

---

# Reasoning Models Hallucinate More: Factuality-Aware Reinforcement Learning for Large Reasoning Models

---

Junyi Li      Hwee Tou Ng

Department of Computer Science, National University of Singapore  
cheneyjunyi@gmail.com, nght@comp.nus.edu.sg

## Abstract

Large language models (LLMs) have significantly advanced in reasoning tasks through reinforcement learning (RL) optimization, achieving impressive capabilities across various challenging benchmarks. However, our empirical analysis reveals a critical drawback: reasoning-oriented RL fine-tuning significantly increases the prevalence of hallucinations. We theoretically analyze the RL training dynamics, identifying high-variance gradient, entropy-induced randomness, and susceptibility to spurious local optima as key factors leading to hallucinations. To address this drawback, we propose Factuality-aware Step-wise Policy Optimization (**FSPO**), an innovative RL fine-tuning algorithm incorporating explicit factuality verification at each reasoning step. FSPO leverages automated verification against given evidence to dynamically adjust token-level advantage values, incentivizing factual correctness throughout the reasoning process. Experiments across mathematical reasoning and hallucination benchmarks using Qwen2.5 and Llama models demonstrate that FSPO effectively reduces hallucinations while enhancing reasoning accuracy, substantially improving both reliability and performance.

🔗 <https://github.com/nusnlp/FSPO>

## 1 Introduction

Large language models (LLMs) have recently achieved remarkable breakthroughs in natural language generation and complex reasoning tasks [9, 15]. This progress is fueled by new training paradigms that imbue LLMs with reasoning capabilities, including fine-tuning on long chain-of-thought (CoT) solutions [43] and reinforcement learning (RL) based optimization [9]. By leveraging these approaches, state-of-the-art models can decompose complex problems into reasoning steps and iteratively refine their answers, yielding impressive results on challenging benchmarks in mathematics [10], multi-hop question answering [41], coding [2], and decision-making [32].

Along with improved reasoning ability, a critical question has emerged: *To what extent do reasoning models maintain factuality after RL fine-tuning?* In this work, our empirical observations indicate that when LLMs are trained with outcome-driven RL, they tend to produce more factually incorrect or fabricated statements (i.e., hallucinations [13, 46]) during their reasoning process. We find that these hallucinations usually stem from incorrect intermediate reasoning, i.e., the model may generate unsupported or false claims in the chain-of-thought, even if its final answer is occasionally correct. Indeed, RL-aligned models can be highly fluent and confident in their step-by-step explanations while subtly introducing factual errors [40]. This issue undermines the reliability of RL-trained reasoning models, as a flawed or untruthful answer is problematic for trust and interpretability.

To understand why outcome-based RL fine-tuning exacerbates hallucinations, we conduct a theoretical analysis of the RL training dynamics for reasoning tasks. Our analysis identifies three primary factors.

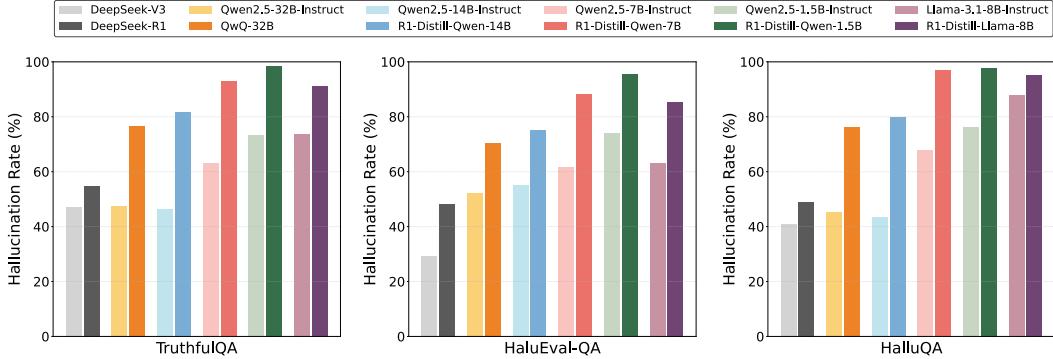


Figure 1: Hallucination rate (%) for six groups of twelve LLMs across three benchmarks.

First, optimizing solely for correct final answers leads to extremely high variance in the policy gradient when such answers are rare, yielding unstable training updates. Second, the need to discover rewarding outputs forces the policy to maintain a high level of entropy in its predictions, heightening the risk of hallucinations. Third, standard RL objectives admit spurious local optima whereby the model can converge to a confidently held wrong answer that yields zero reward. These three factors, i.e., high-variance gradient, entropy-induced randomness, and spurious local optima, together explain why a purely outcome-based RL approach can cause a reasoning model to hallucinate and struggle to learn faithful reasoning patterns.

To mitigate these issues, we propose **Factuality-aware Step-wise Policy Optimization (FSPO)**, a novel RL fine-tuning algorithm that explicitly incorporates factuality feedback at each reasoning step. The key idea is to adjust the advantage values of individual tokens using a step-wise factuality reward signal, encouraging LLMs to generate more factual content. Specially, we employ an automated verifier to check whether each generated reasoning sentence is entailed by the given evidence, yielding step-wise factuality scores. These scores are then integrated into the overall reward signal to adjust the advantages of tokens during optimization, rewarding factually correct tokens while penalizing incorrect tokens. This factuality-aware reward shaping provides denser and more informative feedback to the policy, addressing the sparse-signal problem and reducing training instability. FSPO steers the policy toward solutions that are not only correct but also generated via faithful, verifiable chains of reasoning, thereby directly mitigating hallucinations in the RL training process.

We conduct extensive experiments to evaluate the effectiveness of FSPO on both mathematical reasoning and hallucination benchmarks using Qwen2.5-7B-Base/Instruct and Llama-3.1-8B-Instruct. The results show that our method achieves significantly improved mathematical reasoning performance and reduced hallucinations. Further detailed analyses indicate that our method can enhance the factuality of reasoning steps without compromising the quality of generated content.

## 2 Preliminary Experiments

To explore the effect of RL on hallucinations, we conduct preliminary experiments based on several LLMs and hallucination benchmarks.

**Models and Datasets.** For comparative analysis, we select several groups of LLMs without and with large-scale RL training: (1) *DeepSeek-V3* [21] and *DeepSeek-R1* [9]; (2) *Qwen2.5-32B-Instruct* [39] and *QwQ-32B* [28]; (3) *Qwen2.5-14B/7B/1.5B-Instruct* [39] and *R1-Distill-Owen-14B/7B/1.5B* [9]; (4) *Llama3.1-8B-Instruct* [8] and *R1-Distill-Llama-8B* [9]. These distilled models are fine-tuned on the synthesized long-CoT data from DeepSeek-R1, thus exhibiting similar thinking modes to RL-tuned models. For the hallucination datasets, we select three benchmarks, namely *TruthfulQA* [20], *HaluEval* [19], and *HalluQA* [4]. In *TruthfulQA*, we adopt the generation task, where the model is prompted to generate an answer to a question, and then utilize a fine-tuned GPT-3 to predict the truthfulness of the answer. The hallucination rate is equal to the ratio of untruthful answers. *HalluQA* follows the same method as *TruthfulQA*, except that it employs GPT-4 as the evaluator. For *HaluEval*, we adopt the QA subset, where the LLM is provided with a question and two candidate answers and

it needs to assess which answer is factually correct. The hallucination rate refers to the ratio of false judgments made by the model.

