Detection and Mitigation of Hallucination in Large Reasoning Models: A Mechanistic Perspective

Zhongxiang Sun **Qipeng Wang** **Haoyu Wang** **Xiao Zhang** **Jun Xu***
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
{sunzhongxiang}@ruc.edu.cn

Abstract

Large Reasoning Models (LRMs) have shown impressive capabilities in multi-step reasoning tasks. However, alongside these successes, a more deceptive form of model error has emerged—*Reasoning Hallucination*—where logically coherent but factually incorrect reasoning traces lead to persuasive yet faulty conclusions. Unlike traditional hallucinations, these errors are embedded within structured reasoning, making them more difficult to detect and potentially more harmful. In this work, we investigate reasoning hallucinations from a mechanistic perspective. We propose the **Reasoning Score**, which quantifies the depth of reasoning by measuring the divergence between logits obtained from projecting late layers of LRMs to the vocabulary space, effectively distinguishing shallow pattern-matching from genuine deep reasoning. Using this score, we conduct an in-depth analysis on the ReTruthQA dataset and identify two key reasoning hallucination patterns: early-stage fluctuation in reasoning depth and incorrect backtracking to flawed prior steps. These insights motivate our **R**easoning **H**allucination **D**etection (**RHD**) framework, which achieves state-of-the-art performance across multiple domains. To mitigate reasoning hallucinations, we further introduce **GRPO-R**, an enhanced reinforcement learning algorithm that incorporates step-level deep reasoning rewards via potential-based shaping. Our theoretical analysis establishes stronger generalization guarantees, and experiments demonstrate improved reasoning quality and reduced hallucination rates.

1 Introduction

Hallucination has long been a critical safety challenge for Large Language Models (LLMs). In this context, hallucination refers to outputs that appear fluent and coherent but are semantically inaccurate or lack factual grounding. With the advent of Large Reasoning Models (LRMs)—such as DeepSeek-R1 [10] and OpenAI's O-series [36]—AI systems have demonstrated unprecedented potential in solving complex real-world tasks. These models are typically trained with outcome-based reinforcement learning (RL) and explicitly generate multi-step reasoning traces prior to final answers.

Recent studies have uncovered a subtler form of hallucination emerging in LRMs [45, 48, 36], which we refer to as **Reasoning Hallucination**. Unlike traditional hallucinations, reasoning hallucinations are often embedded within logically coherent reasoning traces, making incorrect information more persuasive and harder to detect. This form of "plausible but incorrect" reasoning can elicit user trust, resembling the conjunction fallacy, where detailed yet misleading explanations are perceived as more credible than simpler ones [44, 47]. Prior studies mainly assess the correctness of reasoning paths in standard Chain-of-Thought (CoT) tasks over relatively simple problems [52, 37], with limited

*Corresponding author. Work partially done at Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: ResponsibleFM.

investigation into the mechanisms of hallucinations in LRMs. Recent work has extended evaluation to long CoT generated by LRMs [59], yet remains focused on error identification rather than uncovering underlying causes. However, directly analyzing model-generated traces can be misleading due to the subtle nature of reasoning hallucinations. The emergence of Latent CoT, where reasoning is embedded in hidden states rather than surface text, further obscures detection [17]. These challenges call for probing the internal mechanisms behind reasoning hallucinations, enabling interpretable and robust hallucination detection.

Recent studies on the reasoning capabilities of LRMs [32, 53] have shown that models often produce incorrect answers when their reasoning process relies on shallow pattern-matching rather than genuine deep reasoning. This mirrors findings in cognitive science, where human thinking patterns are closely linked to the emergence of cognitive illusions [22, 4, 50]. Inspired by these observations, we investigate reasoning hallucinations in LRMs through the lens of internal thinking patterns, where a central challenge is how to quantify whether a model is performing deep reasoning or merely matching surface-level patterns from training data. Prior mechanistic interpretability studies highlight a functional division within language models: early layers primarily transmit information, while later layers perform more complex reasoning over aggregated context [34, 6]. Based on this insight, we introduce **Reasoning Score**, which measures the divergence between logits obtained from projecting late layers of LRMs to the vocabulary space. Through synthetic experiments, we validate the effectiveness of the Reasoning Score in measuring the depth of reasoning in LRMs, which reflects whether the model engages in shallow pattern-matching or deep reasoning (§ 3.1).

Building on the proposed reasoning score, we conduct extensive analyses on reasoning hallucinations using the ReTruthQA dataset. We identify three key patterns of reasoning hallucination: **Pattern #1**: large fluctuations in reasoning depth during the early steps, and **Pattern #2**: incorrect backtracking from later steps to earlier incorrect steps. We attribute these patterns to the presence of shallow pattern-matching and overthinking steps, which undermine the LRM's inherent abilities in self-verification and backtracking, ultimately leading to reasoning hallucinations (§ 3.2). Moreover, we observe that **Pattern #3:** overthinking steps exhibit a positive correlation between reasoning scores and perplexity, indicating spurious verification behaviors (§ 3.3). Based on these findings, we design the **R**easoning **H**allucination **D**etection (**RHD**) method, which significantly outperforms baselines across diverse domains in the reasoning hallucination detection dataset (§ 4.1).

We further investigate the underlying cause of shallow pattern-matching and overthinking steps in LRMs and attribute it to the outcome-based RL paradigm commonly used during training. This paradigm incentivizes correct final answers but neglects whether intermediate reasoning steps reflect deep and meaningful thinking. To address this challenge, we introduce a step-level deep reasoning reward based on the reasoning score and propose **GRPO-R**, a variant of Group Relative Policy Optimization (GRPO) [41, 10] that incorporates potential-based reward shaping. GRPO-R encourages deep—but not excessive—reasoning during RL fine-tuning. Our theoretical analysis shows that GRPO-R leads to better generalization in outcome-based RL, and empirical results confirm that it improves reasoning accuracy compared to standard GRPO (§ 4.2).

2 Related Works

Hallucination of Language Models. Hallucination remains a fundamental safety concern for LLMs, and outcome-supervised LRMs [10, 36] exacerbate this issue by generating logically flawed but persuasive reasoning traces, a consequence of reward-seeking behavior induced by outcome-based RL without step-level supervision [7, 47, 45]. Detection approaches span uncertainty estimation [21, 30, 40], internal signal probing [5, 25, 24], process-level critique models [18], and Process Reward Models (PRMs) [58], though challenges remain due to the deceptive nature of hallucinated traces and the poor generalization of PRM signals [60]. We address this by conducting a mechanistic analysis of reasoning hallucinations and proposing a detection method grounded in internal model behavior.

Mechanistic Interpretability. Mechanistic interpretability [13, 9] seeks to explain model behavior by attributing predictions to internal components. In transformers, attention heads contextualize token representations [12, 51], while FFNs act as knowledge storage [15]. Recent work has applied intervention-based techniques from mechanistic interpretability to analyze how LLMs perform reasoning, revealing a functional division of labor across layers in various tasks such as math reasoning and multimodal reasoning: early layers primarily transmit contextual information, and

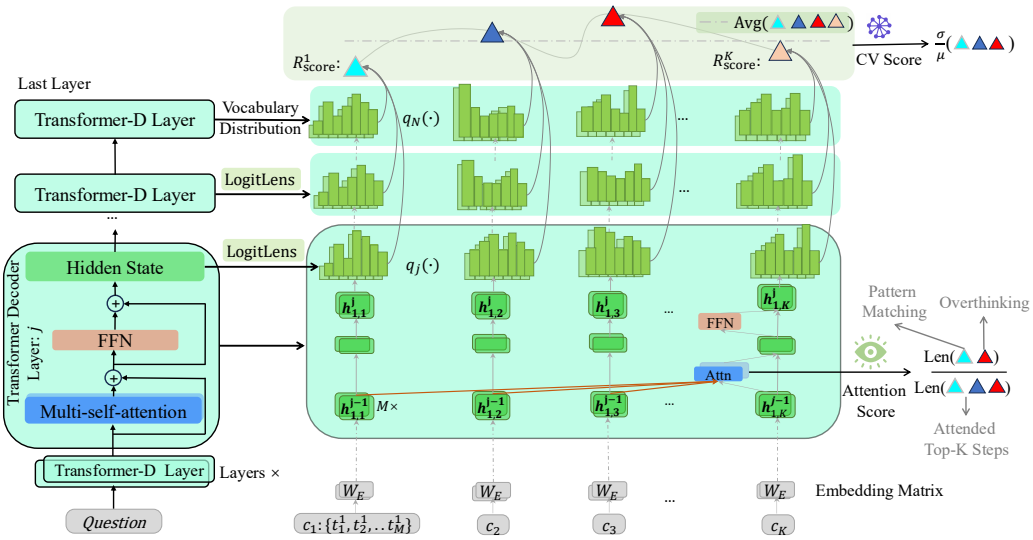


Figure 1: The illustration of the calculation processes for the Reasoning Score (Eq. 2), CV Score (Eq. 3), and Attention Score (Eq. 4).

the reasoning process is predominantly carried out by the later layers. [6, 34, 26]. These insights motivate our design of the *Reasoning Score*, which captures thinking patterns by quantifying hidden state shifts in later layers, laying the groundwork for analyzing reasoning hallucinations in LRMs.

3 Empirical Study of Reasoning Hallucination

Our empirical study investigates the relationship between reasoning hallucinations and the thinking patterns of LRMs, where thinking patterns are quantified using a reasoning score derived from mechanistic interpretability. This analysis reveals key reasoning hallucination patterns and guides the design of more effective detection and mitigation strategies.

3.1 Reasoning Score: Measuring Reasoning Depth in Large Reasoning Model

To determine whether a reasoning step is generated via shallow pattern matching or genuine deep reasoning, we propose a *Reasoning Score* inspired by mechanistic interpretability. Prior studies analyzing the internal mechanisms of language models reveal a layered functional division: early layers primarily transmit information, while later layers perform more complex reasoning over aggregated context to produce correct outputs [42, 34, 26]. Building on this insight, we define the reasoning score under the hypothesis that deeper reasoning is reflected by meaningful transformations in later-layer representations during generation.

Formally, a LRM-generated reasoning trace $C = [c_1, c_2, \ldots, c_K]$ consists of multiple reasoning steps, each associated with a step-level reasoning score R^k_{score} that quantifies the depth of reasoning in step c_k. Each reasoning step $c_k = \langle t^k_1, \ldots, t^k_M \rangle$ is composed of M tokens. The overall reasoning trace score $\mathcal{R}_{\text{score}}$ is represented as a sequence $[R^1_{\text{score}}, R^2_{\text{score}}, \ldots, R^K_{\text{score}}]$, capturing the model's reasoning dynamics across steps. As shown in Figure 1, each score is defined as the mean Jensen–Shannon divergence (JSD) between vocabulary distributions induced by hidden states from selected later layers and the anchor distribution from the final layer. To obtain the output distribution from each token hidden state $h^{(j)}_{m,k}$ of token t^k_m at layer j, we apply the LogitLens [35], which projects each layer-normalized hidden state into vocabulary space via the unembedding matrix $\boldsymbol{W}_U$: $\text{LogitLens}(h^{(j)}_{m,k}) = \text{LayerNorm}(h^{(j)}_{m,k})\boldsymbol{W}_U$. This provides a layer-wise interpretation of token prediction behavior and has been widely adopted for interpreting LLM internal representations [16, 61, 56].

