Are Reasoning Models More Prone to Hallucination?

Anonymous ACL submission

Abstract

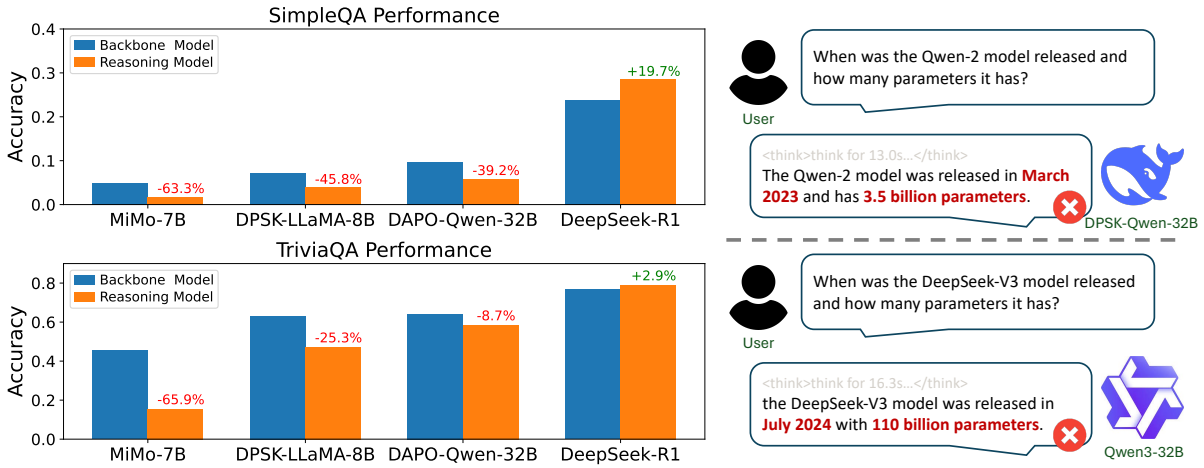
Large reasoning models (LRMs) show powerful performance in solving complex tasks with the help of long chain-of-thought (CoT) reasoning capability. However, whether they generalize the reasoning capability to help reduce hallucination in fact-seeking tasks remains unclear and debated. For example, DeepSeek-R1 reports increased performance on SimpleQA, a fact-seeking benchmark, while OpenAI-o3 observes even more severe hallucination. This discrepancy raises our research question: ***Are reasoning models more prone to hallucination?*** This paper addresses the question from three perspectives. (1) We first holistically evaluate the hallucination in LRMs, which reveals that LRMs undergo a full post-training pipeline with cold start supervised fine-tuning (SFT) and verifiable reward reinforcement learning (RL) generally alleviate their hallucination. In contrast, both SFT alone and RL training without cold start SFT introduce more nuanced hallucinations. (2) We further analyze behavior differences to explore how different post-training pipelines alters the hallucination in LRMs. We find two critical cognitive behaviors that directly affect the factuality of an LRM: ***Flaw Repetition***, where the surface-level reasoning attempts repeatedly follow the same underlying flawed logic, and ***Think-Answer Mismatch***, where the final answer fails to match the thinking process. (3) Moving further, we investigate the mechanism behind the hallucination of LRMs. We find that increased hallucination of LRMs is usually associated with the misalignment between model uncertainty and factual accuracy. Overall, we provide an initial understanding of the hallucination in LRMs.

1 Introduction

Large language models (LLMs) demonstrate their capability to solve complex tasks through long chain-of-thought (CoT) reasoning (Wei et al., 2022). Guided by this intuition, recent advancements in LLMs development introduce a specialized post-training phase aiming at further enhancing long-form reasoning capabilities. This stage often involves reinforcement learning (RL) with verifiable reward (Lambert et al., 2024; Guan et al., 2024; Luo et al., 2025) or supervised fine-tuning (SFT) on long-form reasoning data (Ye et al., 2025), giving birth to large reasoning models (LRMs), such as OpenAI-o1, o3, and o4-mini (Jaech et al., 2024; OpenAI, 2025), DeepSeek-R1 (Guo et al., 2025), GLM-Z1 (Hou et al., 2025; GLM, 2025), and Qwen-3 (Qwen, 2025; Yang et al., 2025).

While post-training is mainly conducted on formal reasoning tasks, *e.g.,* math reasoning, logic reasoning, and coding, whose answers are formally verifiable, it is widely believed that LRMs can generalize their reasoning abilities to non-formal tasks. It is expected that the long CoT reasoning also helps to reduce hallucination. For example, DeepSeek-R1 reports its improved performance on SimpleQA (Wei et al., 2024), a fact-seeking question answering benchmark, after post-training. On the contrary, OpenAI observes even severer hallucination on more powerful LRMs (OpenAI-o3 *v.s.* OpenAI-o1, Jaech et al., 2024). Thus, there remains a lack of systematic understanding regarding whether and how this form of reasoning contributes to more reliable factual inference. We demonstrate the varied performance changing trends for different LRMs after post-training in Figure 1.

In view of this gap, this paper aims to answer the following research question: *Are reasoning models more prone to hallucination?* Following OpenAI (2025), we first comprehensively experiment to quantify hallucination of differently developed LRMs on fact-seeking tasks, where the LRMs need to draw upon their parametric knowledge and to reason through the combination of multiple facts to reach the final answer. Then, we conduct behavior analysis to identify key cognitive behaviors of LRMs that directly affect hallucination. Finally,



(a) LRM Compared to Backbone Model on Factuality Bench

(b) LRM Hallucination Example

Figure 1: Comparison of factual accuracy between LRMs and their backbone counterparts, along with illustrative examples of LRM hallucinations. (a) Accuracy comparison on two fact-seeking benchmarks (SimpleQA and TriviaQA), with relative changes denoted on the bars. It shows that LRMs *seem* to suffer from factuality degradation compared to their backbone models, especially on SimpleQA, *except DeepSeek-R1*. (b) Examples of hallucinated answers from two LRMs: DeepSeek-Qwen-Distill-32B and Qwen3-32B provide incorrect responses to queries about model information. In reality, Qwen-2 was released in June 2024 with no 3.5 billion variant, and DeepSeek-V3 was released in December 2024 with 671 billion parameters.

we investigate the mechanistic for hallucination.