**Results and Analysis.** The results are shown in Figure 1. We can clearly observe that after large-scale training with RL or long CoT data, models demonstrate dramatically higher degree of hallucinations across the three benchmarks. To further investigate the underlying causes of this phenomenon, we randomly select 200 samples from HaluEval-QA that DeepSeek-V3 answers correctly while DeepSeek-R1 does not. We manually examine these cases and classify the source of errors (refer to Appendix E for more details). In Figure 2, we can see that most errors stem from incorrect reasoning steps. We hypothesize that this may be attributed to the outcome reward modeling, where rewards and penalties are assigned solely based on the correctness of the final answer, while ignoring signals from intermediate reasoning steps. In the following sections, we will provide a theoretical analysis of how outcome-based reward modeling in RL could lead to hallucinations, and then propose the factuality-aware step-wise RL algorithm to mitigate this issue.

### 3 Background

**Reinforcement Learning.** In reinforcement learning, language model generation can be formulated as a token-level Markov decision process, denoted by  $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{R}, \pi_\theta\}$ . At the  $t$ -th generation step, the state  $s_t \in \mathcal{S}$  is the concatenation of the input prompt  $x$  and the generated sequence so far  $y_{<t} = [o_1, \dots, o_{t-1}]$ . The policy  $\pi_\theta(\cdot|s_t)$  (parameterized by  $\theta$ ) selects the next action  $o_t \in \mathcal{A}$  by generating a token from the vocabulary  $\mathcal{A}$ , leading to a deterministic transition to the next state  $s_{t+1} = [s_t, o_t]$ . Let  $Y^*$  denote the set of correct answers for a given input prompt  $x$ . Existing research mostly adopts a binary reward  $\mathcal{R}$  for each output  $y$ , which is sparse and discontinuous [32, 18]. For simplicity, we consider a prevalent format in this paper:  $\mathcal{R}(y) = 1$  when  $y \in Y^*$ , or  $\mathcal{R}(y) = 0$  when  $y \notin Y^*$ . The expected return (reward) under the policy  $\pi_\theta$  is:

$$\mathcal{J}(\theta) = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} [\mathcal{R}(y)] = \sum_y \mathcal{R}(y) \pi_\theta(y|x). \quad (1)$$

With the binary reward, the expected return is essentially the probability of producing a correct output under the current policy  $\pi_\theta$ . In the context of RL, we can update the policy using policy gradient algorithms. According to the policy gradient, the gradient of this objective can be written as:

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} [\mathcal{R}(y) \nabla_\theta \log \pi_\theta(y|x)]. \quad (2)$$

The policy gradient methods maximize the expected total reward by repeatedly estimating the gradient. Several policy gradient algorithms have been proposed such as Proximal Policy Optimization [29].

**Group Relative Policy Optimization.** In DeepSeek-Math [30], researchers proposed a new policy gradient method, Group Relative Policy Optimization (GRPO), which improves the effectiveness and efficiency of RL training based on Proximal Policy Optimization (PPO) [29]. The PPO algorithm maximizes the following surrogate objective:

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \sum_{t=1}^{|y|} \left\{ \min \left[ \frac{\pi_\theta(o_t|x, y_{<t})}{\pi_{\theta_{\text{old}}}(o_t|x, y_{<t})} A_t, \text{clip} \left( \frac{\pi_\theta(o_t|x, y_{<t})}{\pi_{\theta_{\text{old}}}(o_t|x, y_{<t})}, 1 - \epsilon, 1 + \epsilon \right) A_t \right] \right\}, \quad (3)$$

where  $\pi_{\theta_{\text{old}}}$  is the policy before the update,  $\epsilon$  is the clip hyperparameter, and  $A_t$  is the advantage of the  $t$ -th token. In PPO, a value model is learned to estimate the advantage based on the reward  $\mathcal{R}$ , which is computationally expensive. Thus, GRPO uses the average reward of multiple sampled outputs as baseline in advantage estimation. Specifically, for each input  $x$ , GRPO samples a group of outputs  $\{y_1, \dots, y_G\}$  and computes their rewards  $\{\mathcal{R}_1, \dots, \mathcal{R}_G\}$ , then sets the advantage of all tokens from  $y_i$  as  $A_{i,t} = (\mathcal{R}_i - \text{mean}(\{\mathcal{R}_1, \dots, \mathcal{R}_G\})) / \text{std}(\{\mathcal{R}_1, \dots, \mathcal{R}_G\})$ . The GRPO objective can be written as:

$$\begin{aligned} \mathcal{J}_{\text{GRPO}}(\theta) &= \mathbb{E}_{y \sim \pi(\cdot|x)} \\ &\frac{1}{G} \sum_{i=1}^G \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \left\{ \min \left[ \frac{\pi_\theta(o_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|x, y_{i,<t})} A_{i,t}, \text{clip} \left( \frac{\pi_\theta(o_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|x, y_{i,<t})}, 1 - \epsilon, 1 + \epsilon \right) A_{i,t} \right] \right\}. \end{aligned} \quad (4)$$

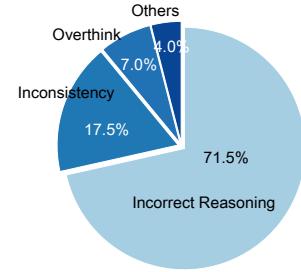


Figure 2: Distribution of hallucination causes over 200 samples from HaluEval-QA.

Compared to PPO, GRPO foregoes the value model to estimate advantages that is typically the same size as the policy model, relieving the memory and computational burden.

## 4 Theoretical Analysis

In this section, we present a theoretical analysis that characterizes how the simplified and prevalent binary reward format (i.e., 1/0) affects the learning dynamics and induces hallucinations. The analysis can extend to any arbitrary bounded real rewards.