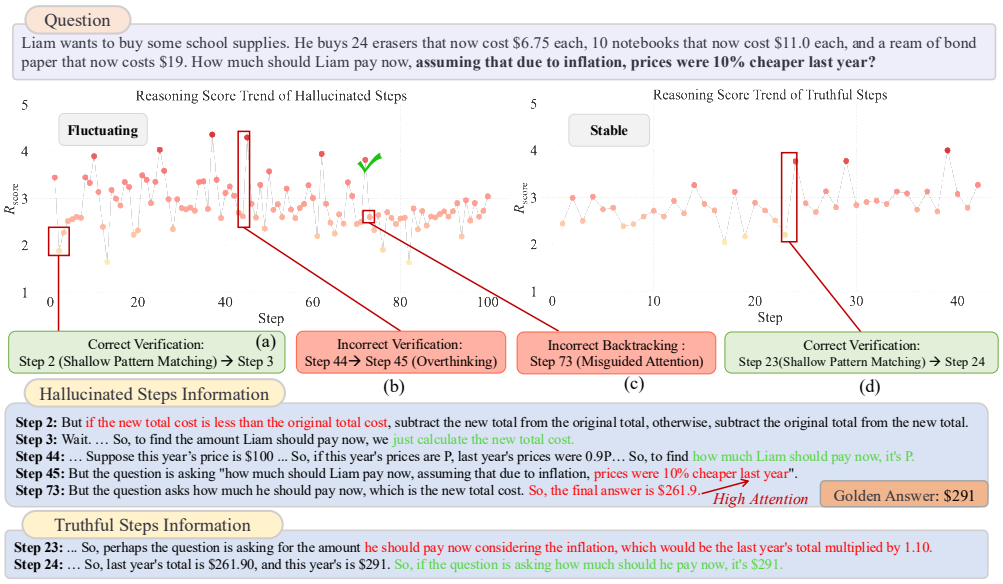


Figure 2: Case study from GSM-NoOp dataset [32] on R1-7B. We sample both a hallucinated reasoning trace (left) and a truthful reasoning trace (right) for the same question as a preliminary analysis of reasoning hallucinations. Reasoning scores are scaled by 1e5.

The final step-level Reasoning Score R^k_{score} is computed as:

$$R^k_{\text{score}} = \frac{1}{|c_k|} \sum_{t^k_{m+1} \in c_k} \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \text{JSD}\left(q_N\left(t^k_{m+1}\right), q_j\left(t^k_{m+1}\right)\right), \tag{1}$$

$$q_j\left(t^k_{m+1}\right) = \text{softmax}\left(\text{LogitLens}\left(h^{(j)}_{m,k}\right)\right), \quad j \in \mathcal{J}, \tag{2}$$

where $\mathcal{J}$ denotes the set of selected later layers and q_N is the anchor distribution from the final layer.

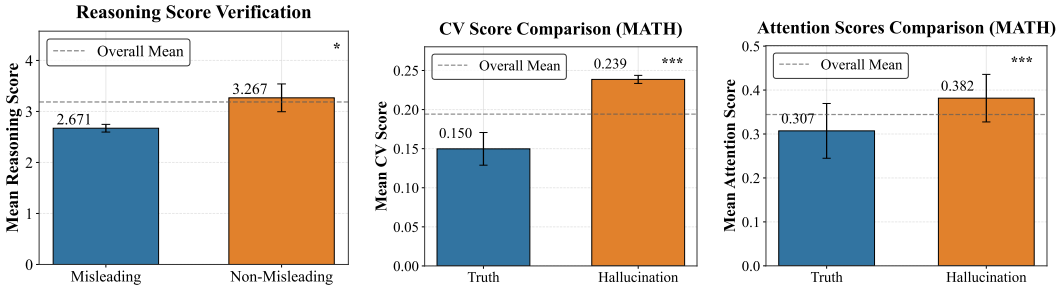
Intuitively, a larger score R_{score} indicates substantial transformation in output distributions within late layers, suggesting the model is actively engaging in deep reasoning by integrating earlier contextual information. In contrast, a smaller score implies distributional stability in late layers, indicating shallow pattern matching or heuristic-based processing without further reasoning, consistent with prior findings on the differential roles of early versus later layers.

Validating the Reasoning Score with GSM-NoOp. We validate whether the Reasoning Score faithfully reflects reasoning depth using GSM-NoOp [32], a GSM8K-derived dataset where semantically irrelevant but plausible `No-Op` phrases are injected into problems. Although these phrases do not alter the correct reasoning path, prior work shows that LRMs are often misled by them, revealing their reliance on shallow pattern matching [32]. This makes GSM-NoOp a suitable testbed: if the Reasoning Score captures reasoning depth, then steps misled by No-Op phrases should yield lower scores. We validate this using correct outputs from `DeepSeek-R1-Distill-Qwen-7B (R1-7B)` to avoid confounds from hallucinated traces. Misled steps are labeled via GPT-4o. As GSM-NoOp is not publicly available, we re-implement a compatible version following the original paper's methodology, with prompts and details provided in Appendix D.

Results. Our empirical results in Figure 3 (a) show that reasoning steps misled by No-Op phrases consistently receive significantly lower Reasoning Scores compared to non-misled steps. This supports our hypothesis that the Reasoning Score effectively captures shallow pattern-matching behavior and serves as an indicator of whether a model is engaging in deep reasoning.

3.2 Reasoning Hallucination Analysis Based on Reasoning Score

In this section, we leverage the mechanistically derived Reasoning Score as a proxy for the thinking patterns of LRMs and investigate its relationship with the emergence of reasoning hallucinations. We



(a) (b) (c)

Figure 3: (a) Reasoning Score validation on GSM-NoOp. (b) Evaluation of Pattern #1 (early fluctuations), and (c) Pattern #2 (misguidedly attention) on ReTruthQA. Asterisks indicate statistical significance based on a t-test: * for p-value < 0.05, and *** for p-value < 0.001.

begin with a preliminary analysis to identify characteristic patterns associated with hallucinated reasoning traces. We then analyze the generality of these patterns across domains using the ReTruthQA dataset, and further examine the underlying mechanism that leads LRMs to exhibit such behaviors.

3.2.1 Case Analysis on GSM-NoOps

In this section, we conduct a preliminary analysis using the LRM R1-7B on a question from GSM-NoOp [32], where a "NoOp" statement is appended to the end of a math problem. To enable controlled comparison of reasoning hallucination patterns, we sample both a truthful and a hallucinated response from R1-7B on the same question. Figure 2 presents the question along with step-level reasoning scores R_{score}, which quantify the depth of thinking at each step.

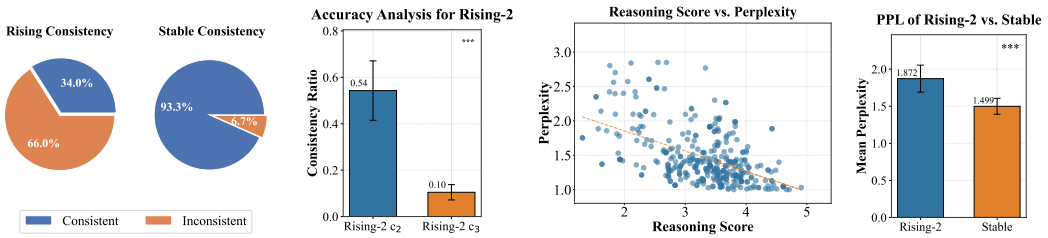
We observe that when the model generates reasoning steps that attend to the added NoOp content, these steps typically receive lower R_{score}, which in turn triggers the model's *Self-Verification* mechanism [23], producing later steps with higher R_{score} that attempt to correct the earlier deviation (e.g., (a) and (d) in Figure 2). However, in the hallucinated reasoning trace, we also observe *overthinking* phenomena—steps with excessively high R_{score} that incorrectly revise the previous correct reasoning steps (e.g., (b) in Figure 2). These hallucinated traces contain more shallow pattern-matching and overthinking steps, resulting in an overall unstable reasoning trajectory. From this case study, we identify the reasoning hallucination **Pattern #1**: hallucinated traces typically exhibit large fluctuations in reasoning score, especially during the early steps of the process.

Furthermore, we observe that even when the model briefly arrives at correct intermediate steps, it often fails to maintain this correctness. In later steps, it performs *Incorrect Backtracking*, attending to earlier shallow or overthinking steps, ultimately leading to hallucination (e.g., (c) in Figure 2). This motivates the reasoning hallucination **Pattern #2**: in the later stages of reasoning, the model tends to misguidedly attend to earlier hallucinated steps, either shallow or overthinking, making it difficult to correct earlier errors and leading to hallucinated reasoning.

3.2.2 Reasoning Hallucination Pattern Analysis

In this section, we validate the two reasoning hallucination patterns identified in preliminary analysis(§ 3.2.1): **Pattern #1**: large fluctuations in reasoning scores during early steps, and **Pattern #2**: incorrect backtracking to earlier hallucinated reasoning steps in later stages. We aim to assess whether these patterns generalize across broader domains and tasks. To this end, we conduct experiments on the ReTruthQA dataset using the R1-7B model. ReTruthQA covers three reasoning domains: `Math`, `Science`, and `MultiHopQA` (Details in § 5.1). For each domain, we construct two balanced subsets using gold hallucination labels: one with hallucinated traces and one with truthful traces.

To evaluate **Pattern #1**, we measure the fluctuation of reasoning depth in the early phase of reasoning using the Coefficient of Variation (**CV Score**) [11], a standard metric for quantifying sequence variability (shown in Figure 1). Specifically, we focus on the first $\lceil K/r \rceil$ steps of the reasoning trace $\mathcal{C} = \langle c_1, c_2, \ldots, c_K \rangle$, and define: $\mathcal{R}_{\text{score}}^{\text{early}} = \left[R_{\text{score}}^1, R_{\text{score}}^2, \ldots, R_{\text{score}}^{\lceil K/r \rceil} \right]$, where $r > 1$ is a constant controlling the size of the early-step window. The CV score over early reasoning steps is then given



(a) (b) (c) (d)

Figure 4: Analysis of Pattern #1: (a) Consistency Analysis (**Q1**); (b) Accuracy Comparison in Rising-2 triples (**Q2**); (c) Reasoning score vs. perplexity and (d) Perplexity of Rising-2 vs. Stable (**Q3**).

by:

$$\mathrm{CV}(\mathcal{C}) = \frac{\sigma(\mathcal{R}_{\text{score}}^{\text{early}})}{\mu(\mathcal{R}_{\text{score}}^{\text{early}})}, \tag{3}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and standard deviation, respectively.

To assess **Pattern #2**, we introduce a **Attention Score** that quantifies the extent to which later reasoning steps attend to earlier shallow-pattern matching or overthinking steps (Figure 1). Let the full reasoning trace be $\mathcal{C} = \langle c_1, c_2, \ldots, c_K \rangle$, and define the later reasoning steps as $\mathcal{C}_{\text{later}} = \{c_k\}_{k=\lceil \eta K \rceil}^{K}$. For a step $c_k \in \mathcal{C}_{\text{later}}$, we compute the mean attention from c_k to each earlier step c_j as:

$$\bar{a}_{k \to j} = \frac{1}{|c_k||c_j|} \sum_{t \in c_k} \sum_{s \in c_j} \left(\frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \frac{1}{H} \sum_{h=1}^{H} a_{t,s}^{l,h} \right),$$

where $a_{t,s}^{l,h}$ denotes the attention weight from token t to token s at head h in layer l, H is the number of heads per layer, $\mathcal{L}$ is the set of selected layers for aggregation, and the constant η defines late steps.

We then identify the top-K most attended earlier steps based on $\bar{a}_{k \to j}$: $\mathcal{T}_k = \mathrm{TopK}\left(\{\bar{a}_{k \to j}\}_{j=1}^{k-1}, K\right)$, where $\mathcal{T}_k$ is the set of indices corresponding to the top-attended steps. The step-level attention score for c_k is then defined as the proportion of these steps whose Reasoning Scores fall outside the normal range, either in the lower quartile or exceeding a high threshold τ:

$$\mathrm{AttnScore}(c_k) = \frac{1}{K} \sum_{j \in \mathcal{T}_k} \mathbb{1}_{\left(R_{\text{score}}^{j} \leq \mathrm{Quantile}_{1/4}(\mathcal{R}_{\text{score}}) \text{ or } R_{\text{score}}^{j} \geq \tau\right)},$$

where $\mathbb{1}_{(\cdot)}$ is the indicator function, $\mathrm{Quantile}_{1/4}(\mathcal{R}_{\text{score}})$ denotes the first quartile of the reasoning scores (i.e., potentially shallow pattern-matching steps), and τ is a threshold identifying potentially overthinking steps.

The trace-level attention score is computed by averaging over all later steps:

$$\mathrm{AttnScore}(\mathcal{C}) = \frac{1}{|\mathcal{C}_{\text{later}}|} \sum_{c_k \in \mathcal{C}_{\text{later}}} \mathrm{AttnScore}(c_k), \tag{4}$$

which reflects the extent to which later reasoning steps attend to earlier incorrect steps.

