 4.2).

4.2 Ablation Study

To understand the contribution of PDS' components, we have carried out an extensive ablation study involving the following variants (See Appendix C.4):

PDS-ADS. we remove the $\overline{ADS}$ in our Equation (1), and the alternative of PDS becomes $PDS - ADS = PSS$.

PDS w/o ans. The original PSS considers all $PPSS(R_i, R_j)$, where each response R_i equals a reasoning path r_i and an answer a_i. As the information from the answers a_i' is already used to obtain the final answer a, we can discard them and only compute the entailment score among rationale pairs $PPSS(r_i, r_j)$, that is, $PDS_{w/o\ ans} = (\overline{ADS} + \text{mean}_{i \neq j} PPSS(r_i, r_j))/2$.

PDS-avg. We can change the aggregation operation min of $PPSS(\cdot,\cdot)$ (Equation (2)) by taking the average. Then $PPSS(A, B) = \text{avg}_j \max_i(ent(a_i, b_j) - con(a_i, b_j))$.

PDS-HaloCheck;**PDS-SelfCheckNLI**. HaloCheck [Elaraby *et al.*, 2023] and SelfCheckNLI [Manakul *et al.*, 2023] can be alternatives to PSS, denoted by PDS-halocheck and PDS-selfcheckNLI, respectively (see Appendix C.5).

As shown in Table 3 and 4, our PDS implementation gains the best results.

4.3 PDS for Improving Downstream Performance

VE [Zhao *et al.*, 2023] post-edits reasoning process using external knowledge (see Appendix D). VE first checks the answer agreement and then edits the reasoning chains based on a retriever if $ADS < H$. That is to say, VE selects potential incorrect samples by ADS. Our PDS can be incorporated into the VE framework, dubbed ”VE + PDS”.

Experiments are carried out on (HotpotQA from GPT-4) and (2WikiMultihop from GPT-4), utilizing ground-truth supporting contexts along with distractor paragraphs as the retrieved documents to negate the influence of the retrieval system. Our baseline is VE, and the standard prompting and CoT-SC are also included as references.

Figure 4 illustrates that integrating our method with the VE framework leads to enhanced accuracy for both datasets.

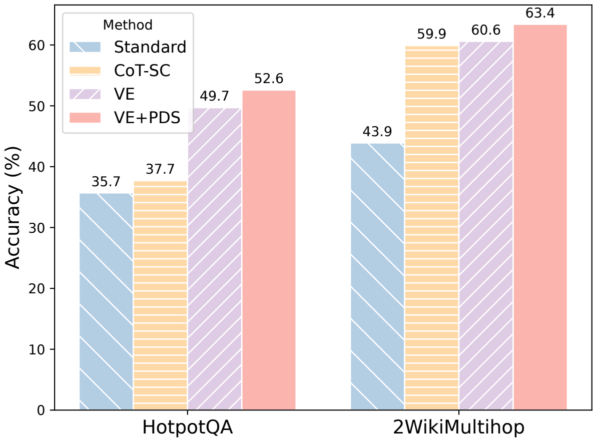


Figure 4: PDS can be integrated with verify-and-edit to further improve accuracy on open-domain QA.

Type of Errors	Type I	Type II	Type III	Type IV
Number of Cases	14	14	18	7
Percentage (in %)	26.4	26.4	33.9	13.2

Table 5: Distribution of Error Types

Specifically, VE+PDS leads to a notable 2.9% absolute increase in accuracy for HotpotQA, and 2.8% for 2WikiMultihop. Given that the retrieval system in our experiments is assumed to be near-perfection, the observed improvements in performance are attributed to the more effective answer verification through the substitution of ADS with PDS. This success is likely a consequence of the PDS's capability to supervise processes across multiple reasoning chains, leading to better selection of all potential incorrect predictions.

4.4 Error Analysis

We also conduct an error analysis on HotpotQA using a GPT-4 subset of R2PE. Specifically, we find 68 cases where PDS correctly identifies the falsehood of answers that ADS does not. In these cases, all the labels are ”False” and the answers are consistent, meaning ADS could not verify the answers due to its reliance on answer consistency alone. In contrast, PDS is able to identify potential inconsistencies among different reasoning chains, even when the answers are consistent. However, we also find 53 cases where the labels are ”True” and the answers are consistent, where ADS succeeds but PDS fails. After a careful manual inspection, we classify these cases into the following categories: I. Conflicting information among chains despite consistent and correct answers. II. Excessive information that is correct in some chains, leads to inconsistency among chains. III. Imperfections of the NLI models even when there is no conflicting information. IV. Different solutions with diverging chains lead to the same correct answers. We present the number of cases in each category in Table 5. See discussion in Appendix E.