We first holistically evaluate the factuality on the widely used fact-seeking benchmarks—SimpleQA (Wei et al., 2024) and TriviaQA (Joshi et al., 2017)—across a wide range of LRMs and their non-reasoning counterparts. Strikingly, we find that not all LRMs benefit from their long CoT reasoning capabilities. Specifically, LRMs developed with SFT stage only or RL stage only even fall short of their non-reasoning backbone on factuality. In contrast, LRMs developed with both training stages, *i.e.,* reasoning capability cold start SFT and verifiable reward RL, show increased factuality from the initial non-reasoning model. However, several most recently developed LRMs only focus on distilling from strong reasoning model from carefully curated long CoT data. There are also attempts to build LRMs with AlphaGo-Zero (Silver et al., 2017) like RL without cold start. As these works mostly aim at improving the performance on formal reasoning tasks, they neglect to evaluate the performance on factuality of the final delivered LRMs. Thus, the results stem from our pilot experiments advocate the community to evaluate factuality, alongside other formal reasoning tasks, during the development of LRMs.

We further delve into the details to examine why LRMs developed with different post-training pipelines show different factuality performance. In particular, we characterize two critical cognitive behaviors that directly affect the factuality of an LRM: ***Flaw Repetition***, where the LRM repeatedly attempts different surface-level reasoning paths, yet consistently adheres to the same underlying, and often flawed, line of thought; and ***Think-Answer Mismatch***, where the LRM fails to provide an answer that faithfully matches its reasoning process. Our analysis reveals that both RL-only and SFT-only training process tend to encourage the LRM to exhaustively explore the reasoning space, leading the LRM to get trapped in repetitive loops. In the meanwhile, the RL-only training pipeline teaches the LRM to follow shallow reasoning format, neglecting the inner connection between the thinking process and the final answer. On the contrary, LRMs developed from a complete cold start then RL post-training pipeline show less tendency to produce non-factual answer with the above two cognitive behaviors, leading to a more fact-faithful model output.

Finally, we aim to explore the mechanism for the factuality in LRMs from the perspective of miscalibrated internal *uncertainty* (Zhu et al., 2023; Geng et al., 2024). We find that both RL-only and SFT-only LRMs suffer from corrupted calibration by showing that the probability of model outputs and the probability that the answer is correct are not well aligned. We further conduct a probing

experiment to extract the uncertainty from the hidden states of the LRM and find that RL-only and SFT-only LRMs even partially lose the uncertainty information in their hidden states, while the complete cold start then RL pipeline LRMs demonstrate increased accuracy in uncertainty probing. Our results suggest that calibration error could serve as a potential monitoring signal for hallucination in developing LRMs.

Our investigation establishes the relationship between the hallucination pattern of an LRM and its post-training strategy. We also show that hallucination cannot be simply explained by catastrophic forgetting by examining the volume of parameter updating from different training methods. Our initial understanding of hallucinations in LRMs potentially helps to develop more trustworthy models.

2 Related Works

We first introduce the large reasoning models and how they are developed. We then introduce the side effect of LRMs producing hallucination in the scope of factuality.

2.1 Large Reasoning Models

LRMs are motivated by the chain-of-thought intuition—by decomposing complex tasks into multiple sub-goals and introducing extra tokens to "think" before giving the final answer. This has shown better reasoning performance (Wei et al., 2022). Early attempts consider CoT as a native capability of LLMs and try to encourage them to be more engaged in CoT reasoning without an extra training stage. These strategies incorporate CoT instructions in the prompt (Kojima et al., 2022; Zhang et al., 2023), provide LLMs with few-shot CoT reasoning examples (Gandhi et al., 2023; Wei et al., 2022), and design specific CoT reasoning framework with expert heuristics (Gao et al., 2023; Chen et al., 2023; Fu et al., 2023; Cao et al., 2023). There are also efforts devoted to constructing fine-tuning corpora for CoT reasoning (Zelikman et al., 2022; Magister et al., 2023; Qu et al., 2024). However, the scale of these fine-tuning works do not constitute an extra training stage.

Starting from OpenAI-o1 (Jaech et al., 2024), an extra post-training stage for developing long CoT reasoning capability is introduced. The post-training stage involves a significantly large amount of computations to incentivize long CoT reasoning from LLMs, producing a brand new type of models, referred to as LRMs (Xu et al., 2025). These LRMs usually adopt a similar response format, engage in a thinking phase before generating the final token during inference time. There are three typical post-training pipelines that are widely adopted by the main-stream LRMs.

Cold Start SFT with Reasoning RL. Starting from a non-reasoning model, *e.g.,* base model and instruct model, cold start SFT fine-tunes the non-reasoning model on long CoT reasoning data, which produces an intermediate model with primed long-thought reasoning capability. The intermediate model is subsequently engaged in RL training, allowing the model to search for optimal reasoning path on formal reasoning tasks with verifiable rewards (Lambert et al., 2024; Trung et al., 2024). It has become the *de facto* pipeline for developing flagship LRMs of commercial companies, including the DeepSeek-R1 (Guo et al., 2025), Qwen3 (Qwen, 2025), and GLM-Z1 (Hou et al., 2025).

Zero-Style RL. Motivated by AlphaGo-Zero (Silver et al., 2017), which trains an agent to play Go with reinforcement learning from scratch, zero-style RL expects to incentivize long CoT reasoning from non-reasoning model without cold start SFT. This branch of works originates from RL from human feedback (RLHF, Ouyang et al., 2022; Lee et al., 2024). It motivates many subsequent works that aim to stablize the RL training process of LLMs (Yu et al., 2025; He et al., 2025) and provides more accurate reward in the environment. It is worth noting that, at the current stage, zero-style RL has not been found to produce more competent LRMs as compared to full cold start then RL pipeline. It is usually used to produce good intermediate LRMs, which help to gather and synthesize long CoT reasoning data for cold start SFT.

Reasoning Distillation SFT. Most recently, there is general understanding that the RL phase is not necessary to develop LRMs. Instead, performing SFT on non-reasoning model using high-quality long CoT data is sufficient to produce LRMs, such as LIMO (Ye et al., 2025). As manually constructing long CoT data is extremely challenging for human annotators, there are efforts to distill from stronger models to produce LRMs with fewer parameters. The distilled small reasoning models are usually released under the same model family along with many flagship LRMs, such as Qwen3 series LRMs with fewer than 32 billion parame-

ters (Qwen, 2025) and DeepSeek series distilled LRMs (Guo et al., 2025).

2.2 Hallucination

Hallucination (Sahoo et al., 2024) is a notorious phenomenon of LLM, wherein the model fabricates information that is not verified, often leading to incorrect responses. While some hallucinations may reflect creativity (Jiang et al., 2024), our focus is on minimizing outputs that lack trustworthiness or factual grounding. Following OpenAI (2025), we primarily examine hallucination with fact-seeking benchmarks (Wei et al., 2024; Joshi et al., 2017).