Results. As shown in Figure 3(b) and (c) and Appendix G, across all three domains, hallucinated reasoning traces consistently yield significantly higher CV scores and Attention scores than truthful traces. This confirms that hallucinated traces are more fluctuating in reasoning depth (Pattern #1) and more likely to attend prior incorrect steps (Pattern #2), demonstrating the generalizability of both patterns beyond the initial case study (Section 3.2.1). Detailed settings are shown in Appendix G.

3.3 Analyzing the Mechanisms Behind Reasoning Fluctuation

We investigate the underlying mechanism behind **Pattern #1**, where hallucinated reasoning traces exhibit large fluctuations in reasoning depth. Building on our case study in Section 3.2.1, we hypothesize this stems from a built-in self-verification mechanism. Key questions still include: **Q1**:

What triggers verification behavior in LRMs? **Q2**: Do excessively high reasoning scores reliably signal overthinking? **Q3**: If Q2 holds, what factors lead to the emergence of such overthinking steps?

To explore these, we construct step triples (c_1, c_2, c_3) from reasoning traces: (1) **Stable** triples with minimal score variation from truthful traces; (2) **Rising-1** triples from hallucinated traces with a moderate score spike ($R_{\text{score}}(c_3) < 4$), potentially triggered by shallow pattern-matching in c_2; and (3) **Rising-2** triples with extreme score spikes ($R_{\text{score}}(c_3) > 4$), to probe overthinking behaviors.

Analysis. For **Q1**, we compare the logical consistency between c_1 and c_2 in Rising vs. Stable triples using GPT-4o judgments. As shown in Figure 4(a), stable triples show significantly higher consistency, suggesting that verification is more likely to be triggered when earlier steps are inconsistent.

Regarding **Q2**, we assess the accuracy of c_2 and c_3 in Rising-2 triples. Figure 4(b) shows that while c_2 is often correct, c_3 introduces errors, confirming that excessively high reasoning scores reliably signal overthinking. Prompts of **Q1** and **Q2** are shown in Appendix F.

To investigate **Q3**, we firstly analyze the correlation between reasoning depth and perplexity. As shown in Figure 4(c), reasoning steps with higher R_{score} generally exhibit lower perplexity, indicating more certainty outputs. However, Figure 4(d) reveals that in Rising-2 triples, c_3 steps, despite higher reasoning scores, have higher perplexity than those in stable triples, suggesting that overthinking may produce internally unstable generations. We term this phenomenon *spurious verification*, where the model performs misguided validation driven by outcome-based reward optimization. This insight leads us to identify a new hallucination pattern: **Pattern #3:** Overthinking steps exhibit a positive correlation between R_{score} and perplexity. More details are provided in Appendix E.

4 Methods

4.1 Reasoning Hallucination Detection

Building upon the patterns uncovered in our empirical study, we propose the **R**easoning **H**allucination **D**etection algorithm (**RHD**). Our approach leverages the step-level Reasoning Score R_{score} to quantify thinking depth throughout the reasoning trace, and incorporates three identified indicators of hallucination: (1) Pattern #1: large fluctuations in reasoning scores during early steps, (2) Pattern #2: incorrect backtracking to earlier shallow or overthinking steps in later stages, and (3) Pattern #3: overthinking behavior where R_{score} and perplexity exhibit a positive correlation.

Given a question Q and its reasoning trace $\mathcal{C}$ with step-level scores $\mathcal{R}_{\text{score}}$, we define the overall Reasoning Hallucination Score as:

$$\mathcal{H}_{\mathcal{C}} = \underbrace{\alpha_1 \cdot \mathrm{Avg}(\mathcal{R}_{\text{score}})}_{\text{Overall Reasoning Depth}} + \underbrace{\alpha_2 \cdot \mathrm{CV}(\mathcal{C})}_{\text{Pattern \#1}} + \underbrace{\alpha_3 \cdot \mathrm{AttnScore}(\mathcal{C})}_{\text{Pattern \#2}} + \underbrace{\alpha_4 \cdot \mathrm{PCC}(\mathcal{R}_{\text{score}}, \mathrm{PPL}(\mathcal{C}))}_{\text{Pattern \#3}}, \quad (5)$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \geq 0$ are regression coefficients. Avg denotes the average reasoning score, CV (Eq. 3) measures fluctuations during early-steps, $\mathrm{AttnScore}$ (Eq. 4) captures attention on earlier hallucinated steps, and PCC refers to the Pearson correlation coefficient between reasoning scores and step-level perplexity $\mathrm{PPL}(\mathcal{C})$, computed according to Eq. 11.

4.2 Mitigating Reasoning Hallucinations via Step-Level Reasoning Score Shaping

Reasoning hallucinations often stem from two types of flawed steps: (1) shallow pattern-matching, reflecting shortcut behaviors, and (2) overthinking, induced by excessive and misguided verification. A core factor is outcome-based RL, which only rewards the final answer and neglects intermediate steps [7, 47, 45], encouraging reward-hacking heuristics that may propagate through distillation [49].

To address this, we introduce an auxiliary process-level reward based on the **reasoning score** R_{score} from Section 3.1, which measures the reasoning depth at each step. This encourages meaningful reasoning while penalizing shallow or overthinking steps. We model the reasoning process as a finite-horizon MDP $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $s_t \in \mathcal{S}$ is the reasoning state at step t, $a_t \in \mathcal{A}$ denotes the next reasoning step, P is the transition probability and r_t is the reward:

$$r_t = \begin{cases} 0, & t < T, \\ R_{\text{final}}, & t = T. \end{cases}$$

Reward Shaping with Reasoning Score. We apply potential-based reward shaping [33]:

$$\bar{r}_t = r_t + \gamma\Phi(s_{t+1}) - \Phi(s_t), \quad \text{with } \Phi(s_T) = 0,$$

which preserves the optimal policy while redistributing credit: $V'(s_t) = V(s_t) - \Phi(s_t)$, where $V(s_t) = \mathbb{E}_\pi\left[\sum_{k=t}^{T}\gamma^{k-t}r_k \,\middle|\, s_t\right]$ is the value function of original reward and $V'(s_t)$ is the shaped.

Potential Function Design. To avoid encouraging overthinking, we clip the reasoning score:

$$\tilde{R}_{\text{score}}(s_t) = \begin{cases} \alpha \cdot R_{\text{score}}(s_t), & R_{\text{score}}(s_t) \le \tau, \\ 0, & \text{otherwise}, \end{cases} \qquad \Phi(s_t) = -\tilde{R}_{\text{score}}(s_t),$$

where $\alpha > 0$ and τ control the weighting strength and the threshold for overthinking, respectively.

To understand the generalization benefit of our proposed reasoning score–based shaping, we derive a uniform convergence bound under augmented rewards:

Theorem 1 (Generalization Gap with Augmented Rewards)**.** *Let the policy class Π be such that for any $\pi \in \Pi$, the augmented return $R(\pi, \xi) = \sum_{t=1}^{T}\gamma^{t-1}\bar{r}_t(\xi)$ is uniformly bounded in $[0, \bar{R}_{\max}]$ for any trajectory ξ sampled from the environment. Each trajectory $\xi = (s_1, a_1, \bar{r}_1, \ldots, s_T, a_T, \bar{r}_T)$ denotes a complete multi-step reasoning trace. Suppose that Π has Rademacher complexity $\mathcal{R}_n(\Pi)$ based on n independent training samples $\{\xi_i\}_{i=1}^{n}$. Then, with probability at least $1-\delta$, for any $\pi \in \Pi$ the following holds:*

$$J_{test}(\pi) - J_{train}(\pi) \le 2\bar{R}_{\max}\,\mathcal{R}_n(\Pi) + \bar{R}_{\max}\sqrt{\frac{\log(1/\delta)}{2n}},$$

where $J_{test}(\pi) = \mathbb{E}_\xi[R(\pi,\xi)]$ is the expected test return and $J_{train}(\pi) = \frac{1}{n}\sum_{i=1}^{n}R(\pi,\xi_i)$ is the empirical training return.

The proof is given in Appendix A. Intuitively, our reasoning score acts as a regularizer that encourages logically consistent behaviors and effectively reduces the Rademacher complexity $\mathcal{R}_n(\Pi)$, thereby tightening the bound and improving generalization to unseen reasoning tasks.

Integrate into GRPO. To demonstrate compatibility with standard RL algorithms, we integrate the reasoning score shaping framework into the Group Relative Policy Optimization (GRPO), a scalable and widely used RL algorithm for reasoning model training [10, 41], yielding **GRPO-R**. All implementation and formulation details of GRPO-R are provided in Appendix B.

5 Experiments

5.1 Reasoning Hallucination Detection

Data and Evaluation. We evaluate our RHD method on the **ReTruthQA** dataset spanning three reasoning domains: `Math`, `Science`, and `MultiHopQA` (construction details in Appendix C). We adopt two evaluation settings: (1) **Binary Detection**, which assesses the model's ability to detect hallucinations in individual (Q, C) pairs using AUC and PCC; (2) **Multi-Trace Ranking**, which evaluates whether the model can rank truthful traces higher among multiple candidates $(Q, \{C_1, ..., C_N\})$, following TruthfulQA-MC [28]. We report MC1, MC2, and MC3 to measure hallucination ranking accuracy (Evaluation details are in Appendix H).

Models and Baselines. We conduct experiments on two open-source LRM: `DeepSeek-R1-Distill-Qwen-7B` (R1-7B) and `DeepSeek-R1-Distill-Qwen-14B` (R1-14B) [10]. We compare our method against six categories of hallucination detection baselines: (1) Ensemble based self-evaluation (e.g., ChainPoll [14]); (2) Uncertainty based methods (e.g., P(True) [21]); (3) Self-Awareness based approaches (e.g., UQAC [25]); (4) LLM-as-Critic (LCM) models (e.g., `GPT-4o`); (5) Process Reward Models (PRMs) with step-level supervision (e.g., `Qwen2.5-Math-PRM`); (6) Length-based scoring, which uses trace length as a proxy for hallucination likelihood. Baselines and RHD implementation details are provided in Appendix H and I.

Main Results. As shown in Table 1, RHD consistently outperforms most baselines across all ReTruthQA domains, model backbones, and evaluation settings, demonstrating strong robustness.

Table 1: Performance comparisons between RHD and baselines for Reasoning Hallucination Detection. The boldface represents the best performance, and the underline represents the second-best.