**Theorem 4.1** (Variance of Policy Gradient in Binary Reward RL). *Under a binary reward function, the variance of the policy gradient estimator is directly proportional to the probability of generating correct answers. Let  $p$  denote the probability of producing a correct answer, so the variance of the policy gradient estimator is on the order of  $p(1 - p)\|\nabla_\theta \log \pi_\theta(y^*|x)\|^2$  (where  $y^*$  is a correct answer). For small  $p$ , this variance is extremely high, hindering stable learning.*

*Proof.* The REINFORCE gradient for a given input is  $g = R(y) \nabla_\theta \log \pi_\theta(y|x)$ . This is a random vector depending on the sampled output  $y$ . There is a  $p$  chance that the sampled output is correct (yielding  $R = 1$ ) and a  $1 - p$  chance that it is wrong (yielding  $R = 0$ ). Hence, the gradient  $g$  is zero with probability  $1 - p$ , and equals  $\nabla_\theta \log \pi_\theta(y^*|x)$  with probability  $p$ . The expected gradient is  $\mathbb{E}[g] = p \nabla_\theta \log \pi_\theta(y^*|x)$ . The second moment is  $\mathbb{E}[\|g\|^2] = p \|\nabla_\theta \log \pi_\theta(y^*|x)\|^2$ . Therefore, the covariance (and in particular the trace as a measure of variance magnitude) is on the order of

$$p \|\nabla_\theta \log \pi_\theta(y^*|x)\|^2 - \|p \nabla_\theta \log \pi_\theta(y^*|x)\|^2 = p(1 - p) \|\nabla_\theta \log \pi_\theta(y^*|x)\|^2. \quad (5)$$

For very small  $p$ , the factor  $1 - p$  is near 1, so roughly  $\text{Var}(g) \approx p \|\nabla \log \pi_\theta(y^*)\|^2$ . In other words, when the model has low chance to generate correct answers, the gradient estimator is almost always zero but occasionally a large spike, yielding extremely high variance relative to its mean. This high variance makes the training updates very unstable, as gradient estimates will fluctuate widely between zero and large values. We conclude that, with a binary reward, most of the time the model receives no learning signal, and when it does, the signal is an outsized jump.  $\square$

**Theorem 4.2** (Exploration-Exploitation Trade-off and Hallucination Risk). *Given the high-variance gradient under a binary reward, maximizing the expected binary reward objective  $\mathcal{J}(\theta)$  will enforce an implicit constraint that the entropy  $H_\theta(x)$  remains sufficiently large. Let  $\epsilon > 0$  be a small constant reflecting minimum acceptable exploration probability. To avoid stagnation at incorrect local optima without gradient feedback, the optimal policy update rule must satisfy:  $H_\theta(x) \geq H_{\min}(\epsilon) > 0$ .*

*Proof.* The goal of policy model is to increase the probability of producing correct answers. In early stages or under uncertain knowledge (i.e.,  $p$  is small), gradient updates are extremely noisy, often zero or very large (Theorem 4.1). To reliably find rewarding outputs, the agent must ensure it samples broadly from its policy distribution. Given  $\epsilon > 0$  the minimal exploration threshold of ensuring a non-negligible probability of sampling potentially correct outputs, the exploration constraint is:

$$\pi_\theta(y|x) \geq \epsilon, \quad \forall y \in Y_{\text{unexplored}}, \quad (6)$$

where  $Y_{\text{unexplored}}$  is a subset of unexplored outputs that have unknown correctness. Maintaining such a constraint implies a sufficiently high entropy  $H_\theta(x)$ , because low entropy distributions concentrate their mass on few outputs, increasing the risk that these outputs yield zero reward consistently:

$$H_\theta(x) = - \sum \pi(y|x) \log \pi(y|x) \geq H_{\min}(\epsilon) > 0. \quad (7)$$

The necessity for exploration to ensure non-stagnation at zero-reward local minima introduces the exploration-exploitation trade-off. On one hand, the entropy must remain relatively high to explore potential answers, increasing the probability of hallucinated outputs. On the other hand, maximizing immediate reward requires concentrating distribution on already known rewarding answers.  $\square$

**Theorem 4.3** (Local Optima and Incorrect Convergence). *Binary reward modeling admits spurious local optima where the policy produces incorrect answers with high confidence and receives zero reward, yet has no gradient incentive to change. Any policy that deterministically outputs a particular wrong answer is a stationary point of the optimization, i.e., once the model heavily commits to a wrong answer, the policy gradient provides no immediate push away from that policy.*

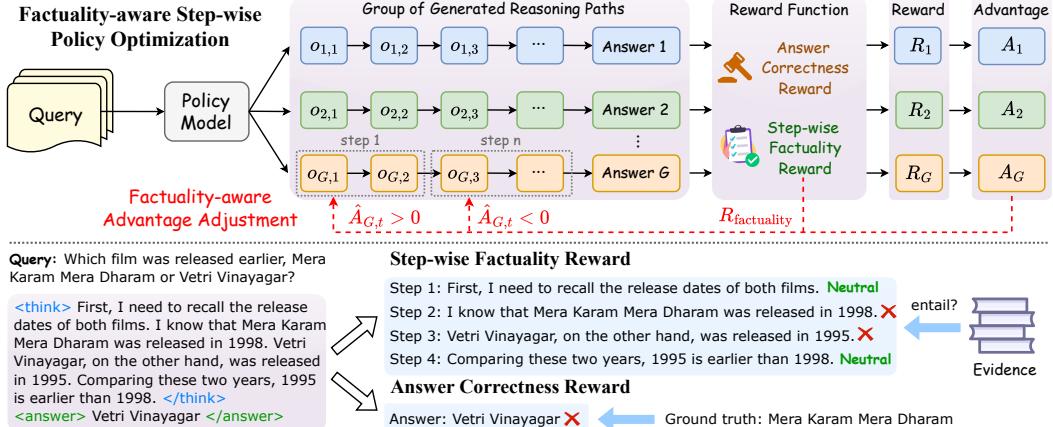


Figure 3: Overall framework of our proposed FSPO approach.

*Proof.* Given the objective  $\mathcal{J}(\theta) = \sum \mathcal{R}(y)\pi_\theta(y|x)$ , it is maximized at  $\mathcal{J} = 1$  when the policy outputs a correct answer with probability 1. However, when a policy is certain but wrong for some incorrect answers  $y_{\text{false}}$ , the probability  $\pi_\theta(y_{\text{false}}|x) = 1$  (so  $\pi_\theta(y^*|x) = 0$  for all correct answers  $Y^*$ ). For this policy, the expected reward  $\mathcal{J}(\theta) = 0$ . Importantly, the gradient  $\nabla_\theta \mathcal{J}(\theta)$  is zero as well, since  $\nabla_\theta \mathcal{J} = \mathbb{E}[\mathcal{R}(y)\nabla \log \pi_\theta(y|x)]$  and  $\mathcal{R}(y)$  is always 0 under this policy (only generating incorrect answers). Thus,  $\theta$  is a stationary point (the gradient is zero and no update is favored).

In a strict mathematical sense, the policy of always outputting a particular wrong answer is actually a local minimum of the objective  $\mathcal{J}(\theta)$ , since any small increase in the probability of a correct answer would increase  $\mathcal{J}(\theta)$  slightly. However, the surface around it is a broad, flat plateau at reward zero for every direction not immediately introducing the correct answer. Since REINFORCE updates are driven solely by sampled positive rewards, the algorithm has no gradient signal to push the parameters off this plateau. In effect, the training has converged to a policy that outputs a wrong answer with full confidence, and because it never explores the correct answer, it gets no feedback. We call such states spurious local optima: the model is confidently wrong and training fails to progress.  $\square$

## 5 Approach

To mitigate the hallucination issue during large-scale RL training, we propose the **Factuality-aware Step-wise Policy Optimization (FSPO)** algorithm. As shown in Figure 3, by integrating step-wise factuality signals into the policy optimization process, our approach encourages the model to generate more factual and faithful reasoning steps.