3 Hallucination Evaluation

We begin our investigation by addressing the following research question: Will post-training for LRMs increase or reduce hallucination? In particular, we explore what are the factors and techniques in the post-training stage that affect hallucination.

3.1 Experiment Setup

We evaluate the hallucination of LRMs based on fact-seeking benchmarks. The results are evaluated via LLM-as-a-Judge (Zheng et al., 2023; Bai et al., 2023; Fan et al., 2024). We introduce how to type an LRM based on its post-training pipeline and briefly describe the implementation details.

Benchmarks. Fact-seeking benchmarks are usually presented in the form of question answering (QA) task. Specifically, the input is a compositional question which is composed of multiple atomic fact knowledge related sub-questions. The expected answer is usually a relatively short form text. We use SimpleQA (SQA, Wei et al., 2024) and TriviaQA (TQA, Joshi et al., 2017), two of the widely used benchmarks to evaluate hallucination. In particular, TriviaQA is originally proposed for distant supervision training in the context of machine reading comprehension. It provides an excerpt from Wikipedia along with QA pairs. To evaluate hallucination, we omit the provided excepts and only input the question to the model and to see whether the model is able to extract the relevant knowledge from its parametric knowledge. We use the validation set in TriviaQA for evaluation, which consists of $17,944$ QA pairs. SimpleQA is another fact-seeking QA benchmark. The question format is similar to that of TriviaQA, while it is significantly more challenging as its questions are adversarially collected against GPT-4 responses. It consists of $4,326$ different questions.

Model	Ver.	SQA	TQA
DeepSeek-V3	Inst.	23.8	76.8
DeepSeek-R1	LRM	$28.5_{\uparrow 4.7}$	$79.0_{\uparrow 2.2}$
Qwen3-32B	Inst.	5.8	61.1
Qwen3-32B	LRM	$6.7_{\uparrow 0.9}$	$65.1_{\uparrow 4.0}$
GLM-4-9B	Inst.	3.5	48.2
GLM-4-Z1-9B	LRM	$4.5_{\uparrow 1.0}$	$50.6_{\uparrow 2.4}$
GLM-4-32B-Base	Base	8.4	68.1
GLM-4-Z1-32B	LRM	$9.0_{\uparrow 0.6}$	$70.8_{\uparrow 2.7}$

Table 1: Accuracy (%) of LRMs post-trained with *SFT+RL* pipeline on SimpleQA (SQA) and TriviaQA (TQA). We use $\uparrow$ to denote the improvement of the LRM over the non-reasoning backbone model. Ver. and Inst. stands for version and Instruct, respectively.

Evaluation. While the original evaluation protocol for TriviaQA is exact match score, we find it infeasible for generative language models, as the same answer may be represented with diversified texts. For example, the answer to question *"which country is separated into two parts by the town of Limbang of Malaysia?"* is *"Abode of Peace"*. However, *Abode of Peace* is an alias of *Brunei*, while exact match fails to determine *Brunei* as a correct answer. In view of this, we use LLM-as-a-Judge for both TriviaQA and SimpleQA. We deploy the most recently released LLM, Qwen3-32B with reasoning enabled, as our judging model and report the accuracy in percentile.

Model Collection. We classify LRMs into three categories based on their post-training pipelines. (1) **SFT+RL** LRMs are developed with both cold start SFT and verifiable reward RL. (2) **RL-only LRMs** incentivize the long CoT reasoning capability with zero-style RL, omitting the cold start SFT stage. (3) **SFT-only LRMs** only undergo a SFT stage, usually distilled from larger LRMs. In our experiments, we compare these LRMs with their accessible non-reasoning models, including base model or instruct model (*accessible model* means either model API or model checkpoint is released). We prioritize comparing against the model that serves as the post-training initialization to minimize confounding factors.

Implementation Details. For all the LRMs, we prioritize using their officially application programming interface (API). If their API is not provided, we deploy the model with vLLM on $8\times$A100 GPUs. To enable formatted output of base version language models, we provide three demonstration

Model	Ver.	SQA	TQA
MiMo-7B-Base	Base	4.9	45.4
MiMo-7B-RL-Zero	LRM	$1.8_{\downarrow 3.1}$	$15.5_{\downarrow 29.9}$
Qwen2.5-7B	Base	5.3	54.5
DeepMath-Zero	LRM	$5.9_{\uparrow 0.6}$	$34.4_{\downarrow 20.1}$
Qwen2.5-32B	Base	9.7	64.1
DAPO-Qwen-32B	LRM	$5.9_{\downarrow 3.8}$	$58.5_{\downarrow 5.6}$

Table 2: Accuracy (%) of LRMs post-trained with *RL-only* pipeline on SimpleQA (SQA) and TriviaQA (TQA). We use ↓ (↑) to denote the drop (increase) of the LRM over the non-reasoning backbone model.

Model	Ver.	SQA	TQA	Avg
Qwen2.5-14B	Base	5.1	64.3	34.7
DPSK-Qwen-14B	LRM	5.2	58.6	$31.9_{\downarrow 2.8}$
Qwen2.5-32B	Base	9.7	64.1	36.9
DPSK-Qwen-32B	LRM	7.4	63.0	$35.2_{\downarrow 1.7}$
LLaMA-3.1-8B	Base	7.2	63.2	35.2
DPSK-LLaMA-8B	LRM	3.9	47.2	$25.6_{\downarrow 9.6}$
LLaMA-3.3-70B	Base	21.9	74.1	48.0
DPSK-LLaMA-70B	LRM	16.9	75.2	$46.0_{\downarrow 2.0}$
Qwen3-14B-Base	Base	7.8	60.6	34.2
Qwen3-14B	LRM	5.2	62.3	$33.8_{\downarrow 0.4}$

Table 3: Accuracy (%) of LRMs post-trained with *SFT-only* pipeline on SimpleQA (SQA) and TriviaQA (TQA). We also report the average accuracy (Avg) across the two datasets. We use ↓ to denote the drop of the LRM over the non-reasoning backbone model.

examples. For fair comparison, we provide the same demonstrations for models requiring a chat template and enclose those examples in the user query. We show more details in Appendix D.