LRMs	Categories	Methods	ReTruthQA (MATH)					ReTruthQA (Science)					ReTruthQA (MultiHopQA)				
			AUC	PCC	MC1	MC2	MC3	AUC	PCC	MC1	MC2	MC3	AUC	PCC	MC1	MC2	MC3
R1-7B	Ensemble	ChainPoll [14]	0.6384	0.2603	0.3020	0.2952	0.3583	0.6468	0.2612	0.2700	0.2580	0.3098	0.6297	0.2233	0.4208	0.3019	0.3954
		LMvLM [8]	0.6364	0.3728	0.3204	0.2504	0.3402	0.5345	0.1890	0.2600	0.2100	0.3113	0.6331	0.2759	0.3649	0.3049	0.3984
		SelfCheckGPT [31]	0.7727	0.4598	0.4091	0.2784	0.4119	0.6819	**0.2669**	0.3793	0.3655	0.5320	0.6886	0.2955	0.2553	0.1915	0.3118
	Uncertainty	P(True) [21]	0.7216	0.2681	0.5455	0.4068	0.5182	0.6207	0.2572	0.5172	0.4276	0.5533	0.5400	0.1684	0.4026	0.3030	0.4032
		LN-Entropy [40]	0.6896	0.3099	0.5000	0.3917	0.5096	0.5553	0.1129	0.3700	0.3200	0.4329	0.6123	0.2149	0.4156	0.3208	0.4461
		PPL [30]	0.7025	0.2856	0.5909	0.4205	0.5267	0.5434	0.1144	0.3793	0.3034	0.3990	0.6432	0.2249	0.5745	0.4532	0.5241
	Length	Length-Score [57]	0.5351	0.0922	0.4318	0.2568	0.3408	0.5510	0.0911	0.5793	0.5034	0.5737	0.5815	0.1496	0.5106	0.3887	0.4674
	PRM	Qwen2.5-PRM800K [59]	0.6601	0.2746	0.4773	0.3000	0.4572	0.6153	0.2203	0.4400	0.3605	0.4444	0.5694	0.1074	0.5065	0.4167	0.4990
		Qwen2.5-PRM-7B [58]	0.5563	0.1354	0.4318	0.2701	0.3913	0.5690	0.1275	0.2200	0.1425	0.2382	0.5422	0.0866	0.4026	0.2952	0.3947
	LCM	GPT4-o [1]	0.7513	0.3794	0.4091	0.2705	0.4131	**0.7045**	0.2026	0.2500	0.2965	0.3200	0.7123	0.2204	0.4043	0.2830	0.3704
		Qwen2.5-32B [54]	0.6942	0.2082	0.2500	0.1955	0.2935	0.6525	0.2635	0.3103	0.2897	0.4458	0.6424	0.2056	0.4400	0.3300	0.4187
	Self-Aware	UQAC [25]	0.6671	0.2902	0.5833	0.3715	0.5298	0.6303	0.2369	0.4700	0.3925	0.4885	0.6736	0.2583	0.6623	0.5335	0.6425
		EigenScore [5]	0.7539	0.3868	0.4583	0.3250	0.3007	0.6488	0.2601	0.4260	0.3777	0.3815	0.6696	0.2858	0.5195	0.4113	0.3885
	Ours	RHD	**0.7978**	**0.4852**	**0.6591**	**0.4765**	**0.5699**	0.6528	0.2662	**0.6207**	**0.5448**	**0.6009**	**0.7361**	**0.3863**	**0.7660**	**0.6255**	**0.7103**
R1-14B	Ensemble	ChainPoll [14]	0.5858	0.1658	0.2704	0.2535	0.3394	0.6640	0.3134	0.3261	0.1775	0.2188	0.5846	0.1607	0.2319	0.1972	0.2638
		LMvLM [8]	0.6620	**0.3835**	0.2563	0.2507	0.3133	0.5435	0.2132	0.3333	0.2300	0.3421	0.6250	0.2914	0.2042	0.1885	0.2506
		SelfCheckGPT [31]	0.5714	0.2774	0.2462	0.2167	0.2930	0.5109	0.1048	0.3287	0.2566	0.3683	0.5208	0.1268	0.3167	0.3083	0.0320
	Uncertainty	P(True) [21]	0.6460	0.1443	0.2615	0.2374	0.4570	0.6645	0.2582	0.4828	0.3460	0.4885	0.6090	0.2057	0.3147	0.2508	0.4107
		LN-Entropy [40]	0.6423	0.2242	0.3479	0.2939	0.4754	0.6248	0.2134	0.5862	0.4147	0.5264	0.5337	0.0494	0.3125	0.2340	0.3678
		PPL [30]	0.6526	0.2330	**0.3846**	0.2744	0.4444	0.6219	0.1182	0.6000	0.4215	0.5162	0.5337	0.1701	0.3058	0.2521	0.3630
	Length	Length-Score [57]	0.5184	0.0810	0.2817	0.2329	0.3400	0.5814	0.1487	0.5345	0.3848	0.4211	0.5971	0.1843	0.4711	0.3434	0.4284
	PRM	Qwen2.5-PRM800K [59]	0.5708	0.1285	0.3077	0.2697	0.4028	0.7267	0.4100	0.5862	0.3819	0.5132	0.6579	0.2451	0.4476	0.3366	0.4702
		Qwen2.5-PRM-7B [58]	0.5416	0.1249	0.3538	0.2918	0.4429	0.6983	0.3633	0.6133	0.4556	0.5449	0.6674	0.2758	0.5045	0.3642	0.4853
	LCM	GPT4-o [1]	0.6604	0.2458	0.2154	0.1785	0.3073	0.6265	0.1344	0.3333	0.1628	0.1933	0.6328	0.2356	0.2517	0.1878	0.2683
		Qwen2.5-32B [54]	0.6650	0.3055	0.2676	0.2451	0.3632	0.6974	0.2381	0.3833	0.2150	0.3428	0.7071	0.2716	0.3472	0.2517	0.4177
	Self-Aware	UQAC [25]	0.6374	0.2303	0.3444	0.2836	**0.5104**	0.7157	0.3732	0.6207	0.4170	0.5050	0.6952	0.3397	0.5417	0.4222	0.4988
		EigenScore [5]	0.6706	0.3496	0.3282	0.2282	0.3388	0.6146	0.2228	0.4469	0.3508	0.3337	0.6719	0.3056	0.3694	0.3542	0.3750
	Ours	RHD	**0.7292**	0.3476	0.3692	**0.3005**	0.4644	**0.7649**	**0.4506**	**0.6667**	**0.4714**	**0.5671**	**0.7255**	**0.3742**	**0.5785**	**0.4421**	**0.5154**

Table 2: Performance comparisons between GRPO-R and baselines. Bold indicates the best result.

Models	DeepSeek-R1-1.5B					Qwen2.5-1.5B-Instruct				
	MATH500	AIME(2024)	GPQA(diamond)	GPQA(main)	GPQA(extended)	MATH500	AIME(2024)	GPQA(diamond)	GPQA(main)	GPQA(extended)
Base	0.772	0.333	0.354	0.333	0.339	0.466	0.100	0.202	0.197	0.211
+GRPO	0.770	0.333	0.359	0.335	**0.359**	0.480	0.033	**0.247**	0.214	0.266
+GRPO-R	**0.788**	**0.367**	**0.414**	**0.371**	0.357	**0.490**	**0.133**	**0.247**	**0.243**	**0.275**

Ensemble and LCM methods perform well in binary detection but struggle in multi-trace ranking, indicating difficulty in fine-grained comparison. Uncertainty-based methods are sensitive to output length, while Process Reward Models often suffer from limited generalization. In contrast, RHD directly leverages reasoning mechanisms for more accurate detection. Self-awareness methods perform competitively but lack explicit reasoning analysis. Interestingly, the Length-based baseline performs well in multi-trace settings—supporting the intuition that overly long traces are more error-prone, but underperforms in binary detection, limiting its generality. These findings highlight the effectiveness of RHD modeling internal reasoning patterns for hallucination detection. Additional ablations and sensitivity studies are provided in Appendix J and K.

5.2 Reasoning Hallucination Mitigation

Experimental Setting. To assess the effectiveness of GRPO-R in reducing reasoning hallucinations, we fine-tune `Qwen2.5-1.5B-Instruct` and `DeepSeek-R1-1.5B` on 2,000 examples from `OpenR1-Math-220K` [43] using either GRPO or our proposed GRPO-R. We evaluate the accuracy [20] on two in-domain math benchmarks—`MATH500` [27] and `AIME 2024` [2]—and an out-of-distribution science benchmark—`GPQA` [39]. Implementation details are in Appendix L.

Main Results. As shown in Table 2, GRPO-R outperforms GRPO across most of the tasks, indicating that shaping reasoning steps via the reasoning score enhances both factual accuracy and reasoning reliability. Gains on `GPQA` further suggest improved generalization beyond training distribution. Additional sensitivity analyses are in Appendix M. Hallucination mitigation experiments in data distillation in Appendix N further validate the effectiveness of our proposed RHD model.

6 Conclusion and Limitation

We tackle the challenge of *Reasoning Hallucination* in LRMs, where models produce logically coherent but factually incorrect reasoning traces. To address this, we propose the **Reasoning Score**, a step-level metric derived from the mechanistic interpretability of reasoning, which effectively quantifies reasoning depth. Based on this score, we identify three key hallucination patterns: early-stage depth fluctuations, incorrect backtracking and spurious verification-induced overthinking—and develop the **RHD** framework for hallucination detection. Building on these insights, we introduce **GRPO-R**, a reinforcement learning method that integrates step-level reasoning rewards via potential-based shaping, improving both accuracy and robustness across reasoning benchmarks.

Limitation. RHD relies on internal model activations and is thus limited to open-source LRMs with accessible activations. Its application to black-box models remains an open challenge. Nonetheless, the discovered patterns and metrics could inspire proxy-based extensions. Additionally, experiments are conducted on moderate-scale models and datasets due to computational constraints; future work includes scaling up to broader domains and model families.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

[2] AI-MO. Ai-mo/aimo-validation-aime. `https://huggingface.co/datasets/AI-MO/aimo-validation-aime`, 2024. Apache 2.0 License.

[3] AI-MO. Ai-mo/aimo-validation-amc. `https://huggingface.co/datasets/AI-MO/aimo-validation-amc`, 2024. Apache 2.0 License.

[4] G. Bruckmaier, S. Krauss, K. Binder, S. Hilbert, and M. Brunner. Tversky and kahneman's cognitive illusions: who can solve them, and why? Frontiers in psychology, 12:584689, 2021.

[5] C. Chen, K. Liu, Z. Chen, Y. Gu, Y. Wu, M. Tao, Z. Fu, and J. Ye. Inside: Llms' internal states retain the power of hallucination detection. In The Twelfth International Conference on Learning Representations, 2024.

[6] S. Chen, J. Zhang, T. Zhu, W. Liu, S. Gao, M. Xiong, M. Li, and J. He. Bring reason to vision: Understanding perception and reasoning through model merging. arXiv preprint arXiv:2505.05464, 2025.

[7] Y. Chen, J. Benton, A. Radhakrishnan, J. Uesato, C. Denison, J. Schulman, A. Somani, P. Hase, M. Wagner, F. Roger, et al. Reasoning models don't always say what they think. arXiv preprint arXiv:2505.05410, 2025.

[8] R. Cohen, M. Hamri, M. Geva, and A. Globerson. Lm vs lm: Detecting factual errors via cross examination. arXiv preprint arXiv:2305.13281, 2023.

[9] N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, N. DasSarma, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. A mathematical framework for transformer circuits. Transformer Circuits Thread, 2021. URL `https://transformer-circuits.pub/2021/framework/index.html`.

[10] D.-A. et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025. URL `https://arxiv.org/abs/2501.12948`.

[11] B. Everitt. The cambridge dictionary of statistics. In The Cambridge dictionary of statistics, pages 360–360. 1998.

[12] J. Ferrando and E. Voita. Information flow routes: Automatically interpreting language models at scale. arXiv preprint arXiv:2403.00824, 2024.

[13] J. Ferrando, G. Sarti, A. Bisazza, and M. R. Costa-jussà. A primer on the inner workings of transformer-based language models. arXiv preprint arXiv:2405.00208, 2024.

[14] R. Friel and A. Sanyal. Chainpoll: A high efficacy method for llm hallucination detection. arXiv preprint arXiv:2310.18344, 2023.

[15] M. Geva, R. Schuster, J. Berant, and O. Levy. Transformer feed-forward layers are key-value memories. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5484–5495, 2021.

[16] M. Hanna, O. Liu, and A. Variengien. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. Advances in Neural Information Processing Systems, 36, 2024.

[17] S. Hao, S. Sukhbaatar, D. Su, X. Li, Z. Hu, J. Weston, and Y. Tian. Training large language models to reason in a continuous latent space. arXiv preprint arXiv:2412.06769, 2024.

[18] Y. He, S. Li, J. Liu, W. Wang, X. Bu, G. Zhang, Z. Peng, Z. Zhang, Z. Zheng, W. Su, and B. Zheng. Can large language models detect errors in long chain-of-thought reasoning?, 2025. URL `https://arxiv.org/abs/2502.19361`.

[19] X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. arXiv preprint arXiv:2011.01060, 2020.

[20] Hugging Face. Math-verify: A rule-based mathematical answer verification library, 2025. URL `https://github.com/huggingface/Math-Verify`.

[21] S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, Z. Hatfield-Dodds, N. DasSarma, E. Tran-Johnson, et al. Language models (mostly) know what they know. arXiv preprint arXiv:2207.05221, 2022.

[22] D. Kahneman. Thinking, fast and slow. macmillan, 2011.