5 Related Work

Extensions of CoT. CoT [Wei *et al.*, 2022] improves LLM performance by decomposing reasoning into steps via few-shot examples. Refinements include sampling-based decoding [Wang *et al.*, 2022], automated exemplar selection [Kojima *et al.*, 2022; Zhang *et al.*, 2022], and hybrid strategies [Zou *et al.*, 2023]. Task-specific adaptations like PHP [Zheng *et al.*, 2023] for mathematical reasoning and verify-and-edit (VE) [Zhao *et al.*, 2023] focus on improving reasoning accuracy. While much work emphasizes end-task metrics, our study explores reasoning quality-outcome linkage, verifying predictions through chain analysis. We will also extend PDS to Long-CoT LLMs [Jaech *et al.*, 2024], which benefit from sampling-based verification [Wen *et al.*, 2025], and propose potential extensions to test-time scaling methods [Wu *et al.*, 2025] via chain consistency analysis.

Reasoning Chain Quality Evaluation. Rationale analysis is key for understanding AI performance and limitations[Golovneva *et al.*, 2022; Prasad *et al.*, 2023; He *et al.*, 2023]. Rationale evaluation methods are either reference-based, comparing against a gold standard [Clinciu *et al.*, 2021; Saparov and He, 2022], or reference-free, using metrics like ROSCOE [Golovneva *et al.*, 2022]. While they link high-quality reasoning to better task performance, the direct impact of reasoning quality on prediction accuracy is under-researched. Our study fills this gap with a detailed quantitative analysis of this relationship. There is also research focused on evaluating not only the final predictions but also the reasoning steps in CoT [Xia *et al.*, 2024; Huang *et al.*, 2024; Xu *et al.*, 2025].

Uncertainty Estimation. A range of methods measure LLM uncertainty by token probabilities [Kuhn *et al.*, 2023; Malinin and Gales, 2020; Kadavath *et al.*, 2022]; however, they are inapplicable to R2PE, which does not furnish probabilities for responses. Moreover, they concentrate on short answer sequences, aligning more closely with the standard prompting. In contrast, our work verifies the veracity of CoT answers, which encompasses both the reasoning chains and answer sequences. Different from [Kuhn *et al.*, 2023], the entailment model in our work is used to measure the informational similarities between two responses.

6 Conclusion

To shed more light on the connection between the quality of the reasoning steps and the final outcome, we present R2PE, the first benchmark that quantitatively analyzes whether we can validate the answer veracity by evaluating the reasoning chains on a variety of reasoning tasks across five different domains. Six different LLMs are used in the creation of R2PE to guarantee the generalizability of our findings. We develop the PDS framework by integrating multiple rationales' information to predict the falsehood of the final predictions, which significantly boosts the performance of answer checking on our R2PE benchmark. Furthermore, our approach can be easily combined with the verify-and-edit framework to improve end-task performance.

Acknowledgements

The work was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (AoE/E-601/24-N)

References

[Bubeck *et al.*, 2023] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *ArXiv preprint*, abs/2303.12712, 2023.

[Clinciu *et al.*, 2021] Miruna-Adriana Clinciu, Arash Eshghi, and Helen Hastie. A study of automatic metrics for the evaluation of natural language explanations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2376–2387, Online, 2021. Association for Computational Linguistics.

[Cobbe *et al.*, 2021] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *ArXiv preprint*, abs/2110.14168, 2021.

[Diao *et al.*, 2023] Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. Active prompting with chain-of-thought for large language models. *ArXiv preprint*, abs/2302.12246, 2023.

[Elaraby *et al.*, 2023] Mohamed Elaraby, Mengyin Lu, Jacob Dunn, Xueying Zhang, Yu Wang, and Shizhu Liu. Halo: Estimation and reduction of hallucinations in open-source weak large language models. *ArXiv preprint*, abs/2308.11764, 2023.

[Geva *et al.*, 2021] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021.

[Golovneva *et al.*, 2022] Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. Roscoe: A suite of metrics for scoring step-by-step reasoning. *ArXiv preprint*, abs/2212.07919, 2022.

[He *et al.*, 2023] Hangfeng He, Hongming Zhang, and Dan Roth. Socreval: Large language models with the socratic method for reference-free reasoning evaluation. *ArXiv preprint*, abs/2310.00074, 2023.