3.2 Evaluation Results

Takeaway #1: SFT+RL LRMs tend to be less prone to hallucination. We conduct experiments on LRMs spanning DeepSeek-R1, Qwen3, and GLM-4 families. Particularly, we evaluate DeepSeek-R1 (Guo et al., 2025), Qwen3-32B (Qwen, 2025; Yang et al., 2025), GLM-4-Z1-9B-0414 (GLM-4-Z1-9B), and GLM-4-Z1-32B-0414 (GLM-4-Z1-32B) (Hou et al., 2025; GLM, 2025). As deploying DeepSeek-V3-Base is extremely resource consuming and there is no API available, we compare DeepSeek-R1 with DeepSeek-V3 (Liu et al., 2024). For Qwen3-32B, as it fuses reasoning mode with non-reasoning mode, we compare Qwen3-32B LRM version against itself with no-reasoning template. For GLM-4-Z1-9B, we compare with GLM-4-9B, an instruct model, as the base model used to develop the reasoning model is not released. We compare GLM-4-Z1-32B with GLM-4-32B-Base-0414 (GLM-4-32B-Base).

We show their experiment results in Table 1, and mainly focus on the accuracy change between the LRMs and their non-reasoning counterpart (marked with ↑). We find that all four LRMs post-trained with full SFT+RL pipeline consistently obtain enhanced accuracy on both SimpleQA and TriviaQA. This is because long CoT reasoning allows the LRM to explicitly verbalize their induced knowledge, which helps to ease their knowledge reasoning process. We thus conclude that the SFT+RL developed LRMs benefit from increased factuality.

Takeaway #2: RL-only LRMs are more prone to hallucination. We next evaluate LRMs post-trained without the cold start SFT phase, including MiMO-7B-RL-Zero (Team, 2025), DeepMath-Zero (He et al., 2025), and DAPO-Qwen-32B (Yu et al., 2025), whose RL start from MiMo-7B-Base, Qwen2.5-7B, and Qwen2.5-32B, respectively.

The comparison between RL-only LRMs and their non-reasoning version models are shown in Table 2. We can find that most of the RL-only LRMs suffer from a performance drop on SimpleQA and TriviaQA, as compared to their respective base models. The only exception is DeepMath-Zero, which slightly outperforms Qwen2.5-7B on SimpleQA by merely 0.6%, while significantly degrades its performance on TriviaQA by 20.1%. Our experiments show that, post-training with RL only increases hallucination, We hypothesize that the degradation in factuality stems mainly from the natural instability of RL training. This is partially supported by our observations that RL-only LRMs have a higher tendency to be trapped in repetition, or cannot correctly produce an answer faithful to the CoT, which will be explored in Section 4.

Takeaway #3: SFT-only LRMs are also more prone to hallucination. As RL is much more computationally expensive than SFT, there are several attempts to develop lightweight LRMs through SFT with high quality data. These data are usually harvested from a stronger LRM, and the SFT-only post-training pipeline is also referred to as distillation. We evaluate several officially distilled LRMs that are developed by distillation SFT only, including DeepSeek series distilled

LRMs (Guo et al., 2025) and Qwen3 (Yang et al., 2025) series lightweight models. In particular, DeepSeek series distilled LRMs include DeepSeek-R1-Distill-Qwen-14B (DPSK-Qwen-14B), DeepSeek-R1-Distill-Qwen-32B (DPSK-Qwen-32B), DeepSeek-R1-Distill-LLaMA-8B (DPSK-LLaMA-8B), and DeepSeek-R1-Distill-LLaMA-70B (DPSK-LLaMA-70B), which are post-trained on top of Qwen2.5-14B, Qwen2.5-32B, LLaMA-3.1-8B, and LLaMA-3.3-70B, respectively. Our evaluated Qwen series distilled LRM is Qwen3-14B, which is based on Qwen3-14B-Base. We do not include other distilled Qwen3 series models, since only the 14B version released the corresponding non-reasoning model.

We report the experiment results in Table 3. The performance of the evaluated LRMs drops by a large margin on at least one fact-seeking benchmark. Even though we observe minor accuracy increases, such as DPSK-LLaMA-70B on TriviaQA, from 74.1% to 75.2%, we find that the average accuracy of the evaluated LRMs on SimpleQA and TriviaQA drops consistently. This indicates that merely post-training the LRMs with SFT-only undermines their factuality. This could be caused by the fact that SFT only teaches LRMs shallow reasoning format without utilizing long CoT to search for the factual knowledge in their parameters with proper retry and reflection, as explored in Section 4.

4 Behavior Analysis

Based on the observation that both RL-only and SFT-only LRMs are more prone to hallucination, we next investigate which kind of cognitive behaviors encourage these models to make mistakes in fact-seeking tasks. We characterize two primary cognitive behaviors in the long CoT reasoning that affects hallucination: (1) Flaw repetition, where the LRM is trapped in a loop of repeated thoughts. Those reasoning thoughts could be different in language surface but are semantically similar. Even though the required knowledge is encoded in the model parameters, the LRM fails to retrieve it because it cannot end thinking before running out of the context length limit. (2) Think-Answer mismatch, where the answer given by the LRM does not semantically align with its CoT. These are demonstrated by cases in Appendix A.

To explore to what extent the two cognitive behaviors are present in the hallucinated outputs of LRMs, we conduct a statistical analysis on the generated outputs of LRMs. Specifically, we gather wrong answers from GLM-4-Z1-32B, DeepMath-Zero, and DPSK-Qwen-32B, which are SFT+RL, RL-only, and SFT-only LRMs, respectively. To check whether the generated outputs exhibit the two cognitive behaviors, we use Qwen3-32B, a reasoning LLM, to judge whether the generated outputs, especially the CoT part, satisfy flaw repetition or think-answer mismatch.

Model (Type)	SQA		TQA	
	FR	TAM	FR	TAM
GLM-4-Z1-32B (SFT+RL)	5.6	1.0	5.8	1.0
DeepMath-Zero (RL-only)	17.8	4.6	14.7	4.3
DPSK-Qwen-7B (SFT-only)	9.7	1.0	17.9	1.7

Table 4: Frequency (%) of hallucinated outputs that are identified to contain fraw repetition (FR) and think-answer mismatch (TAM) on SimpleQA and TriviaQA.

Takeaway #4: Flaw repetition and think-answer mismatch are two important causes of hallucination for RL-only and SFT-only LRMs. We first verify that both flaw repetition and think-answer mismatch only exist in the output of LRMs. This is testified by our observation that the non-reasoning counterparts, including Qwen2.5-7B and Qwen2.5-32B, do not produce any outputs with the two cognitive behaviors. Then, for LRMs, we show the statistics of the two cognitive behaviors in the hallucinated outputs in Table 4. We find that the SFT-only LRM, DPSK-Qwen-7B, has a clear tendency to produce outputs with flaw repetition, compared to the SFT+RL LRM, GLM-4-Z1-32B. In the meanwhile, the RL-only LRM, DeepMath-Zero, only produces outputs with flaw repetition, but also suffers from a higher rate of think-answer mismatch than the SFT+RL LRM.