[23] D. Li, S. Cao, T. Griggs, S. Liu, X. Mo, E. Tang, S. Hegde, K. Hakhamaneshi, S. G. Patil, M. Zaharia, et al. Llms can easily learn to reason from demonstrations structure, not content, is what matters! arXiv preprint arXiv:2502.07374, 2025.

[24] K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. Advances in Neural Information Processing Systems, 36, 2024.

[25] Y. Li, R. Qiang, L. Moukheiber, and C. Zhang. Language model uncertainty quantification with attention chain, 2025. URL `https://arxiv.org/abs/2503.19168`.

[26] Z. Li, G. Jiang, H. Xie, L. Song, D. Lian, and Y. Wei. Understanding and patching compositional reasoning in llms. arXiv preprint arXiv:2402.14328, 2024.

[27] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let's verify step by step. In The Twelfth International Conference on Learning Representations, 2023.

[28] S. Lin, J. Hilton, and O. Evans. Truthfulqa: Measuring how models mimic human falsehoods. arXiv preprint arXiv:2109.07958, 2021.

[29] L. Luo, Y. Liu, R. Liu, S. Phatale, M. Guo, H. Lara, Y. Li, L. Shu, Y. Zhu, L. Meng, et al. Improve mathematical reasoning in language models by automated process supervision. arXiv preprint arXiv:2406.06592, 2024.

[30] A. Malinin and M. Gales. Uncertainty estimation in autoregressive structured prediction. arXiv preprint arXiv:2002.07650, 2020.

[31] P. Manakul, A. Liusie, and M. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 9004–9017, 2023.

[32] I. Mirzadeh, K. Alizadeh, H. Shahrokhi, O. Tuzel, S. Bengio, and M. Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. arXiv preprint arXiv:2410.05229, 2024.

[33] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In Icml, volume 99, pages 278–287. Citeseer, 1999.

[34] Y. Nikankin, A. Reusch, A. Mueller, and Y. Belinkov. Arithmetic without algorithms: Language models solve math with a bag of heuristics. In The Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=O9YTt26r2P.

[35] nostalgebraist. Interpreting GPT: the logit lens. AI Alignment Forum, 2020. URL https://www.alignmentforum.org/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

[36] OpenAI. Openai o3 and o4-mini system card, April 2025. URL https://openai.com/index/o3-o4-mini-system-card/.

[37] A. Prasad, S. Saha, X. Zhou, and M. Bansal. Receval: Evaluating reasoning chains via correctness and informativeness. arXiv preprint arXiv:2304.10703, 2023.

[38] O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis. Measuring and narrowing the compositionality gap in language models. arXiv preprint arXiv:2210.03350, 2022.

[39] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In First Conference on Language Modeling, 2024.

[40] J. Ren, J. Luo, Y. Zhao, K. Krishna, M. Saleh, B. Lakshminarayanan, and P. J. Liu. Out-of-distribution detection and selective generation for conditional language models. In The Eleventh International Conference on Learning Representations, 2022.

[41] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

[42] A. Stolfo, Y. Belinkov, and M. Sachan. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In The 2023 Conference on Empirical Methods in Natural Language Processing, 2023. URL https://openreview.net/forum?id=aB3Hwh4UzP.

[43] O.-R. Team. Openr1-math-220k: A dataset for mathematical reasoning, 2024. URL https://huggingface.co/datasets/open-r1/OpenR1-Math-220k.

[44] K. Tentori, N. Bonini, and D. Osherson. The conjunction fallacy: A misunderstanding about conjunction? Cognitive Science, 28(3):467–477, 2004.

[45] Transluce Research. Investigating truthfulness in a pre-release o3 model, 2024. URL https://transluce.org/investigating-o3-truthfulness.

[46] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. Musique: Multihop questions via single-hop question composition. Transactions of the Association for Computational Linguistics, 10:539–554, 2022.

[47] K. Valmeekam, K. Stechly, and S. Kambhampati. Llms still can't plan; can lrms? a preliminary evaluation of openai's o1 on planbench. In NeurIPS 2024 Workshop on Open-World Agents.

[48] Vectara Research. Deepseek-r1 hallucinates more than deepseek-v3, 2025. URL https://www.vectara.com/blog/deepseek-r1-hallucinates-more-than-deepseek-v3.

[49] X. Wang, S. Tan, M. Jin, W. Y. Wang, R. Panda, and Y. Shen. Do larger language models imply better reasoning? a pretraining scaling law for reasoning. arXiv preprint arXiv:2504.03635, 2025.

[50] P. P. Weis and W. Kunde. Switching between different cognitive strategies induces switch costs as evidenced by switches between manual and mental object rotation. Scientific Reports, 14(1): 6217, 2024.

[51] W. Wu, Y. Wang, G. Xiao, H. Peng, and Y. Fu. Retrieval head mechanistically explains long-context factuality. arXiv preprint arXiv:2404.15574, 2024.

[52] X. Xu, S. Diao, C. Yang, and Y. Wang. Can we verify step by step for incorrect answer detection? arXiv preprint arXiv:2402.10528, 2024.

[53] K. Yan, Y. Xu, Z. Du, X. Yao, Z. Wang, X. Guo, and J. Chen. Recitation over reasoning: How cutting-edge language models can fail on elementary school-level reasoning problems? arXiv preprint arXiv:2504.00509, 2025.

[54] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115, 2024.

[55] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. arXiv preprint arXiv:1809.09600, 2018.

[56] Q. Yu, J. Merullo, and E. Pavlick. Characterizing mechanisms for factual recall in language models. In The 2023 Conference on Empirical Methods in Natural Language Processing, 2023.

[57] Z. Zeng, Q. Cheng, Z. Yin, Y. Zhou, and X. Qiu. Revisiting the test-time scaling of o1-like models: Do they truly possess test-time scaling capabilities? arXiv preprint arXiv:2502.12215, 2025.

[58] Z. Zhang, C. Zheng, Y. Wu, B. Zhang, R. Lin, B. Yu, D. Liu, J. Zhou, and J. Lin. The lessons of developing process reward models in mathematical reasoning. arXiv preprint arXiv:2501.07301, 2025.

[59] C. Zheng, Z. Zhang, B. Zhang, R. Lin, K. Lu, B. Yu, D. Liu, J. Zhou, and J. Lin. Processbench: Identifying process errors in mathematical reasoning. arXiv preprint arXiv:2412.06559, 2024.

[60] C. Zheng, Z. Zhang, B. Zhang, R. Lin, K. Lu, B. Yu, D. Liu, J. Zhou, and J. Lin. Processbench: Identifying process errors in mathematical reasoning, 2024. URL `https://arxiv.org/abs/2412.06559`.

[61] H. Zhou, Z. Feng, Z. Zhu, J. Qian, and K. Mao. Unibias: Unveiling and mitigating llm bias through internal attention and ffn manipulation. arXiv preprint arXiv:2405.20612, 2024.

A Proof of Generalization Gap with Augmented Rewards

Proof of Theorem 1. For any policy $\pi \in \Pi$, define the augmented return

$$R(\pi, \xi) = \sum_{t=1}^{T} \gamma^{t-1} \bar{r}_t(\xi).$$

Assume that $\bar{r}_t(\xi) \in [0, \bar{r}_{\max}]$ for each t, so that

$$R(\pi, \xi) \in [0, \bar{R}_{\max}].$$

Define the expected return:

$$J_{\text{test}}(\pi) = \mathbb{E}_{\xi \sim \mathcal{D}} \left[R(\pi, \xi) \right],$$

and the empirical return:

$$J_{\text{train}}(\pi) = \frac{1}{n} \sum_{i=1}^{n} R(\pi, \xi_i).$$

We aim to bound the expected generalization gap between the test return and empirical return for policies in class Π via Rademacher complexity. Let the function class be defined as

$$\mathcal{F} = \{f_\pi(\xi) = R(\pi, \xi) \mid \pi \in \Pi\},$$

where $R(\pi, \xi)$ is the total return over trajectory ξ under policy π using the augmented reward $\bar{r}_t$. Our goal is to bound:

$$\sup_{\pi \in \Pi} |J_{\text{test}}(\pi) - J_{\text{train}}(\pi)| = \sup_{f \in \mathcal{F}} \left| \mathbb{E}[f(\xi)] - \frac{1}{n} \sum_{i=1}^{n} f(\xi_i) \right|.$$

Let $\xi_1, \ldots, \xi_n$ be the training samples drawn i.i.d. from the environment distribution $\mathcal{D}$, and $\xi'_1, \ldots, \xi'_n$ be another independent copy drawn from the same distribution. By using an independent ghost sample set and the triangle inequality, we have:

$$\begin{aligned}
\mathbb{E}_{\{\xi_i\}} \left[\sup_{f \in \mathcal{F}} \left(\mathbb{E}_{\xi \sim \mathcal{D}}[f(\xi)] - \frac{1}{n} \sum_{i=1}^{n} f(\xi_i) \right) \right] &= \mathbb{E}_{\{\xi_i\},\{\xi'_i\}} \left[\sup_{f \in \mathcal{F}} \left(\frac{1}{n} \sum_{i=1}^{n} f(\xi'_i) - f(\xi_i) \right) \right] \\
&\leq \mathbb{E}_{\{\xi_i\},\{\xi'_i\}} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \left(f(\xi'_i) - f(\xi_i) \right) \right].
\end{aligned}$$

To simplify the expression, we now introduce independent Rademacher variables $\sigma_1, \ldots, \sigma_n \in \{-1, +1\}$, where each σ_i takes value $+1$ or -1 with equal probability. Since $f(\xi'_i) - f(\xi_i)$ is symmetric around zero due to $\xi_i \sim \xi'_i$, we can write:

$$\mathbb{E}_{\{\xi_i\},\{\xi'_i\}} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \left(f(\xi'_i) - f(\xi_i) \right) \right] = \mathbb{E}_{\{\xi_i\},\{\xi'_i\},\{\sigma_i\}} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \left(f(\xi'_i) - f(\xi_i) \right) \right].$$

We now apply the triangle inequality again:

$$\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \sigma_i \left(f(\xi'_i) - f(\xi_i) \right) \leq \sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \sigma_i f(\xi'_i) + \sup_{f \in \mathcal{F}} \sum_{i=1}^{n} (-\sigma_i) f(\xi_i).$$

Since $-\sigma_i$ is still a Rademacher variable and ξ_i and ξ'_i have the same distribution, the two expectations are equal. Thus, we obtain:

$$\begin{aligned}
\mathbb{E}_{\{\xi_i\},\{\xi'_i\}} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \left(f(\xi'_i) - f(\xi_i) \right) \right] &\leq 2\mathbb{E}_{\{\xi_i\},\{\sigma_i\}} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i f(\xi_i) \right] \\
&= 2\mathcal{R}_n(\mathcal{F}),
\end{aligned}$$

where $\mathcal{R}_n(\mathcal{F})$ is the empirical Rademacher complexity of $\mathcal{F}$.

Assume every return is bounded, $0 \leq f_\pi(\xi) \leq \bar{R}_{\max}$, and that $f_\pi(\xi)$ is linear in the augmented per–step rewards $\bar{r}_t(\xi)$:

$$f_\pi(\xi) \;=\; \sum_{t=1}^{T} \gamma^{t-1} \bar{r}_t(\xi).$$

Introduce the normalised return $\tilde{f}_\pi(\xi) := f_\pi(\xi) / \bar{R}_{\max} \in [0, 1]$ and let $\tilde{\mathcal{F}} := \{\tilde{f}_\pi \mid \pi \in \Pi\}$. Because Rademacher complexity is positively homogeneous in its function class,

$$\mathcal{R}_n(\mathcal{F}) \;=\; \mathcal{R}_n\big(\bar{R}_{\max} \tilde{\mathcal{F}}\big) \;=\; \bar{R}_{\max} \mathcal{R}_n(\tilde{\mathcal{F}}).$$

We measure the complexity of the policy class precisely through these normalised returns and set

$$\mathcal{R}_n(\Pi) := \mathcal{R}_n(\tilde{\mathcal{F}}).$$

Justification. Even if the mapping $\pi \mapsto \tilde{f}_\pi$ is not injective, Rademacher complexity is **monotone** with respect to set inclusion: enlarging the function class can only increase $\mathcal{R}_n$. Hence analysing the

(possibly larger) class $\tilde{\mathcal{F}}$ yields a conservative upper bound on the true policy complexity—exactly what we need for a valid generalisation bound.