[Hendrycks *et al.*, 2021] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *ArXiv preprint*, abs/2103.03874, 2021.

[Ho *et al.*, 2020] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics.

[Huang *et al.*, 2024] Zhen Huang, Zengzhi Wang, Shijie Xia, and Pengfei Liu. Olympicarena medal ranks: Who is the most intelligent ai so far? *arXiv preprint arXiv:2406.16772*, 2024.

[Jaech *et al.*, 2024] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[Kadavath *et al.*, 2022] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

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

[Kuhn *et al.*, 2023] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

[Laban *et al.*, 2022] Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177, 2022.

[Li *et al.*, 2023] Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Lidong Bing, Shafiq Joty, and Soujanya Poria. Chain of knowledge: A framework for grounding large language models with structured knowledge bases. *ArXiv preprint*, abs/2305.13269, 2023.

[Malinin and Gales, 2020] Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*, 2020.

[Manakul *et al.*, 2023] Potsawee Manakul, Adian Liusie, and Mark JF Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *ArXiv preprint*, abs/2303.08896, 2023.

[OpenAI-Blog, 2022] OpenAI-Blog. Chatgpt: Optimizing language models for dialogue. OpenAI Blog, 2022. [Online; accessed on 2023/12/8].

[OpenAI, 2023] OpenAI. Gpt-4 technical report. *ArXiv preprint*, abs/2303.08774, 2023.

[Prasad *et al.*, 2023] Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. Receval: Evaluating reasoning chains via correctness and informativeness. *ArXiv preprint*, abs/2304.10703, 2023.

[Saparov and He, 2022] Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *ArXiv preprint*, abs/2210.01240, 2022.

[Shum *et al.*, 2023] KaShun Shum, Shizhe Diao, and Tong Zhang. Automatic prompt augmentation and selection with chain-of-thought from labeled data. *ArXiv preprint*, abs/2302.12822, 2023.

[Srivastava *et al.*, 2022] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv preprint*, abs/2206.04615, 2022.

[Thorne *et al.*, 2018] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana, 2018. Association for Computational Linguistics.

[Wang *et al.*, 2022] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *ArXiv preprint*, abs/2203.11171, 2022.

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

[Wen *et al.*, 2025] Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, et al. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *arXiv preprint arXiv:2503.10460*, 2025.

[Wu *et al.*, 2025] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for llm problem-solving. In *The Thirteenth International Conference on Learning Representations*, 2025.

[Xia *et al.*, 2024] Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. Evaluating mathematical reasoning beyond accuracy. *arXiv preprint arXiv:2404.05692*, 2024.

[Xu *et al.*, 2024] Xin Xu, Tong Xiao, Zitong Chao, Zhenya Huang, Can Yang, and Yang Wang. Can llms solve longer math word problems better? *arXiv preprint arXiv:2405.14804*, 2024.

[Xu *et al.*, 2025] Xin Xu, Qiyun Xu, Tong Xiao, Tianhao Chen, Yuchen Yan, Jiaxin Zhang, Shizhe Diao, Can Yang, and Yang Wang. Ugphysics: A comprehensive benchmark for undergraduate physics reasoning with large language models. *arXiv preprint arXiv:2502.00334*, 2025.

[Yang *et al.*, 2018] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, 2018. Association for Computational Linguistics.

[Yang *et al.*, 2024a] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[Yang *et al.*, 2024b] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[Yao *et al.*, 2023] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *ArXiv preprint*, abs/2305.10601, 2023.

[Ye and Durrett, 2022] Xi Ye and Greg Durrett. The unreliability of explanations in few-shot prompting for textual reasoning. *Advances in neural information processing systems*, 35:30378–30392, 2022.

[Yoran *et al.*, 2023] Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. Answering questions by meta-reasoning over multiple chains of thought. *ArXiv preprint*, abs/2304.13007, 2023.

[Yu *et al.*, 2023] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

[Zhang *et al.*, 2022] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. *ArXiv preprint*, abs/2210.03493, 2022.

[Zhao *et al.*, 2023] Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. *ArXiv preprint*, abs/2305.03268, 2023.

[Zheng *et al.*, 2023] Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. Progressive-hint prompting improves reasoning in large language models. *ArXiv preprint*, abs/2304.09797, 2023.

[Zou *et al.*, 2023] Anni Zou, Zhuosheng Zhang, Hai Zhao, and Xiangru Tang. Meta-cot: Generalizable chain-of-thought prompting in mixed-task scenarios with large language models. *ArXiv preprint*, abs/2310.06692, 2023.