### 5.1 Reward Function Design

The core of reinforcement learning is to design appropriate reward functions that encourage high-quality reasoning trajectories and penalizes low-quality ones, guiding the policy model towards better responses using the RL algorithm. In our method, we introduce two rewards, i.e., answer correctness reward and step-wise factuality reward.

**Step-wise Factuality Reward.** Typically, factuality is mainly related to world knowledge, so we focus on complex knowledge-intensive tasks in this work, e.g., multi-hop question answering [41]. Specifically, for an input question  $x$ , we assume that it can be associated with knowledge snippets in a source such as Wikipedia, denoted by  $\mathcal{K}$ , which can be called *evidence*. We first split the reasoning text into multiple sentences as steps, denoted by  $\{z_1, \dots, z_N\}$ , and then employ an LLM to determine whether the evidence  $\mathcal{K}$  can entail each sentence  $z_j$  to assign a step-wise factuality reward as:

$$\mathcal{R}_{\text{factuality}}(z_j) = \begin{cases} 1, & \text{if the sentence } z_j \text{ can be entailed from the evidence } \mathcal{K} \\ 0, & \text{if the sentence } z_j \text{ is neutral to the evidence } \mathcal{K} \\ -1, & \text{if the sentence } z_j \text{ contradicts the evidence } \mathcal{K} \end{cases} \quad (8)$$

We observe that the policy model might generate texts unrelated to the question in the reasoning process such as connective phrases or exploratory sentences (e.g., “Aha”, “Wait”), which can be regarded as neutral to the evidence.

**Answer Correctness Reward.** In most existing research [9, 38, 33], answer correctness reward is usually adopted due to its significant scalability and simplicity. Specially, given an input prompt  $x$ , the policy model generates an output  $y$ , which includes intermediate reasoning texts followed by the final answer. We use a rule-based reward function to evaluate an answer’s correctness as follows:

$$\mathcal{R}_{\text{answer}}(y) = \begin{cases} 1, & \text{if the final answer fully matches the ground truth} \\ 0, & \text{if the final answer does not match the ground truth} \end{cases} \quad (9)$$

We simplify the answer reward design aiming to alleviate reward hacking and promote more diverse problem-solving behaviors based on mere outcome feedback.

To reduce the effect of learning from sparse outcome reward, the final reward for an output  $y$  consists of both answer correctness and step-wise factuality rewards, which can provide denser feedback and mitigate the *high-variance gradient* issue (Theorem 4.1) in typical RL binary reward modeling:

$$\mathcal{R}_{\text{final}}(y) = \mathcal{R}_{\text{answer}}(y) + \frac{1}{N} \sum_{j=1}^N \mathcal{R}_{\text{factuality}}(z_j). \quad (10)$$

## 5.2 Factuality-Aware Policy Optimization

For online policy optimization, we adopt Group Relative Policy Optimization (GRPO) [30], which samples a group of outputs  $\{y_1, \dots, y_G\}$  for each input  $x$  and then obtains their rewards  $\{\mathcal{R}_1, \dots, \mathcal{R}_G\}$  to compute the advantage  $A_i$  for  $y_i$  as  $A_i = (\mathcal{R}_i - \text{mean}(\{\mathcal{R}_1, \dots, \mathcal{R}_G\})) / \text{std}(\{\mathcal{R}_1, \dots, \mathcal{R}_G\})$ .

**Factuality-Aware Advantage Adjustment.** Most existing work assigns the same advantage (derived from the overall response) to all tokens [30, 38], which is suboptimal and does not differentiate between them. We argue that, regardless of whether the final answer is correct, factually correct reasoning tokens should be encouraged, while incorrect reasoning tokens should be penalized. Therefore, we adjust the advantage of each token  $o_{i,t} \in z_j$  for the output  $y_i$  as follows:

$$\hat{A}_{i,t} = \begin{cases} A_i, & \text{if } A_i > 0 \wedge \mathcal{R}_{\text{factuality}}(z_j) = 1 \text{ or } A_i < 0 \wedge \mathcal{R}_{\text{factuality}}(z_j) = -1 \\ -A_i, & \text{if } A_i > 0 \wedge \mathcal{R}_{\text{factuality}}(z_j) = -1 \text{ or } A_i < 0 \wedge \mathcal{R}_{\text{factuality}}(z_j) = 1 \\ A_i, & \text{if } A_i = 0 \text{ or } \mathcal{R}_{\text{factuality}}(z_j) = 0 \end{cases} \quad (11)$$

In this manner, each token  $o_{i,t}$  is re-weighted by the factuality-aware advantage  $\hat{A}_{i,t}$ . Whenever the generated reasoning contains at least one sentence that is contradicted or entailed by the evidence ( $\mathcal{R}_{\text{factuality}} = 1, -1$ ), the advantages for tokens inside that sentence are not zero, even if the overall answer is still wrong. Consequently, the gradient is strictly non-zero, supplying a descent (or ascent) direction that pushes probability mass away from factually-incorrect tokens and toward factually-verified ones and enabling gradient-based learning to leave the *spurious local optima* (Theorem 4.3). Based on the adjusted advantage, we adopt the objective in Eq. 4 to optimize the policy model:

$$\mathcal{J}_{\text{FSPO}}(\theta) = \mathbb{E}_{y \sim \pi(\cdot|x)} \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \left\{ \min \left[ \frac{\pi_\theta(o_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|x, y_{i,<t})} \hat{A}_{i,t}, \text{clip} \left( \frac{\pi_\theta(o_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|x, y_{i,<t})}, 1 - \epsilon, 1 + \epsilon \right) \hat{A}_{i,t} \right] \right\} \right]. \quad (12)$$

## 6 Experiments

In this section, we present our experimental setup, the main results, and our analysis of the results.

### 6.1 Experimental Setup

**Datasets and Metrics.** To perform FSPO, we adopt a challenging subset of HotpotQA [41] and 2WikiMultiHopQA [11] from previous work [33] and randomly select 2K examples for training

(including questions, answers, and corresponding Wikipedia segments). Besides, to improve the complex reasoning capabilities of LLMs, we incorporate the SimpleRL dataset [45] with 8K mathematical problems for vanilla RL training. For hallucination evaluation, we adopt three widely-used benchmarks, i.e., TruthfulQA [20], HalluEval [19], and HalluQA [4]. For HalluEval, we adopt the QA subset and report the accuracy with which the model can discriminate correct and hallucinated answers. For TruthfulQA and HalluQA, we report the ratio of outputs that the judge model predicts as truthful. We employ GSM8K [5], MATH-500 [10], AIME 2024 [22], and AIME 2025 [23] to evaluate mathematical reasoning performance and report the Pass@1 accuracy results in these datasets. For more details, refer to Appendix C.