Combining the two displays yields

$$\mathcal{R}_n(\mathcal{F}) \;\le\; \bar{R}_{\max}\,\mathcal{R}_n(\Pi)$$

(the identity can be written as "$\le$" because any alternative normalisation would only shrink the right–hand side).

Substituting the above bound into the symmetrisation result, we obtain

$$\mathbb{E}\Big[\sup_{\pi\in\Pi}\big|J_{\text{test}}(\pi) - J_{\text{train}}(\pi)\big|\Big] \;\le\; 2\,\bar{R}_{\max}\,\mathcal{R}_n(\Pi),$$

We now move from the expected generalization gap to a high-probability bound that holds uniformly over all policies $\pi \in \Pi$.

Let $X_i = R(\pi, \xi_i) = \sum_{t=1}^{T} \gamma^{t-1}\bar{r}_t(\xi_i)$ be the augmented return of policy π on the i-th training trajectory. Then $J_{\text{train}}(\pi) = \frac{1}{n}\sum_{i=1}^{n} X_i$ and $J_{\text{test}}(\pi) = \mathbb{E}_{\xi\sim\mathcal{D}}[X_i]$. By assumption, $X_i \in [0, \bar{R}_{\max}]$.

Applying Hoeffding's inequality for bounded i.i.d. variables, we have for any fixed $\pi \in \Pi$:

$$\Pr\left(|J_{\text{test}}(\pi) - J_{\text{train}}(\pi)| \ge \varepsilon\right) \le 2\exp\left(-\frac{2n\varepsilon^2}{(\bar{R}_{\max})^2}\right).$$

Solving for ε yields that with probability at least $1-\delta$,

$$|J_{\text{test}}(\pi) - J_{\text{train}}(\pi)| \le \bar{R}_{\max}\sqrt{\frac{\log(1/\delta)}{2n}}. \tag{16}$$

Define the worst-case generalization gap over the policy class:

$$\Delta(\mathcal{S}) := \sup_{\pi\in\Pi}\left(J_{\text{test}}(\pi) - J_{\text{train}}(\pi)\right),$$

where $\mathcal{S} = \{\xi_1, \ldots, \xi_n\}$ is the training set.

(i) Expected bound from above: Using symmetrization and Rademacher complexity arguments, we already established:

$$\mathbb{E}_{\mathcal{S}}[\Delta(\mathcal{S})] \le 2\bar{R}_{\max}\mathcal{R}_n(\Pi). \tag{6}$$

(ii) High-probability deviation bound via McDiarmid's inequality: Let us show that $\Delta(\mathcal{S})$ concentrates around its expectation. Consider replacing any single sample ξ_i in $\mathcal{S}$ by an independent copy ξ_i'. Because each return $X_i = R(\pi, \xi_i)$ is bounded in $[0, \bar{R}_{\max}]$ and each contributes $\frac{1}{n}$ to the empirical mean, the influence of changing ξ_i is bounded by:

$$\left|\Delta(\mathcal{S}) - \Delta(\mathcal{S}^{(i)})\right| \le \frac{\bar{R}_{\max}}{n}.$$

Hence, $\Delta(\mathcal{S})$ is $\bar{R}_{\max}/n$-Lipschitz in each of its n arguments.

Applying McDiarmid's inequality:

$$\Pr\left(\Delta(\mathcal{S}) - \mathbb{E}[\Delta(\mathcal{S})] \ge \varepsilon\right) \le \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{n}(\bar{R}_{\max}/n)^2}\right) = \exp\left(-\frac{2n\varepsilon^2}{(\bar{R}_{\max})^2}\right).$$

Solving for ε again yields that with probability at least $1-\delta$,

$$\Delta(\mathcal{S}) \le \mathbb{E}[\Delta(\mathcal{S})] + \bar{R}_{\max}\sqrt{\frac{\log(1/\delta)}{2n}}. \tag{7}$$

(iii) Final generalization gap: Combining Equation 6 and 7, with probability at least $1-\delta$ over the random draw of the training set $\mathcal{S}$, we obtain:

$$\sup_{\pi\in\Pi}\left[J_{\text{test}}(\pi) - J_{\text{train}}(\pi)\right] \le 2\bar{R}_{\max}\mathcal{R}_n(\Pi) + \bar{R}_{\max}\sqrt{\frac{\log(1/\delta)}{2n}}.$$

Equivalently, for all $\pi \in \Pi$,

$$J_{\text{test}}(\pi) - J_{\text{train}}(\pi) \leq 2\bar{R}_{\max}\,\mathcal{R}_n(\Pi) + \bar{R}_{\max}\sqrt{\frac{\log(1/\delta)}{2n}} \tag{8}$$

□

Conclusion. Equation 8 provides a uniform generalization gap for any policy $\pi \in \Pi$, showing that the expected test-time performance is lower bounded by the training performance minus a complexity-dependent regularization term. According to this theorem, as the augmented reward $R_{\text{score}}(s_t)$ is well-aligned with genuine logical reasoning, it acts as a regularizer that effectively reduces the Rademacher complexity $\mathcal{R}_n(\Pi)$, thereby tightening the bound. This theoretical result highlights that our proposed process supervision framework not only improves credit assignment during training but also enhances generalization to unseen reasoning tasks.

B Detailed Implementation of GRPO-R

Our proposed process-level reasoning score supervision is compatible with any token-level RL algorithm. In this work, we instantiate it within Group Relative Policy Optimization (GRPO), yielding **GRPO-R**. GRPO is a scalable and widely used RL framework for reasoning model training, which promotes the generation of high-quality reasoning trajectories by ranking G candidate outputs based on their relative returns, without relying on explicit value estimation [10, 41].

Given a prompt q and G outputs $\{o_i\}_{i=1}^{G}$, each output o_i corresponds to a sequence of reasoning states $\{s_{i,1}, \ldots, s_{i,K}\}$ produced over K reasoning steps. In the original GRPO setup, only the final step receives a nonzero reward:

$$r_i^{\text{step}}(j) = \begin{cases} r_i^{\text{final}}, & j = K, \\ 0, & j < K, \end{cases}$$

where r_i^{final} denotes the scalar reward assigned to the final outcome.

We replace this sparse signal with our shaped step-level reward using potential-based reward shaping:

$$\bar{r}_i^{\text{step}}(j) = \tilde{r}_i^{\text{step}}(j) \;-\; \gamma\,\tilde{R}_{\text{score}}(s_{i,j+1}) \;+\; \tilde{R}_{\text{score}}(s_{i,j}),$$

where $\tilde{R}_{\text{score}}(s) = \min\big(R_{\text{score}}(s), \tau\big)$ and we set $\gamma = 1$. These shaped rewards are collected into the set $\mathbf{R}'$, standardized as:

$$\hat{r}_i^{\text{step}}(j) = \frac{\bar{r}_i^{\text{step}}(j) - \text{mean}(\mathbf{R}')}{\text{std}(\mathbf{R}')},$$

and used to compute token-level advantages:

$$\hat{A}_{i,t} = \sum_{j:\,\text{step}(j) \geq t} \hat{r}_i^{\text{step}}(j).$$

Finally, we optimize the policy using the enhanced GRPO objective, termed GRPO-R:

$$\begin{aligned}\mathcal{J}_{\text{GRPO-R}}(\theta) = \mathbb{E}_{q\sim P(Q),\,\{o_i\}\sim\pi_{\theta_{\text{old}}}(O|q)}\Bigg[\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\min\Big(&\frac{\pi_\theta(o_{i,t}\mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid q, o_{i,<t})}\,\hat{A}_{i,t}, \\ &\text{clip}\Big(\frac{\pi_\theta(o_{i,t}\mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid q, o_{i,<t})}, 1-\epsilon, 1+\epsilon\Big)\,\hat{A}_{i,t}\Big) \\ &- \beta\cdot \mathrm{D}_{\text{KL}}\left[\pi_\theta \| \pi_{\text{ref}}\right]\Bigg].\end{aligned} \tag{9}$$

Table 3: Statistics of ReTruthQA dataset across domains.

Dataset	#Samples	#Traces	Avg Truthful Traces	Avg Hallucination Traces
MATH	57	417	3.35	3.96
Science	88	541	3.05	3.10
MultiHopQA	184	1186	2.74	3.70

C ReTruthQA Construction

C.1 Data Sources and Models

Due to the lack of dedicated datasets for evaluating reasoning hallucination detection methods, especially for strong open-source models such as `DeepSeek-R1-7B` and `R1-14B`, we construct a new benchmark tailored for hallucination detection in multi-step reasoning tasks. We select three major categories of reasoning tasks: `Math`, `Science`, and `MultiHopQA`.

For `Math`, we construct the dataset using benchmark datasets commonly used for evaluating mathematical reasoning capabilities, including `MATH500` [27], `AMC 2023` [3], and `AIME 2024` [2].

For `Science`, we adopt `GPQA` [39], a PhD-level science multiple-choice QA dataset with questions authored by domain experts in physics, chemistry, and biology.

For `MultiHopQA`, we randomly sample 1000 questions from four multi-hop QA datasets: `HotpotQA` [55], `2WikiMultihopQA` [19], `MuSiQue` [46], and `Bamboogle` [38].

For each question, we generate 20 responses using `DeepSeek-R1-Distill-Qwen-7B` and `DeepSeek-R1-Distill-Qwen-14B` via random sampling. The prompting format is as follows:

Math:

```
Please answer the following math question.
You should provide your final answer in the format \boxed{YOUR_ANSWER}.
Separate your following steps using \n\n.
Question:\n\n
```

Science:

```
Please answer the following multiple-choice question.
You should provide your final choice in the format \boxed{YOUR_CHOICE}.
Separate your following steps using \n\n.
Question:\n\n
```

MultiHopQA:

```
Please answer the following question.
You should provide your final answer in the format \boxed{YOUR_ANSWER}.
Separate your following steps using \n\n.
Question:\n\n
```

C.2 Reasoning Step Segmentation Strategy

We adopt a two-stage segmentation procedure. First, we split the reasoning trace based on cognitive behavior tokens such as `</think>`, `Wait`, `But`, `However`, `Hmm`, `Alternatively`, which typically mark transitions in reasoning patterns. Then, we apply a finer-grained split based on formatting: as specified in the prompt, the LRM is instructed to separate reasoning steps using `\n\n`, which we use as a delimiter. This hybrid approach ensures both rule-based and model-aligned step boundaries.

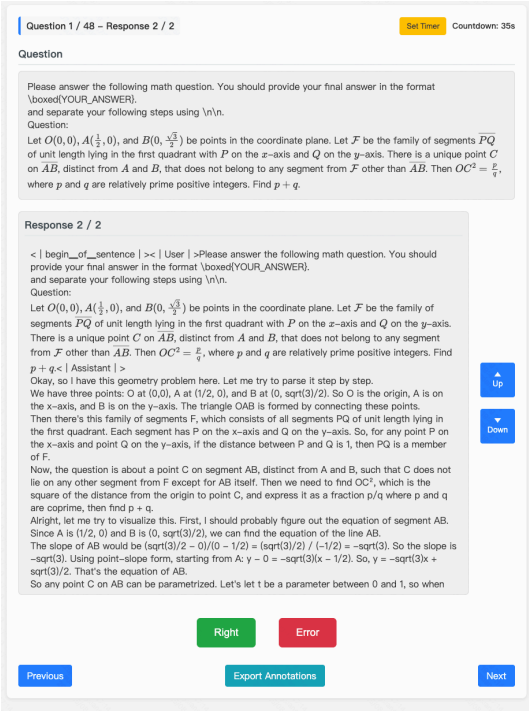


Figure 5: Interface Display of the Data Annotation Platform.

C.3 Annotation Process

1. Automatic hallucination trace identification. To ensure precision and avoid noise caused by random model errors, a reasoning trace is labeled as hallucinated only if its rollout becomes incorrect with a failure rate exceeding 90% from a specific reasoning step onward, measured over 16 rollouts. We adopt a binary search–style trace slicing procedure inspired by OmegaProcess [29] to efficiently identify hallucination points. This strategy ensures stability and causality in hallucination step detection, avoiding incidental errors due to sampling randomness. For the Science domain, which mainly consists of multiple-choice questions and may contain correct guesses, we additionally perform multiple random rollouts for traces with correct answers to ensure a success rate above 90% before labeling them as truthful.