5 Hallucination Mechanism Analysis

The last part of our investigation aims to answer the following research question: Is the hallucination of LRMs rooted in the inner mechanism of the model and how can we interpret it? Following similar works in analyzing the mechanism of non-reasoning LLMs, we hypothesize that the hallucination of LRMs is caused by the corrupted calibration during the RL-only and SFT-only post-training.

5.1 Model Calibration

We first analyze the model calibration (Zhu et al., 2023; Geng et al., 2024) of LRMs and their non-

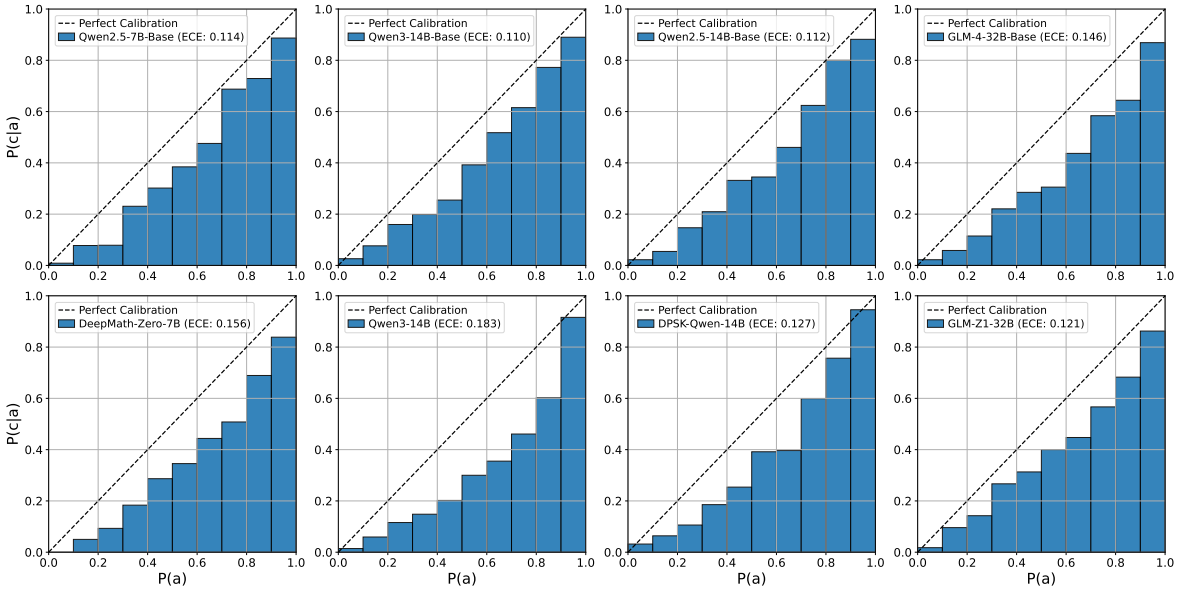


Figure 2: Calibration plot comparing LRMs with their non-reasoning counterparts on TriviaQA. Each plot visualizes the relationship between model confidence $P(a)$—estimated via sampling and majority voting—and the actual correctness probability $P(c|a)$ judged by an external LLM. Models closer to the diagonal with lower Expected Calibration Error (ECE) are better calibrated.

reasoning counterparts. To analyze the calibration of a given LRM, we calculate the uncertainty of the LRM's generated answer a, denoted as $P(a)$, and the probability that the generated answers are correct, denoted as $P(c|a)$. In particular, to calculate $P(a)$, we ask the LRM to answer the question N times. The answer of each response is then extracted with another LLM, where we use Qwen3-32B. We determine the answer of the LRM to the provided question by majority voting among the N extracted answer instances. We then calculate the uncertainty of LRM to its generated answer a as: $P(a) = \sum_{i=1}^{N} \frac{\mathbb{1}(a_i=a)}{N}$, where $\mathbb{1}(a_i = a)$ is an indicator function that returns 1 if a_i is the same as a and 0 otherwise. $P(c|a)$ is calculated as the frequency that a is semantically consistent with the golden answer among the whole benchmark with LLM-as-a-Judge. We implement with $N = 10$ and show the calibration plot on TriviaQA in Figure 2, and calculate their expected calibration error (ECE) to quantify the extent to which the uncertainty of LRMs is mis-calibrated (Naeini et al., 2015).

Takeaway #5: Although SFT+RL pipeline help to improve the calibration, the calibration is corrupted in the SFT-only and RL-only post-training pipelines. We find that the ECE of GLM-4-Z1-32B is 0.12, which is lower than the ECE of its non-reasoning version model, GLM-4-32B-Base, which is 0.146, by 0.026. This indicates that the SFT+RL pipeline helps to improve the calibration of LRMs. However, the ECE of RL-only LRMs, DeepMath-Zero, is 0.156, which are higher than their non-reasoning version models by 0.042. Meanwhile, the ECE of SFT-only LRMs, Qwen3-14B and DPSK-Qwen-14B, is 0.183 and 0.127, which are higher than their non-reasoning version models by 0.073 and 0.015 respectively. These results testify that the calibration of LRMs is corrupted by the SFT-only and RL-only pipelines.

5.2 Uncertainty Probing

We now investigate two different scenarios that the LRM is mis-calibrated. (1) The calibration is corrupted by the SFT-only and RL-only post-training pipelines without any measure to reconstruct from their inner structure. (2) The LRMs fail to verbalize their uncertainty correctly, but the correct uncertainty is still encoded in their inner structure and can be reconstructed with a probing model. To verify which scenario is the case, we conduct a probing experiment to extract the uncertainty from the hidden states of the LRM. If the probe is able to determine whether the model could produce a correct answer, it indicates that the model is calibrated in the hidden space.

For implementation, to train the probe, we use the hidden states of the last token in the question from the LRM as the input. The training objective

Type	Model	Probing Acc.		Diff.
		Backbone	LRM	
SFT+RL	GLM-Z1-32B	68.1	70.8	+2.7
	Qwen3-32B	61.1	65.1	+4.0
RL-only	DeepMath-Zero	54.5	34.4	−20.1
	MiMo-7B-RL-Zero	53.9	15.5	−38.4
SFT-only	DPSK-LLaMA-8B	63.2	47.2	−16.0

Table 5: Uncertainty probing results in accuracy (%).