**Baselines.** We employ Qwen2.5-7B-Base/Instruct [39] and Llama3.1-8B-Instruct [8] as backbone models for our approach. We compare our model against three groups of baselines:

- **API-based Models** include DeepSeek-V3 [21], DeepSeek-R1 [9], OpenAI GPT-4o, and GPT-o1.
- **Reasoning Models** consist of QwQ-32B [28], Distill-Qwen-7B/14B/32B, and Distill-Llama-8B [9], which are fine-tuned on the long CoT data (800K samples) synthesized from DeepSeek-R1.
- **Open-source Models** involve Qwen2.5-7B-Base/Instruct and Llama3.1-8B-Instruct, which are direct comparisons and can be easily accessed through their open-source weights.

**Implementation Details.** We train our model using the verl [31] framework. Specifically, during training, we use a batch size of 8, generate 8 rollouts for each prompt, set a maximum prompt and response length of 2,048 tokens, and train using a mini-batch size of 1,024. We train our model on the mixture dataset of HotpotQA subset and SimpleRL for one epoch with a constant learning rate of 4e-7 and temperature parameter of 1.0. The coefficient for KL loss is set to 1e-3 and the clip ratio is 0.2. We adopt the HHEM-2.1 model [14] to verify step-wise factuality. During evaluation, we set the sampling temperature to 1.0. We present prompt templates throughout this work in Appendix F.

## 6.2 Main Results

The experimental results presented in Table 1 demonstrate the performance of our proposed method, FSPO, across various reasoning and hallucination benchmarks compared to multiple competitive models, including API-based, reasoning, and open-source alternatives. On hallucination benchmarks, FSPO clearly outperforms all open-source, reasoning and even some API-based models. Specifically, FSPO based on Qwen2.5-7B-Base achieves the highest scores among all open-source and reasoning models on hallucination datasets, clearly demonstrating the effectiveness of our training approach in substantially reducing hallucinations and enhancing factual correctness.

On reasoning benchmarks, FSPO achieves superior results within the open-source category, notably surpassing other base models like Qwen2.5-7B-Instruct and Llama3.1-8B-Instruct by significant margins (e.g., GSM8K 89.5% vs. 73.2% and 77.5%, respectively). FSPO even surpasses some reasoning models on GSM8K, and maintains competitive performance on MATH-500. This shows that our FSPO method can enhance reasoning capabilities while reducing hallucinations.

In terms of advanced reasoning tasks, FSPO delivers competitive outcomes, especially on AIME 2024 and 2025, with scores of 20.0% and 13.3% respectively. These scores notably surpass smaller base models, although trailing the reasoning and much larger API-based models. Such results underscore FSPO’s robust mathematical reasoning capabilities, showcasing the efficacy of our specialized fine-tuning methods in elevating performance on complex mathematical reasoning tasks despite the base model’s comparatively modest size.

## 6.3 Further Analysis

We further conduct detailed analysis on several key aspects of our approach. Due to similar findings, the following analysis is focused on MATH-500 and HalluEval-QA with Qwen2.5-7B-Base.

**Ablation Analysis.** To validate the effectiveness of our approach, we ablate its key design elements. We compare FSPO to two variants: 1) *GRPO* trains the model on the mixture dataset using typical GRPO with only answer reward in Eq. 9; and 2) *GRPO w/factuality* incorporates both answer and factuality rewards in Eq. 10. The results from Figure 4 further support the effectiveness of FSPO,

Table 1: Evaluation results of closed-source, distilled, and open-source models across four reasoning benchmarks and three hallucination benchmarks. The **bold** and underlined scores denote the best and second best results among open-source models and our trained models.

Model	GSM8K	MATH 500	AIME24 (Pass@1)	AIME25 (Pass@1)	TruthfulQA↑	HaluEval-QA↑	HalluQA↑
<b>API-based Models</b>							
DeepSeek-V3	89.3	90.2	39.2	26.6	53.0	70.6	59.1
DeepSeek-R1	92.0	97.3	79.8	66.7	45.2	51.8	51.1
GPT-4o-0513	90.8	74.6	9.3	13.3	59.0	62.6	53.1
GPT-o1-1217	92.3	96.4	79.2	76.7	62.4	70.2	56.0
<b>Reasoning Models</b>							
QwQ-32B	88.6	89.8	79.5	56.7	23.2	29.6	23.7
R1-Distill-Qwen-32B	87.4	94.3	72.6	53.3	19.7	33.5	26.9
R1-Distill-Qwen-14B	85.1	93.9	69.7	50.0	18.2	24.8	20.2
R1-Distill-Qwen-7B	84.3	92.8	55.5	33.3	6.9	11.6	3.1
R1-Distill-Llama-8B	82.1	89.1	50.4	26.7	8.8	14.6	4.6
<b>Open-source Models</b>							
Qwen2.5-7B-Base	65.2	35.7	3.3	3.3	38.2	48.0	39.5
Qwen2.5-7B-Instruct	73.2	51.6	6.7	3.3	36.7	38.4	32.0
Llama3.1-8B-Instruct	77.5	33.1	6.7	0.0	26.4	36.7	12.2
FSPO (Qwen-Base)	<b>89.5</b>	<b>75.5</b>	<u>16.7</u>	<b>13.3</b>	<b>58.4</b>	<b>83.0</b>	<b>52.0</b>
FSPO (Qwen-Instruct)	<u>89.4</u>	<u>74.7</u>	<b>20.0</b>	<b>13.3</b>	<u>54.0</u>	64.7	<u>50.0</u>
FSPO (Llama-Instruct)	86.2	68.3	13.3	6.7	41.1	<u>67.1</u>	42.0

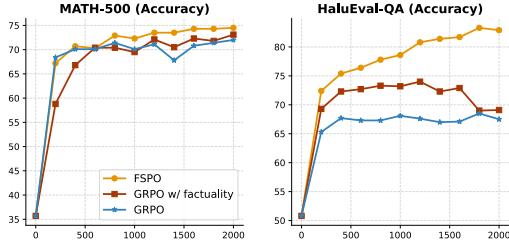


Figure 4: Pass@1 accuracy over training steps for ablation analysis.

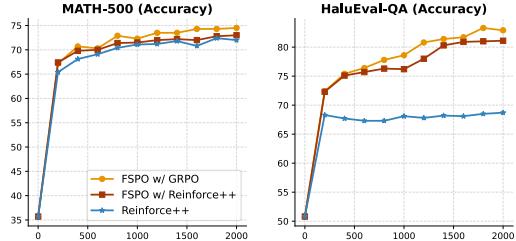


Figure 5: Pass@1 accuracy over training steps with different RL algorithms.

clearly outperforming baseline methods GRPO and GRPO with factuality scores across training steps on both MATH-500 and HaluEval-QA. FSPO maintains consistently superior reasoning performance while achieving continuous improvement in reducing hallucinations, indicating robust generalization capabilities and the practical benefits of our proposed method.