2. Filtering non-hallucination failures. We use GPT-4o-Mini to exclude samples where the incorrect final answer is due to clearly flawed or illogical reasoning, which does not satisfy our definition of hallucination (i.e., coherent and persuasive chains with underlying logical or factual errors). The filtering prompt is:

> Please evaluate if the following reasoning for the given question is logically sound and leads to a correct solution.
> Only respond with a score between 0 and 1, where:
> 0: completely incorrect or illogical reasoning
> 1: perfectly sound and correct reasoning
>
> `Question: {question}`
> `Reasoning: {reasoning}`
> Score (0–1):

3. Human validation. We further perform human annotation to verify borderline cases. Two annotators with at least undergraduate-level backgrounds in computer science independently assess whether the reasoning trace is valid. We developed a web-based annotation platform with a timer (Figure 5) to standardize reading time. Based on average reading speeds (200–300 wpm for academic

text), and trace lengths (typically 2000–3000 words), we set the following maximum judgment times: (1) MultiHopQA: 3 minutes (2) Math: 5 minutes (3) Science: 8 minutes

Annotators must determine within the allotted time whether a reasoning trace contains hallucinations. If they fail to identify an error in time, the trace is labeled as correct. Cases judged correct by humans but verified to be incorrect are labeled as hallucinations, ensuring that the resulting dataset captures only traces that genuinely mislead users, which is aligned with the definition of reasoning hallucination.

Final dataset statistics are shown in Table 3. For the **Multi-Trace Ranking Setting**, we directly use the collected hallucinated and truthful responses. For the **Binary Detection Setting**, which focuses on single-response accuracy, we retain one hallucinated and one truthful response per question to reflect more realistic ad-hoc usage scenarios.

D GSM-NoOp Construction Process

Following the construction procedure proposed in [32], we randomly sample 300 examples from the GSM8K dataset. For each question, we use `GPT-4o` to generate a No-Op phrase using the following prompt:

> Given the following math question, generate a seemingly relevant but ultimately inconsequential statement (No-Op) that can be added to the question without affecting its solution.
> Question: {Question}
> Generate a No-Op statement that:
> 1. Is short and concise
> 2. Seems relevant to the context
> 3. Does not affect the mathematical reasoning
> 4. Is natural and fits grammatically
> No-Op statement:

We then use `GPT-4o` to combine the generated No-Op phrase with the original question using the following prompt:

> Please combine the following math question and No-Op phrase into a single, natural-sounding question. The No-Op phrase should be integrated smoothly without changing the mathematical meaning.
> Math Question: {Question} No-Op Phrase: {NoOp Phrase}
> Combined Question:

The merged questions form our constructed **GSM-NoOp** dataset.

To evaluate whether the generated reasoning steps are misled by the inserted No-Op phrase, we prompt `GPT-4o` with the following instruction:

> Please evaluate if the following reasoning step is being misled by the given No-Op phrase. Provide a score between 0 and 1, where:
> a. 0 means the step is not misled by the No-Op phrase at all
> b. 1 means the step is completely misled by the No-Op phrase
> c. Values in between indicate partial misleading
>
> Note: Simply mentioning the No-Op phrase does not count as being misled. If the step mentions the No-Op phrase but explicitly rejects or explains why it is irrelevant to solving the problem, this should be scored as 0.
> Reasoning step: {Reasoning Step} No-Op phrase: {NoOp Phrase}
> Please provide only a number between 0 and 1, with up to 2 decimal places, wrapped in \boxed{}. For example: \boxed{0.85}

E Details of Understanding the Mechanisms Behind Reasoning Hallucination Patterns

In this section, we focus on analyzing the underlying cause of **Pattern #1**, as **Pattern #2** has already been explained through the attention behavior of LRMs in the previous section. Pattern #1 highlights that hallucinated reasoning traces tend to exhibit larger fluctuations in reasoning depth, particularly in the early steps. Inspired by our preliminary analysis in § 3.2.1, we hypothesize that this may stem from the model's built-in verification capability. However, several key questions remain: **Q1**: What triggers verification behavior in LRMs? **Q2**: Do excessively high reasoning scores genuinely indicate overthinking? **Q3**: If Q2 holds, what factors lead to the emergence of such overthinking steps?

To answer these questions, we construct reasoning step triples (c_1, c_2, c_3) with different properties drawn from reasoning traces: **Stable**: The first type consists of triples from truthful traces where adjacent steps differ in R_{score} by less than 0.1, representing stable reasoning. **Rising-1**: The second type contains hallucinated triples where $R_{\text{score}}(c_3) - R_{\text{score}}(c_2) > 1$ and $R_{\text{score}}(c_3) < 4$, used to analyze verification triggered by shallow pattern-matching. **Rising-2**: The third type is similar to Rising-1 but with $R_{\text{score}}(c_3) > 4$, aimed at understanding overthinking induced by verification.

Analysis. To investigate **Q1**, we analyze whether reasoning steps c_1 and c_2 in the stable and rising (Rising-1 + Rising-2) triples are logically consistent, using GPT-4o as the judge (prompt details in Appendix F). As shown in Figure 4(a), the stable triples exhibit significantly higher consistency between c_1 and c_2 than rising triples, indicating that LRMs are more likely to trigger verification when early steps are internally inconsistent.

To examine **Q2**, we evaluate the correctness of c_2 and c_3 in Rising-2 triples. Using ground-truth answers and GPT-4o-based annotation (prompt details in Appendix F), we assess whether these steps are logically aligned with the ground-truth answers. As shown in Figure 4(b), c_2 in Rising-2 triples is substantially more accurate than c_3, confirming that verification in this case often modifies correct reasoning into incorrect steps. These findings support the hypothesis that excessively high R_{score} values in hallucinated reasoning traces are symptomatic of overthinking—steps that exhibit apparent reasoning depth but in fact reflect spurious or detrimental reasoning.

To address **Q3**, we analyze the relationship between perplexity and R_{score}. Specifically, we randomly sample 200 reasoning steps from ReTruthQA and compute their perplexities as follows:

$$\text{PPL}(c_k) = \exp\left(-\frac{1}{|c_k|} \sum_{t_{m+1}^k \in c_k} \log p\left(t_{m+1}^k \mid t_{\leq m}^k\right)\right), \tag{10}$$

$$\text{PPL}(\mathcal{C}) = \langle \text{PPL}(c_1), \text{PPL}(c_2), \ldots, \text{PPL}(c_K) \rangle . \tag{11}$$

where $p(t_{m+1}^k \mid t_{\leq m}^k)$ denotes the model's predicted probability for token t_{m+1}^k given the prefix $t_{\leq m}^k$ within the reasoning trace.

As shown in Figure 4(c), perplexity and R_{score} are strongly negatively correlated—steps with higher reasoning depth tend to have lower perplexity, which is intuitive since deep reasoning often yields more predictable outputs. However, when comparing the final step c_3 across stable and Rising-2 triples, we find an interesting phenomenon in Figure 4(d): despite having higher R_{score}, c_3 in Rising-2 triples has higher perplexity than in stable triples. This suggests that overthinking steps induced by an incorrect verification result in an uncertain or internally unstable generation.

We hypothesize that such overthinking may reflect *spurious verification*—a behavior where the model performs superficial or misguided validation in pursuit of higher reward during RL fine-tuning. This behavior can persist through distillation into smaller models, propagating reasoning hallucinations. Based on this analysis, we identify a third hallucination pattern: **Pattern #3:** Overthinking reasoning steps exhibit a positive correlation between R_{score} and perplexity (PPL).

F Prompt for Hallucination Patterns Analysis

Prompt for step consistency analysis of **Q1**:

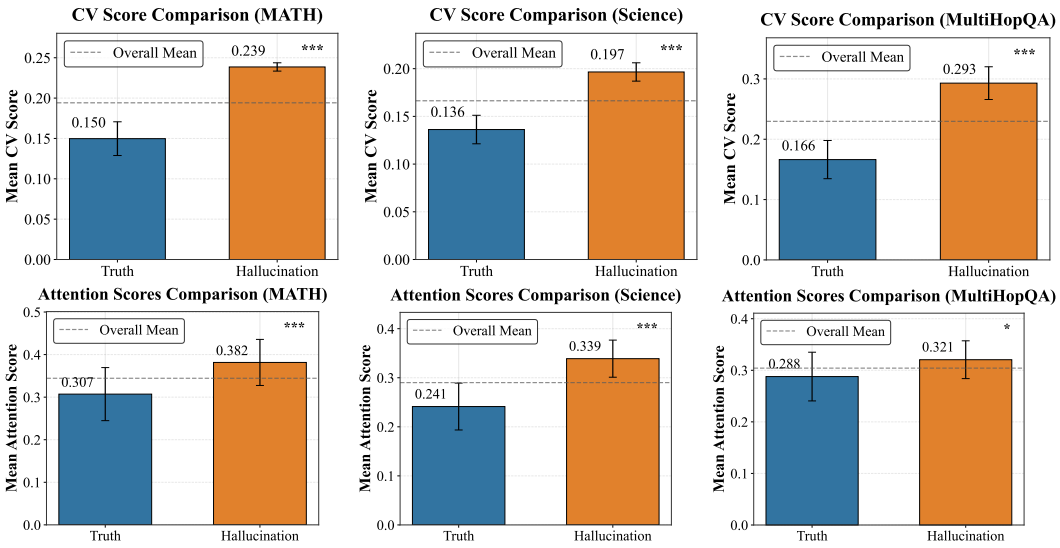


Figure 6: Evaluation of Pattern #1 and Pattern #2 on ReTruthQA. Asterisks indicate statistical significance based on a t-test: * for p-value < 0.05, and *** for p-value < 0.001.

> Please evaluate whether the following reasoning step introduces a new solution approach compared to the preceding steps. Respond with a score of 0 or 1, where:
> 0: The step follows the same solution approach as the previous steps.
> 1: The step explores a new solution approach or direction.
> Reasoning step: {step content}
> Previous steps: {step content}
> Score (0/1):

Prompt for step correctness analysis of **Q2**:

> Please evaluate whether the following reasoning step aligns with the final answer. Respond with a score of 0 or 1, where:
> 0: The step is inconsistent with the final answer.
> 1: The step is consistent with the final answer.
> Reasoning step: {step}
> Final answer: {answer}
> Score (0/1):

G More Results of Reasoning Hallucination Pattern Analysis

The hyperparameter settings involved in Section 3.2 are as follows. The constant r, which controls the size of the early-step window, is empirically set to $r = 2$. The constant η, which defines the portion of late reasoning steps, is set to $\eta = 0.75$. The constant K, used in computing attention to earlier steps, is set to $K = 5$. The threshold τ for identifying potentially overthinking steps is set to $\tau = 4$. These hyperparameters are derived from case analysis and are applied consistently throughout the subsequent reasoning hallucination detection and mitigation experiments.

The validity of Pattern #1 and Pattern #2 is verified across all domains of ReTruthQA, with experimental results shown in Figure 6, where across all three domains, hallucinated reasoning traces consistently exhibit significantly higher CV scores and Attention scores than truthful traces.

H Evaluation and Baseline Details of Reasoning Hallucination Detection

Based on ReTruthQA, we design two evaluation settings for RHD model: **(1) Binary Detection Setting:** This setting assesses the model's ability to detect hallucinations in individual question-reasoning pairs (Q, C), measuring detection performance using the Area Under the ROC Curve (***AUC***) and Pearson Correlation Coefficient (***PCC***); **(2) Multi-Trace Ranking Setting:** This setting evaluates the model's ability to identify the truthful answer among multiple reasoning traces for the same question $(Q, \{C_1, C_2, \ldots, C_3\})$. We follow the evaluation setup of TruthfulQA-MC [28], and report the following metrics: ***MC1***: The percentage of instances where the hallucination score of the most hallucinated reasoning trace exceeds that of all truthful traces; ***MC2***: The normalized total hallucination score assigned to the hallucinated reasoning traces; ***MC3***: The percentage of hallucinated reasoning traces that receive a higher hallucination score than all truthful traces. These metrics collectively measure the ranking quality of hallucination detection in multi-sample generation settings.