Model	MLP	Attn.	Emb	Overall
DeepMath-Zero-7B	0.0060	0.0038	0.0027	0.0040
Qwen3-14B	0.1575	0.1670	0.0942	0.1590
DPSK-Qwen-14B	0.1714	0.2793	0.0749	0.3374
GLM-Z1-32B	0.9694	1.0371	1.3266	1.1529

Table 6: Scaled Mean Absolute Error (MAE) between base models and LRMs. All values are scaled by 10^{-2}. Overall is the average MAE of all parameters. MLP, Attention, and Emb represent the MAE for parameters in the multi-layer perception layer, attention layer, and embedding layer, respectively.

is a binary classification task, where the positive samples are the questions that produce correct answers and the negative samples are the questions that produce hallucinated answers. If the trained probe can successfully predict the correctness of the generated answer, it indicates that the uncertainty of LRM is still encoded in the hidden states and can be reconstructed. We train the probe on TriviaQA and randomly held out 20% samples for testing and show the probing accuracy in Table 5.

Takeaway #6: The mis-calibration is rooted in the hidden states of LRMs. The experiment results show that probes for both RL-only and SFT-only LRMs suffer a performance drop compared to their non-reasoning version models. This indicates that the uncertainty with regard to their generated answers are lost during their post-training stage. In contrast, the probes for SFT+RL LRMs show a significant performance gain compared to their non-reasoning version models, which is in line with our observation that SFT+RL help the LRMs to improve their reasoning ability in fact-seeking tasks.

6 Discussion

Although we provide an initial investigation on the hallucination of LRMs, we mainly rely on static analysis based on released LRMs without conducting the post-training pipeline by ourselves. A more comprehensive investigation should be conducted with a more strict variable control, including the training data, updating steps, and hyper-parameters. To do so requires an extremely large amount of computational resources. However, we believe that our work provides a good starting point for future research on hallucination of LRMs and we call for more comprehensive studies in this direction.

Parameter Analysis. A potential alternative explanation for hallucination in LRMs is that it arises from extensive parameter changes during post-training—especially in pipelines that rely heavily on SFT-only or RL-only stages, which may induce more aggressive parameter updates than mixed SFT+RL approaches, and may suffer from catastrophic forgetting (Luo et al., 2023).

To investigate this hypothesis, we conduct a parameter analysis by calculating the mean absolute error (MAE) between each reasoning model and its corresponding base model. The MAE is defined as $\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\theta_i - \theta_{i,b}|$, where θ_i and $\theta_{i,b}$ are the i-th parameters of the LRM and the base model, respectively.

However, the results in Table 6 challenge this explanation. For example, GLM-Z1-32B shows the largest parameter shift (MAE = 1.1529), yet it exhibits relatively mild hallucination. In contrast, models like DPSK-Qwen-14B and Qwen3-14B undergo smaller parameter shifts (MAE = 0.3374 and 0.1590, respectively), yet suffer more significant hallucination behaviors. These findings suggest that hallucination cannot be fully explained by the magnitude of parameter changes alone. Instead, they point to the presence of other contributing factors—potentially including differences in training data, objectives, or alignment methods—that govern hallucination in LRMs. Thus, the intuition that "more parameter change leads to more hallucination" does not hold universally in practice.

7 Conclusion.

This paper investigates the hallucination of LRMs. Specifically, we identify that the SFT-only and RL-only post-training pipelines are the main causes of hallucination in LRMs, while the SFT+RL pipeline can effectively alleviate hallucination. The hallucination of LRMs is mainly reflected in two cognitive behaviors in their thinking process: flaw repetition and think-answer mismatch. We also find that the hallucinating LRMs show corrupted calibration even by probing their inner representations.

Limitations

While our study provides an in-depth empirical analysis of hallucination in large reasoning models , it also comes with several limitations. First, as an academic research lab, we are unable to afford the computational cost of reproducing full-scale, full-parameter reinforcement learning post-training. As a result, our analysis relies entirely on publicly released models. By *publicly released models*, we mean the model is either accessible via API service or can be deployed on a single DGX A100 node with vLLM/sglang This limits our ability to control for confounding factors such as the pretraining corpus, instruction tuning data, or post-training schedules, which may affect the degree of hallucination observed.

Second, our behavioral and mechanistic evaluations use LLM-as-a-Judge to assess factual correctness, which—despite being widely adopted—may introduce some subjectivity or judge-specific bias. We mitigate this by using a state-of-the-art reasoning LLM (Qwen3-32B) for judging, but acknowledge that further work is needed to assess the robustness of this evaluation across different judges or domains.

Third, our study focuses primarily on hallucination in fact-seeking tasks, such as short-form QA from SimpleQA and TriviaQA. These tasks offer well-defined correctness criteria and facilitate controlled comparisons, but may not generalize to more complex forms of generation like long-form synthesis, retrieval-augmented generation, or multi-turn dialogue.

Finally, while we examine factors such as post-training pipeline, cognitive behaviors, calibration, and parameter shift, we do not claim to exhaustively cover all causes of hallucination. Other contributing elements—such as alignment methods, dataset quality, or prompt structure—may also play a significant role. We leave these aspects to future work under more controlled experimental settings.

Ethical Considerations

Our study investigates the hallucination behavior of large reasoning models , which are increasingly used in high-stakes applications such as education, legal assistance, and medical decision support. A key finding of our work is that reasoning-enhanced models, particularly those trained with incomplete post-training pipelines (*e.g.,* RL-only or SFT-only), can exhibit more factual errors than their non-reasoning counterparts. This raises an important societal concern: models that appear more thoughtful or "intelligent" due to their chain-of-thought reasoning may inadvertently gain user trust while delivering inaccurate outputs. Such hallucinated outputs, if unrecognized, may lead to harmful consequences when users rely on them for factual decisions.

At the same time, our analysis contributes positively to the responsible development of language models. By identifying the specific training pipelines and behavioral patterns associated with hallucination, our work provides actionable insights to reduce factual inconsistencies in future LRM development. In particular, we highlight the effectiveness of combining supervised fine-tuning with verifiable reward reinforcement learning (SFT+RL), and we propose uncertainty-based metrics as additional signals to monitor hallucination risk. We hope that our findings will motivate the community to include factuality assessment as a standard component in the evaluation of reasoning-capable LLMs. Moreover, we hope this paper contributes to the construction of trustworthy AI system.

References

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. 2023. Benchmarking foundation models with language-model-as-an-examiner. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, pages 78142–78167.

Shulin Cao, Jiajie Zhang, Jiaxin Shi, Xin Lv, Zijun Yao, Qi Tian, Lei Hou, and Juanzi Li. 2023. Probabilistic tree-of-thought reasoning for answering knowledge-intensive complex questions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12541–12560.

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

Yuchen Fan, Yantao Liu, Zijun Yao, Jifan Yu, Lei Hou, and Juanzi Li. 2024. Evaluating generative language models in information extraction as subjective question correction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/-*

COLING 2024, 20-25 May, 2024, Torino, Italy, pages 6409–6417.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, pages 10421–10430.