**RL Algorithm Analysis.** We further adapt FSPO to the Reinforce++ algorithm [12] for validating the adaptability and effectiveness of FSPO beyond GRPO. As shown in Figure 5, FSPO with Reinforce++ exhibits significant improvement over Reinforce++ alone, closely following the performance of FSPO with GRPO. Moreover, FSPO with GRPO consistently achieves higher accuracy than the Reinforce++ baselines across training steps on both the MATH-500 and HaluEval-QA benchmarks. These findings indicate that our proposed FSPO approach generalizes well to other reinforcement learning algorithms, enhancing their effectiveness and stability in solving complex reasoning and reducing hallucinations.

**Number of Training Samples.** To investigate the model performance under varying numbers of training samples, we randomly sample 1K, 2K, 4K, and 8K samples from the HotpotQA subset [33], with the results shown in Figure 6. On MATH-500, the accuracies are similar from 1K to 4K samples while diminishing with 8K samples, which might be because more factual QA data lower the math reasoning performance. In contrast, on the HaluEval-QA task, significant accuracy improvement continues up to 2K samples, with 4K and 8K samples achieving mediocre results. This demonstrates that our approach can significantly reduce hallucinations using only a small amount of data, without compromising the model’s capacity for complex reasoning.

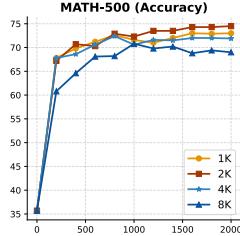


Figure 6: Pass@1 accuracy over training steps with varying number of training samples.

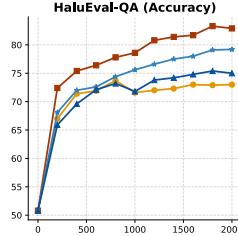


Figure 6: Pass@1 accuracy over training steps with varying number of training samples.

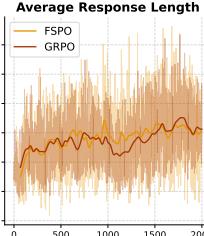


Figure 7: Average response length and factuality score for FSPO and GRPO.



**Factuality Improvement Analysis.** We further explore the benefit of incorporating step-wise factuality reward in our approach. The results in Figure 7 illustrate that both FSPO and GRPO generate responses of similar lengths. However, as our proposed FSPO approach explicitly optimizes for factual correctness during intermediate reasoning steps, it achieves significantly higher factuality scores compared to GRPO. This suggests that FSPO effectively enhances model reliability and factual accuracy without compromising the quality of the generated content.

## 7 Related Work

**Large Reasoning Models.** Recent advancements in large reasoning models have marked a significant shift toward explicitly improving complex reasoning, planning, and problem-solving capabilities [34]. OpenAI’s o1 [15] exemplifies the integration of RL with long CoT reasoning, enabling the model to iteratively refine its reasoning process before producing an answer. This approach significantly enhances its capability in complex problem-solving scenarios, particularly in mathematical reasoning and algorithmic tasks. DeepSeek-R1 [9] and Kimi k1.5 [17] similarly emphasize RL-based reasoning but introduce an open and flexible training pipeline. QwQ model [28] underscores the potential of smaller models when optimized specifically for reasoning tasks. Increasing efforts are devoted to reproducing prominent reasoning performance of these models [43, 27, 3]. However, existing research has predominantly concentrated on enhancing the reasoning capabilities of models, while overlooking the issue of hallucinations emerging within these models.

**Hallucination.** Hallucination has been a fundamental challenge in large language models (LLMs), receiving extensive attention in existing literature [13, 16, 46]. To investigate and detect hallucinations in LLMs, in scenarios where LLMs provide access to their internal states, researchers can closely examine their internals to uncover underlying mechanisms leading to hallucinations [36, 44]. Conversely, when the internal states of LLMs are inaccessible and they can only be queried through external API calls, researchers typically approach hallucination detection by examining correlations between input prompts and generated responses [24, 42]. To mitigate hallucinations, existing studies propose various strategies. During the pre-training phase, mitigation efforts typically revolve around curating and cleaning datasets [7, 35]. Following pre-training, fine-tuning methods such as supervised fine-tuning [37] and reinforcement learning with human feedback [26] are commonly employed to further mitigate hallucinations. Additionally, post-processing mitigation involves verifying the factual correctness of generated content using either the LLM itself [25] or external knowledge-based verification methods [1]. Our work is an early effort focusing on analyzing and mitigating hallucinations in large reasoning models.

## 8 Conclusion

In this paper, we have highlighted and theoretically explained a critical limitation associated with RL-based fine-tuning of LLMs, specifically the increased incidence of hallucinations in generated reasoning content. Our analysis attributed this issue to unstable training dynamics driven by high-gradient variance, entropy-induced randomness, and convergence toward spurious local optima inherent in traditional outcome-based RL training. To address these challenges, we introduced FSPO, a novel factuality-aware RL algorithm that integrates step-wise factual verification, providing a denser and more accurate feedback mechanism during training. Comprehensive experimental evaluations

demonstrate that FSPO significantly mitigates hallucinations across multiple benchmarks without compromising the models’ reasoning capabilities.

## 9 Acknowledgments

This research is supported by the National Research Foundation Singapore under its AI Singapore Programme (Award Number: AISG3-RP-2022-030).

## References

- [1] Anthony Chen, Panupong Pasupat, Sameer Singh, Hongrae Lee, and Kelvin Guu. PURR: Efficiently Editing Language Model Hallucinations by Denoising Language Model Corruptions. *arXiv preprint arXiv:2305.14908*, 2023.
- [2] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code. *CoRR*, abs/2107.03374, 2021.
- [3] Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, et al. An Empirical Study on Eliciting and Improving R1-like Reasoning Models. *arXiv preprint arXiv:2503.04548*, 2025.
- [4] Qinyuan Cheng, Tianxiang Sun, Wenwei Zhang, Siyin Wang, Xiangyang Liu, Mozhi Zhang, Junliang He, Mianqiu Huang, Zhangyue Yin, Kai Chen, and Xipeng Qiu. Evaluating Hallucinations in Chinese Large Language Models. *CoRR*, abs/2310.03368, 2023.
- [5] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [6] Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. The Danger of Overthinking: Examining the Reasoning-Action Dilemma in Agentic Tasks. *arXiv preprint arXiv:2502.08235*, 2025.
- [7] Souvik Das, Sougata Saha, and Rohini K Srihari. Diving Deep into Modes of Fact Hallucinations in Dialogue Systems. *arXiv preprint arXiv:2301.04449*, 2023.
- [8] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.
- [9] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [10] 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. *NeurIPS*, 2021.
- [11] 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*, December 2020.

[12] Jian Hu. REINFORCE++: A Simple and Efficient Approach for Aligning Large Language Models. *arXiv preprint arXiv:2501.03262*, 2025.

[13] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Trans. Inf. Syst.*, 43(2), January 2025. ISSN 1046-8188.

[14] Simon Hughes and Minseok Bae. Vectara Hallucination Leaderboard, 11 2023. URL <https://github.com/vectara/hallucination-leaderboard>.

[15] 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.

[16] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

[17] Kimi-Team. Kimi k1. 5: Scaling Reinforcement Learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025.

[18] Paweł Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. Exploration in Deep Reinforcement Learning: A Survey. *Information Fusion*, 85:1–22, 2022.

[19] Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6449–6464. Association for Computational Linguistics, 2023.

[20] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3214–3252. Association for Computational Linguistics, 2022.

[21] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 Technical Report. *arXiv preprint arXiv:2412.19437*, 2024.

[22] MAA. American Invitational Mathematics Examination - AIME, 2024.

[23] MAA. American Invitational Mathematics Examination - AIME, 2025.

[24] Potsawee Manakul, Adian Liusie, and Mark JF Gales. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. *arXiv preprint arXiv:2303.08896*, 2023.

[25] Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. Self-contradictory Hallucinations of Large Language Models: Evaluation, Detection and Mitigation. *arXiv preprint arXiv:2305.15852*, 2023.

[26] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[27] Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, et al. O1 Replication Journey: A Strategic Progress Report–Part 1. *arXiv preprint arXiv:2410.18982*, 2024.

[28] Qwen-Team. QwQ-32B: Embracing the Power of Reinforcement Learning, March 2025. URL <https://qwenlm.github.io/blog/qwq-32b/>.

[29] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[30] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024.

[31] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. *arXiv preprint arXiv: 2409.19256*, 2024.

[32] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[33] Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-Searcher: Incentivizing the Search Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2503.05592*, 2025.

[34] Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, et al. A Survey of Reasoning with Foundation Models. *arXiv preprint arXiv:2312.11562*, 2023.

[35] Logesh Kumar Umapathi, Ankit Pal, and Malaikannan Sankarasubbu. Med-HALT: Medical Domain Hallucination Test for Large Language Models. *arXiv preprint arXiv:2307.15343*, 2023.

[36] Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. A Stitch in Time Saves Nine: Detecting and Mitigating Hallucinations of LLMs by Validating Low-Confidence Generation. *arXiv preprint arXiv:2307.03987*, 2023.

[37] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning Language Models with Self-Generated Instructions. *arXiv preprint arXiv:2212.10560*, 2022.

[38] Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-RL: Unleashing LLM Reasoning with Rule-Based Reinforcement Learning. *arXiv preprint arXiv:2502.14768*, 2025.

[39] 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.

[40] Jiashu Yang, Ningning Wang, Yian Zhao, Chaoran Feng, Junjia Du, Hao Pang, Zhirui Fang, and Xuxin Cheng. Kongzi: A Historical Large Language Model with Fact Enhancement. *arXiv preprint arXiv:2504.09488*, 2025.

[41] 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, 2018.

[42] Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, and Li Yuan. LLM Lies: Hallucinations are not Bugs, but Features as Adversarial Examples. *arXiv preprint arXiv:2310.01469*, 2023.

[43] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. LIMO: Less is More for Reasoning. *arXiv preprint arXiv:2502.03387*, 2025.

[44] Mert Yuksekgonul, Varun Chandrasekaran, Erik Jones, Suriya Gunasekar, Ranjita Naik, Hamid Palangi, Ece Kamar, and Besmira Nushi. Attention Satisfies: A Constraint-Satisfaction Lens on Factual Errors of Language Models. *arXiv preprint arXiv:2309.15098*, 2023.

[45] Weihao Zeng, Yuzhen Huang, Wei Liu, Keqing He, Qian Liu, Zejun Ma, and Junxian He. 7B Model and 8K Examples: Emerging Reasoning with Reinforcement Learning is Both Effective and Efficient. <https://hkust-nlp.notion.site/simplerl-reason>, 2025. Notion Blog.

[46] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *CoRR*, abs/2309.01219, 2023. doi: 10.48550/ARXIV.2309.01219. URL <https://doi.org/10.48550/arXiv.2309.01219>.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer **[Yes]** , **[No]** , or **[NA]** .
- **[NA]** means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: **[Yes]**

Justification: The abstract and introduction sections accurately reflect the paper's contributions and scope.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: **[Yes]**

Justification: We discuss the limitations of our work at Appendix A.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: We have provided full assumptions and proof for our theoretical analysis in Section 4.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We have provided all dataset and implementation details in Section 6.1 and Appendix C and D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same

dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: We provide full details of our datasets at Appendix C.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: We provide all necessary training and test details in Section 6.1, Appendix D and the used prompt templates in Appendix F.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: In Section 6, we provide the experimental results that support the main claims of our paper and conduct some detailed analysis to examine some hyper-parameters in our approach.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide information about computer resources at Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Our paper follows the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We have discuss both potential positive societal impacts and negative societal impacts of our work at Appendix B.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work uses existing open-source models and datasets.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: **[Yes]**

Justification: We have cited the papers for our used datasets and models and mentioned their license in Section 6.1 and Appendix C.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](http://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

## 13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: **[NA]**

Justification: Our work uses existing datasets and models.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

## 14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: **[NA]**

Justification: We do not involve crowdsourcing and research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

**15. Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: We do not involve crowdsourcing and research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

**16. Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this paper does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

## Technical Appendices and Supplementary Material

### A Limitations

Our theoretical analysis is mainly focused on the prevalent and simplified binary reward modeling. Future work can extend it to any arbitrary bounded real rewards. The scope of our experiments is constrained by the substantial computational cost of much larger models (e.g., 14B, 32B). Future work can explore applying our approach to larger models.

### B Broader Impact

Our proposed FSPO algorithm has the potential to significantly improve the trustworthiness and reliability of LLMs in critical applications, such as education, healthcare, and scientific research, by reducing factual hallucinations during complex reasoning. However, the same technology may also pose negative societal risks if misused, for example, it could be applied to train persuasive yet selectively factual systems in domains such as political propaganda or misinformation campaigns, where step-wise coherence may mask underlying biases.

### C Datasets

To reduce hallucinations while improving the complex reasoning capabilities of LLMs, we mix two datasets for fine-tuning base models, i.e., a subset of HotpotQA [33] and SimpleRL [45]. Previous study [33] construct a subset from HotpotQA [41] and 2WikiMultiHopQA [11] with varying difficulty levels including 4761 and 3737 samples from HotpotQA and 2WikiMultiHopQA, respectively. Each sample includes the question, ground-truth answer, and the corresponding segments from Wikipedia. We randomly select 2K samples from this subset for conducting our proposed FSPO approach. SimpleRL consists of 8523 mathematical samples, which are used to perform traditional RL fine-tuning based on GRPO in our training. For hallucination datasets, TruthfulQA [20], HalluEval-QA [19], and HalluQA [4] contain 817, 10000, 450 samples, respectively. HalluQA is a Chinese dataset while the others are English datasets. For mathematical reasoning datasets, GSM-8K [5] and MATH-500 [10] contains 1319 and 500 samples, respectively. There are 30 samples in both AIME 2024 [22] and AIME 2025 [22]. These datasets are in MIT or Apache-2.0 license.

### D Training Infrastructure

We train our model on  $4 \times$ H100 80G GPU cards for about 24 hours each time. We train our model using the verl [31] framework. During training, we use a train batch size of 8, generate 8 rollouts for each prompt, set a maximum prompt and response length of 2,048 tokens, and train using a mini-batch size of 1,024. We use a constant learning rate of 4e-7 and temperature parameter of 1.0. The coefficient for KL loss is set to 1e-3 and the clip ratio is 0.2. During evaluation, we set the sampling temperature to 1.0 for our models and all baselines.