Kanishk Gandhi, Dorsa Sadigh, and Noah Goodman. 2023. Strategic reasoning with language models. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: program-aided language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, pages 10764–10799.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595.

Team GLM. 2025. Glm-4-0414 model series. https://github.com/THUDM/GLM-4.

Melody Y Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Helyar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, and 1 others. 2024. Deliberative alignment: Reasoning enables safer language models. *arXiv preprint CoRR*, abs/2412.16339.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, and 1 others. 2025. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint CoRR*, abs/2504.11456.

Zhenyu Hou, Xin Lv, Rui Lu, Jiajie Zhang, Yujiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. 2025. Advancing language model reasoning through reinforcement learning and inference scaling. *arXiv preprint CoRR*, abs/2501.11651.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint CoRR*, abs/2412.16720.

Xuhui Jiang, Yuxing Tian, Fengrui Hua, Chengjin Xu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on large language model hallucination via a creativity perspective. *arXiv preprint CoRR*, abs/2402.06647.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, pages 22199–22213.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint CoRR*, abs/2411.15124.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and 1 others. 2024. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint CoRR*, abs/2412.19437.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, and 1 others. 2025. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint CoRR*, abs/2308.08747.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of*

the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA, pages 2901–2907.

OpenAI. 2025. Openai o3 and o4-mini system card. Published April 16, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, pages 27730–27744.

Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching language model agents how to self-improve. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, pages 55249–55285.

Team Qwen. 2025. Qwen3: Think deeper, act faster. https://qwenlm.github.io/blog/qwen3. Accessed: 2025-04-29.

Pranab Sahoo, Prabhash Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. 2024. A comprehensive survey of hallucination in large language, image, video and audio foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11709–11724.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, and 1 others. 2017. Mastering the game of go without human knowledge. *Nat.*, 550:354–359.

Xiaomi LLM-Core Team. 2025. MiMo: Unlocking the reasoning potential of language model – from pretraining to posttraining. *arXiv preprint CoRR*, abs/2505.07608.

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. ReFT: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7601–7614.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint CoRR*, abs/2411.04368.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, pages 24824–24837.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, and 1 others. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint CoRR*, abs/2501.09686.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, and 1 others. 2025. Qwen3 technical report. *arXiv preprint CoRR*, abs/2505.09388.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. LIMO: Less is more for reasoning. *arXiv preprint CoRR*, abs/2502.03387.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, and 1 others. 2025. DAPO:an open-source llm reinforcement learning system at scale. *arXiv preprint CoRR*, abs/2503.14476.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STaR: Bootstrapping reasoning with reasoning. pages 15476–15488.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, pages 46595–46623.

Chiwei Zhu, Benfeng Xu, Quan Wang, Yongdong Zhang, and Zhendong Mao. 2023. On the calibration of large language models and alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9778–9795.

Appendix

A Case Study

We show two typical cases in Table 7 to illustrate the two cognitive behaviors. The first case is generated by the RL-only LRM, DeepMath-Zero, which

Table 7: Error analysis of LRMs. Both cognitive behaviors that are related to hallucination are observed: (1) **Flaw Repetition**. In the first case, the RL-only LRM DeepMath-Zero repeatedly mentions "Life is a game of fear. We live in fear of the throne, in fear of each other, in fear of the future. " without exploring other possibilities from its parametric knowledge. (2) **Think-Answer Mismatch**. In the second case, the SFT-only LRM DPSK-Qwen-32B first mentions "Freddie Keppard" as the founder of the Red Hot Peppers but then provides a different answer "Fred Hager".

Cognitive Behavior 1: Flaw Repetition	
Question 1: What did Daemon Targaryen say to Rhaenyra about living life in fear in Episode 4, Season 1 of House of the Dragon?	**Ground Truth 1:** You cannot live your life in fear, or you will forsake the best parts of it.
Thinking 1:`<think>` ... Daemon Targaryen says: "Life is a game of fear. We live in fear of the throne, in fear of each other, in fear of the future." Daemon Targaryen says to Rhaenyra, "Life is a game of fear. Fear of the throne, fear of each other, fear of the future." Daemon Targaryen tells Rhaenyra, "Life is a game of fear. Fear of the throne, fear of each other, fear of the future." ...	
Answer 1: `Due to endless repetition, no valid answer provided`	
Cognitive Behavior 2: Think-Answer Mismatch	
Question 2: Which musician founded the Red Hot Peppers?	**Ground Truth 2:** Freddie Keppard
Thinking 2:`<think>` ... I think the answer is that the Red Hot Peppers were founded by Freddie Keppard, but I'm not 100% sure. Alternatively, it might have been someone else. I'm going to go with Freddie Keppard as the founder, but I'm not entirely confident. `</think>`	
Answer 2: The Red Hot Peppers were a studio group associated with OKeh Records in the 1920s, featuring various jazz musicians. It was founded by Fred Hager.	

demonstrates flaw repetition, where the LRM repeatedly mentions the false sentence. The second case is generated by the SFT-only LRM, DPSK-Qwen-32B, which demonstrates think-answer mismatch. In its CoT part, the LRM reaches "Freddie Keppard", while it generates "Fred Hager" as the final answer.

B Implementing LLM-as-a-Judge

In this section, we provide extra details on how we implement LLM-as-a-Judge for SimpleQA and TriviaQA. Specifically, we manually inspect the judging results from the LLM to validate the feasibility of using LLM-as-a-Judge on SimpleQA and TriviaQA. In addition, we provide our prompt template to facilitate reproducibility.

B.1 Inspecting LLM-as-a-Judge

To verify that LLM-as-a-Judge is feasible, we manually inspect 100 judging results given by Qwen3-32B for Qwen2-14B on SimpleQA. We find that all the responses are correctly judged by Qwen3-32B. We also inspect the accuracy of LLM judges on the behavior analysis process and find that 96 out of 100 responses are correctly analyzed by the LLMs. The manual inspection ensures the feasibility of using LLM-as-a-Judge in our research.