### E Preliminary Experiments

To demystify the question of “*To what extent do reasoning models maintain factuality after RL fine-tuning?*”, we conduct preliminary experiments on three hallucination benchmarks with twelve LLMs. The results show that after large-scale training with RL or long CoT data, models demonstrate significantly higher degree of hallucinations. To investigate the underlying reasons, we randomly select 200 samples (i.e., examples that DeepSeek-V3 answers correctly but DeepSeek-R1 does not) from HalluEval-QA and manually check their hallucination reasons. Specifically, the hallucination sources within our experiments can be roughly categorized into four types:

- *Intermediate Reasoning Error*: This is the most common source where the intermediate reasoning process is incorrect, which naturally results in wrong answers.
- *Reasoning and Answer Inconsistency*: This is the second most common source for which the intermediate reasoning process is correct but the final answer is wrong.

Table 2: Prompt for base and instruct models in mathematical reasoning tasks.

Model Type	Prompt
Base model	A conversation between User and Assistant. The user asks a question, and the assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the final answer. The reasoning process and answer are enclosed within <code>&lt;think&gt; &lt;/think&gt;</code> and <code>&lt;answer&gt; &lt;/answer&gt;</code> tags, respectively. For example, <code>&lt;think&gt;</code> reasoning process here <code>&lt;/think&gt;</code> <code>&lt;answer&gt;</code> answer here <code>&lt;/answer&gt;</code> . The Assistant shows the reasoning process within <code>&lt;think&gt; &lt;/think&gt;</code> tags, and ONLY return the FINAL ANSWER within <code>&lt;answer&gt; &lt;/answer&gt;</code> tags. For example: <code>&lt;answer&gt; 12 &lt;/answer&gt;</code> . \n\n User: {question} \n Assistant: Let me solve this step by step. <code>&lt;think&gt;</code>
Instruct model	You are a helpful assistant. Given a question, you need to first think about the reasoning process in the mind and then provide the final answer. The reasoning process and answer are enclosed within <code>&lt;think&gt; &lt;/think&gt;</code> and <code>&lt;answer&gt; &lt;/answer&gt;</code> tags, respectively. For example, <code>&lt;think&gt;</code> reasoning process here <code>&lt;/think&gt;</code> <code>&lt;answer&gt;</code> answer here <code>&lt;/answer&gt;</code> . You must show the reasoning process within <code>&lt;think&gt; &lt;/think&gt;</code> tags, and ONLY return the FINAL ANSWER within <code>&lt;answer&gt; &lt;/answer&gt;</code> tags. For example: <code>&lt;answer&gt; 12 &lt;/answer&gt;</code> .

Table 3: Prompt for base and instruct models in question answering tasks.

Model Type	Prompt
Base model	A conversation between User and Assistant. The user asks a question, and the assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the final answer. The reasoning process and answer are enclosed within <code>&lt;think&gt; &lt;/think&gt;</code> and <code>&lt;answer&gt; &lt;/answer&gt;</code> tags, respectively. For example, <code>&lt;think&gt;</code> reasoning process here <code>&lt;/think&gt;</code> <code>&lt;answer&gt;</code> answer here <code>&lt;/answer&gt;</code> . The Assistant shows the reasoning process within <code>&lt;think&gt; &lt;/think&gt;</code> tags, and ONLY return the FINAL ANSWER within <code>&lt;answer&gt; &lt;/answer&gt;</code> tags. For example: <code>&lt;answer&gt; Kim Marton &lt;/answer&gt;</code> . \n\n User: {question} \n Assistant: Let me solve this step by step. <code>&lt;think&gt;</code>
Instruct model	You are a helpful assistant. Given a question, you need to first think about the reasoning process in the mind and then provide the final answer. The reasoning process and answer are enclosed within <code>&lt;think&gt; &lt;/think&gt;</code> and <code>&lt;answer&gt; &lt;/answer&gt;</code> tags, respectively. For example, <code>&lt;think&gt;</code> reasoning process here <code>&lt;/think&gt;</code> <code>&lt;answer&gt;</code> answer here <code>&lt;/answer&gt;</code> . You must show the reasoning process within <code>&lt;think&gt; &lt;/think&gt;</code> tags, and ONLY return the FINAL ANSWER within <code>&lt;answer&gt; &lt;/answer&gt;</code> tags. For example: <code>&lt;answer&gt; Kim Marton &lt;/answer&gt;</code> .

- *Overthink*: This occurs when the intermediate reasoning process contains correct segments that can lead to the right answers but the model still continues to reason further, finally getting the wrong answers. This phenomenon is called “*overthink*”, investigated in extensive studies [6].
- *Others*: This includes the remaining cases, such as the output text is incomplete or a mismatch between generated answers and ground truth.

## F Prompt Templates

We present our prompts used during training and evaluation in Table 2, Table 3, and Table 4.

Table 4: Prompt for base and instruct models in HaluEval-QA dataset.

Model Type	Prompt
Base model	<p>A conversation between User and Assistant. The user asks a question, and the assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the final answer. The reasoning process and answer are enclosed within <code>&lt;think&gt; &lt;/think&gt;</code> and <code>&lt;answer&gt; &lt;/answer&gt;</code> tags, respectively. For example, <code>&lt;think&gt; reasoning process here &lt;/think&gt; &lt;answer&gt; answer here &lt;/answer&gt;</code>. Now the user asks a question and provides two candidate answers, the assistant needs to determine which answer is correct. The Assistant shows the reasoning process within <code>&lt;think&gt; &lt;/think&gt;</code> tags, and ONLY return the FINAL ANSWER within <code>&lt;answer&gt; &lt;/answer&gt;</code> tags. For example: <code>&lt;answer&gt; Kim Marton &lt;/answer&gt;</code>. \n\n User: #Question#: {question} \n #Candidate Answer 1#: {candidate_answer_1} \n #Candidate Answer 2#: {candidate_answer_2} \n Assistant: Let me solve this step by step. &lt;think&gt;</p>
Instruct model	<p>You are a helpful assistant. Given a question, you need to first think about the reasoning process in the mind and then provide the final answer. The reasoning process and answer are enclosed within <code>&lt;think&gt; &lt;/think&gt;</code> and <code>&lt;answer&gt; &lt;/answer&gt;</code> tags, respectively. For example, <code>&lt;think&gt; reasoning process here &lt;/think&gt; &lt;answer&gt; answer here &lt;/answer&gt;</code>. Now the user asks a question and provides two candidate answers, you need to determine which answer is correct. You must show the reasoning process within <code>&lt;think&gt; &lt;/think&gt;</code> tags, and ONLY return the FINAL ANSWER within <code>&lt;answer&gt; &lt;/answer&gt;</code> tags, such as <code>&lt;answer&gt; Kim Marton &lt;/answer&gt;</code>.</p>