B.2 Prompt Template

We use the original grader template from the SimpleQA (Wei et al., 2024) to perform the evaluation of the generated outputs. As the TriviaQA (Joshi et al., 2017), since the answer formulation is similar to the SimpleQA, we also use the same template.

```
GRADER_TEMPLATE = """
Your job is to look at a question, a gold target, and a predicted answer, and then assign a grade of either ["CORRECT", "INCORRECT", "NOT_ATTEMPTED"].
First, I will give examples of each grade, and then you will grade a new example.


The following are examples of CORRECT predicted answers.
```
Question: What are the names of Barack Obama's children?
Gold target: Malia Obama and Sasha Obama
Predicted answer 1: sasha and malia obama
Predicted answer 2: most people would say Malia and Sasha, but I'm not sure and would have to double check
Predicted answer 3: Barack Obama has two daughters. Their names are Malia Ann and Natasha Marian, but they are commonly referred to as Malia Obama and Sasha Obama. Malia was born on July 4, 1998, and Sasha was born on June 10, 2001.
```
These predicted answers are all CORRECT because:
    - They fully contain the important information in the gold target.
    - They do not contain any information that contradicts the gold target.
    - Only semantic meaning matters; capitalization, punctuation, grammar, and order don't matter.
    - Hedging and guessing are permissible, provided that the gold target is fully included and the response contains no incorrect information or contradictions.


The following are examples of INCORRECT predicted answers.
```
Question: What are the names of Barack Obama's children?
Gold target: Malia and Sasha
Predicted answer 1: Malia.
Predicted answer 2: Malia, Sasha, and Susan.
Predicted answer 3: Barack Obama does not have any children.
Predicted answer 4: I think it's either Malia and Sasha. Or it could be Malia and Jackie. Or it could be Joey and Malia.
Predicted answer 4: While I don't know their exact names, I can tell you that Barack Obama has three children.
Predicted answer 5: It's possible you may mean Betsy and Olivia. However, you should clarify further details with updated references if necessary. Is that the correct answer?
Predicted answer 6: It may be the case that Obama's child is named James. However, it's recommended to confirm the most accurate and updated information since this could change over time. This model may not always reflect the most current information.
```
These predicted answers are all INCORRECT because:
    - A factual statement in the answer contradicts the gold target. Incorrect statements that have some hedging (e.g., "it is possible that", "although i'm not sure, i think") are also considered incorrect.
```

```
The following are examples of NOT_ATTEMPTED predicted answers.
```
```
Question: What are the names of Barack Obama's children?
Gold target: Malia and Sasha
Predicted answer 1: I don't know.
Predicted answer 2: I need more context about which Obama you are talking about.
Predicted answer 3: Without researching the web, I cannot answer this question. However, I can tell you that Barack Obama has two children.
Predicted answer 4: Barack Obama has two children. I know that one of them is Malia, but I'm not sure about the other one.
```
```
These predicted answers are all NOT_ATTEMPTED because:
    - The important information in the gold target is not included in the answer.
    - No statements in the answer contradict the gold target.


Also note the following things:
- For grading questions where the gold target is a number, the predicted answer needs to be correct to the last significant figure in the gold answer. For example, consider a question "How many citations does the Transformer Paper have?" with gold target "120k".
    - Predicted answers "120k", "124k", and 115k" are all CORRECT.
    - Predicted answers "100k" and "113k" are INCORRECT.
    - Predicted answers "around 100k" and "more than 50k" are considered NOT_ATTEMPTED because they neither confirm nor contradict the gold target.
- The gold target may contain more information than the question. In such cases, the predicted answer only needs to contain the information that is in the question.
    - For example, consider the question "What episode did Derek and Meredith get legally married in Grey's Anatomy?" with gold target "Season 7, Episode 20: White Wedding". Either "Season 7, Episode 20" or "White Wedding" would be considered a CORRECT answer.
- Do not punish predicted answers if they omit information that would be clearly inferred from the question.
    - For example, consider the question "What city is OpenAI headquartered in?" and the gold target "San Francisco, California". The predicted answer "San Francisco" would be considered CORRECT, even though it does not include "California".
    - Consider the question "What award did A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity win at NAACL '24?", the gold target is "Outstanding Paper Award". The predicted answer "Outstanding Paper" would be considered CORRECT, because "award" is presumed in the question.
    - For the question "What is the height of Jason Wei in meters?", the gold target is "1.73 m". The
```

```
        predicted answer "1.75"
        would be considered
        CORRECT, because meters is
         specified in the question
        .
    - For the question "What is
        the name of Barack Obama's
         wife?", the gold target
        is "Michelle Obama". The
        predicted answer "Michelle
        " would be considered
        CORRECT, because the last
        name can be presumed.
- Do not punish for typos in
    people's name if it's clearly
    the same name.
    - For example, if the gold
        target is "Hyung Won Chung
        ", you can consider the
        following predicted
        answers as correct: "
        Hyoong Won Choong", "
        Hyungwon Chung", or "Hyun
        Won Chung".


Here is a new example. Simply
    reply with either CORRECT,
    INCORRECT, NOT ATTEMPTED. Don'
    t apologize or correct
    yourself if there was a
    mistake; we are just trying to
     grade the answer.
```
```
Question: {question}
Gold target: {target}
Predicted answer: {
    predicted_answer}
```

```
Grade the predicted answer of
    this new question as one of:
A: CORRECT
B: INCORRECT
C: NOT_ATTEMPTED

Just return the letters "A", "B",
     or "C", with no text around
    it.
""".strip()
```

C Licenses for existing assets

The names of the licenses for each asset used in this paper are detailed below.

D Details of Decoding Parameters

We provide the decoding parameters used in our experiments in Table 9. Besides, we use the vLLM framework to deploy the models with verision 0.8.3.

Asset	License
SimpleQA	MIT License
TriviaQA	Apache License Version 2.0
Qwen2.5 Series	Apache License Version 2.0
DeepSeek R1 Series	MIT License
Qwen3 Series	Apache License Version 2.0
DAPO-Qwen	Apache License Version 2.0
Huggingface Transformers	Apache License Version 2.0
vLLM	Apache License Version 2.0
PyTorch	BSD 3-Clause License
Simple-Eval	MIT License

Table 8: Licenses for each asset in the paper.

Model Config	Parameter Settings
MiMo-7B-Series	Temperature = 0.6
DAPO-Qwen-32B	Temperature = 1.0, Top-p = 0.7
Qwen2.5	Temperature = 0.7, Top-p = 0.8
DeepSeek-R1-Distill-Series	Temperature = 0.6, Top-p = 0.95
Deepseek-r1	Temperature = 0.6, Top-p = 0.95
Deepseek-v3	Temperature = 0.6, Top-p = 0.95
GLM-4-Z1	Temperature = 0.6, Top-p = 0.95,Top-k = 40
Qwen3 enable_thinking	Temperature = 0.6, Top-p = 0.95, chat_template_kwargs = {enable_thinking: True}, top_k = 20
Qwen3 close_thinking	Temperature = 0.7, Top-p = 0.8, chat_template_kwargs = {enable_thinking: False}, top_k = 20

Table 9: Sampling parameter settings for different model configurations.