For baselines, we consider the following categories: **(1) Ensemble-based self-evaluation methods**, where hallucination scores are obtained through repeated generation, self-verification, or peer voting among LLMs. This category includes ChainPoll [14], LMvLM [8], and SelfCheck-GPT [31]. **(2) Uncertainty-based methods**, which estimate hallucination likelihood based on model uncertainty, including P(True) [21], LN-Entropy [40], and Perplexity (PPL) [30]. **(3) Self-awareness-based methods**, which rely on internal model representations to detect hallucinations, such as UQAC [25] and EigenScore [5]. **(4) LLM-as-Critic models**, including `GPT-4o` [1] and `Qwen2.5-32B` [54], which act as external evaluators of reasoning traces. **(5) Process reward models**, such as `Qwen2.5-Math-7B-PRM800K` [59] and `Qwen2.5-Math-PRM-7B` [58], trained with step-level supervision for reasoning evaluation. **(6) Length-based scoring**, motivated by recent findings that longer reasoning traces are more prone to hallucinations [57], we include `Length-Score`, which directly uses the length of the reasoning trace as its hallucination score.

I Implementation Details for Reasoning Hallucination Detection

We conduct all experiments on machines equipped with NVIDIA A6000 GPUs and 52-core Intel(R) Xeon(R) Gold 6230R CPUs running at 2.10GHz. We utilize the Huggingface `Transformers` and `TRL` libraries to implement and run our experiments. During response generation, we use random sampling with a temperature of 0.7 and a maximum decoding length of 15,000 tokens for Math tasks and 10,000 tokens for all other tasks. For Reasoning Hallucination Detection (RHD), we perform two-fold validation to select optimal hyperparameters, while baselines are tuned within the ranges specified in their original works. To ensure stability, all randomized experiments are repeated three times and the average results are reported.

We perform a grid search over the interval $[0, 1]$ with a step size of 0.1 to determine the best combination of reasoning score weights α_1, α_2, α_3, and α_4 using two-fold validation to select the hyperparameters. For R1-7B, the best weights in the `Math` domain are $\alpha_1 = 0$, $\alpha_2 = 0.4$, $\alpha_3 = 0$, and $\alpha_4 = 0.3$ for the Multi-Trace Ranking setting, and $\alpha_1 = 0$, $\alpha_2 = 0.9$, $\alpha_3 = 0.8$, and $\alpha_4 = 0.4$ for the Binary Detection setting. In the `Science` domain, the best weights are $\alpha_1 = 0.1$, $\alpha_2 = 1.0$, $\alpha_3 = 0$, and $\alpha_4 = 0$ for Multi-Trace Ranking, and $\alpha_1 = 0$, $\alpha_2 = 0.7$, $\alpha_3 = 0.2$, and $\alpha_4 = 0$ for Binary Detection. In the `MultiHopQA` domain, the best weights are $\alpha_1 = 0.4$, $\alpha_2 = 0.1$, $\alpha_3 = 0.6$, and $\alpha_4 = 0.4$ for Multi-Trace Ranking, and $\alpha_1 = 0$, $\alpha_2 = 0$, $\alpha_3 = 0.3$, and $\alpha_4 = 0$ for Binary Detection.

For R1-14B, the best weights in the `Math` domain are $\alpha_1 = 0.3$, $\alpha_2 = 0.7$, $\alpha_3 = 0.1$, and $\alpha_4 = 0.1$ for Multi-Trace Ranking, and $\alpha_1 = 0$, $\alpha_2 = 0.3$, $\alpha_3 = 1.0$, and $\alpha_4 = 0.2$ for Binary Detection. In the `Science` domain, we obtain $\alpha_1 = 0$, $\alpha_2 = 0.5$, $\alpha_3 = 0.5$, and $\alpha_4 = 0.1$ for Multi-Trace Ranking, and $\alpha_1 = 0$, $\alpha_2 = 0$, $\alpha_3 = 0.8$, and $\alpha_4 = 0.1$ for Binary Detection. In the `MultiHopQA` domain, the optimal weights are $\alpha_1 = 0.7$, $\alpha_2 = 0.9$, $\alpha_3 = 0.1$, and $\alpha_4 = 0.0$ for Multi-Trace Ranking, and $\alpha_1 = 1.0$, $\alpha_2 = 0$, $\alpha_3 = 0.1$, and $\alpha_4 = 0.1$ for Binary Detection.

Candidate reasoning score layers $\mathcal{J}$ are selected from $\{14, 16, 18, 20, 22, 24, 26\}$ for R1-7B and from $\{32, 36, 40, 42, 44, 46\}$ for R1-14B, while attention score layers $\mathcal{L}$ are fixed across models as $\{1, 3, 5, 7, 9, 11, 13\}$. The models used in our experiments, `DeepSeek-R1-Distill-Qwen-7B` and `DeepSeek-R1-Distill-Qwen-14B`, are publicly available

Table 4: Ablation study of the RHD model on three different domains of ReTruthQA. Each row removes one component of the hallucination score.

Model	Variant	MATH			Science			MultiHopQA		
		MC1	MC2	MC3	MC1	MC2	MC3	MC1	MC2	MC3
R1-7B	RHD	0.6591	0.4765	0.5699	0.6207	0.5448	0.6009	0.7660	0.6255	0.7103
	RHD (w/o $\text{Avg}(\mathcal{R}_{\text{score}})$)	0.6591	0.4765	0.5699	0.6128	0.5307	0.5934	0.7383	0.6032	0.7082
	RHD (w/o CV Score)	0.6364	0.4663	0.5330	0.4483	0.3862	0.4977	0.7447	0.6043	0.6996
	RHD (w/o Attention Score)	0.6591	0.4765	0.5699	0.6207	0.5448	0.6009	0.6383	0.5372	0.6123
	RHD (w/o PCC Score)	0.5909	0.3830	0.5210	0.6207	0.5448	0.6009	0.6809	0.5553	0.6323
R1-14B	RHD	0.3692	0.3005	0.4644	0.6667	0.4714	0.5671	0.5785	0.4421	0.5154
	RHD (w/o $\text{Avg}(\mathcal{R}_{\text{score}})$)	0.3538	0.2867	0.4847	0.7241	0.4609	0.5531	0.5589	0.4284	0.5290
	RHD (w/o CV Score)	0.3692	0.2882	0.4725	0.6470	0.4484	0.5332	0.5455	0.4273	0.5403
	RHD (w/o Attention Score)	0.3231	0.2692	0.4503	0.6724	0.4511	0.5190	0.5702	0.4322	0.5180
	RHD (w/o PCC Score)	0.3692	0.2882	0.4725	0.6724	0.4601	0.5683	0.5785	0.4421	0.5154

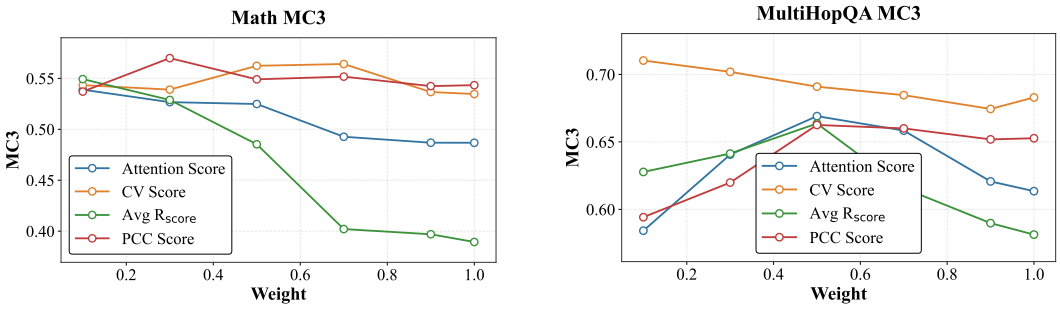
Table 5: Impact of selecting candidate layers from different depth layers of LRMs.

Layers	Math			Science			MultiHopQA		
	MC1	MC2	MC3	MC1	MC2	MC3	MC1	MC2	MC3
High	0.6591	0.4765	0.5699	0.6207	0.5448	0.6009	0.7234	0.5957	0.6799
Middle	0.6591	0.4765	0.5699	0.6207	0.5448	0.6009	0.7021	0.5862	0.6678
Low	0.6591	0.4765	0.5699	0.6207	0.5448	0.6009	0.7660	0.6255	0.7103

at `https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B` and `https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-14B`, respectively.

J Ablation Study of RHD

In this section, we analyze the contribution of each module within the RHD model to reasoning hallucination detection. As shown in Table 4, removing any single component leads to a significant performance drop on most datasets in the Reasoning Hallucination Detection task. This validates the effectiveness of adopting a multivariate regression formulation, where all components jointly serve as covariates. Although some coefficients may appear less influential in certain domains, they demonstrate notable impact in others. This observation suggests that different domains exhibit distinct hallucination pattern preferences, further supporting the validity of the empirically discovered patterns, which can be effectively leveraged for reasoning hallucination detection.



ReTruthQA (Math) ReTruthQA (MultiHopQA)

Figure 7: We conduct a sensitivity analysis of each module in RHD, using R1-7B on the Math and MultiHopQA subsets of ReTruthQA. We vary the weights assigned to different components and observe the resulting performance on the MC3 metric.

K Sensitivity Analysis of RHD

In this section, we conduct sensitivity analysis experiments to investigate the impact of design choices in RHD. Inspired by the underlying reasoning mechanism, we fix the reasoning score to be extracted from the later layers of LRMs. Our primary focus is on selecting the appropriate layers for computing the attention score. Specifically, we evaluate three different layer groups: shallow layers (1, 3, 5, 7, 9, 11, 13), middle layers (8, 10, 12, 14, 16, 18), and deep layers (14, 16, 18, 20, 22, 24, 26) on R1-7B. The experimental results are shown in Table 5. We observe that, across the `Math` and `Science` domains, the choice of attention layers has limited influence on final performance. In contrast, for the `MultiHopQA` domain, shallow layers yield stronger results, aligning with the mechanistic interpretation that earlier layers are primarily responsible for information transmission. Based on these findings, we select the shallow layers as candidate layers for computing the attention score.

We further perform sensitivity analysis on influential feature weights in RHD across domains. We vary the feature weights in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$, and present the results in Figure 7. We observe that most features exhibit an initial increase in performance followed by either a decline or stabilization. The limited variance across settings indicates that the model is not overly sensitive to individual hyperparameter values, demonstrating the robustness and stability of the RHD framework.

L Implementation Details for Reasoning Hallucination Mitigation

We fine-tune the models for reasoning hallucination mitigation using a RL framework with the following hyperparameters: batch size of 8, learning rate of 1.0×10^{-6}, and 1 training epoch. We enable gradient checkpointing to reduce memory usage. The model is configured with a maximum prompt length of 512 and a maximum completion length of 7680. For parameter-efficient tuning, we adopt LoRA with rank $r = 16$ and $\alpha = 16$, applied to all linear layers (`lora_target_modules=all-linear`). During each training step, we sample 16 generations per query.

The reward function is a weighted sum of three components: (1) an accuracy reward that combines a rule-based parser [20] and LLM-as-a-Judge [27] to determine correctness, addressing the issue where the final answer is correct but fails rule-based extraction (reward = 1 for correct, 0 for incorrect); (2) a **format reward** that ensures adherence to the required reasoning format `<think>\n...\n</think>\n<answer>\n...\n</answer>` (reward = 1 if the format is correct, 0 otherwise); and (3) a **tag count reward** that softly encourages the inclusion of each of the four required tags (`<think>`, `</think>`, `<answer>`, `</answer>`) by assigning 0.25 for each tag present. The reward weights are set to 1.0, 0.1, and 0.1 for the accuracy, format, and tag count rewards, respectively.

For evaluation, we use the same accuracy-based metric as in training, and report results by averaging over four sampled generations per input. The fine-tuned model, `DeepSeek-R1-Distill-Qwen-1.5B`, is publicly available at `https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B`.

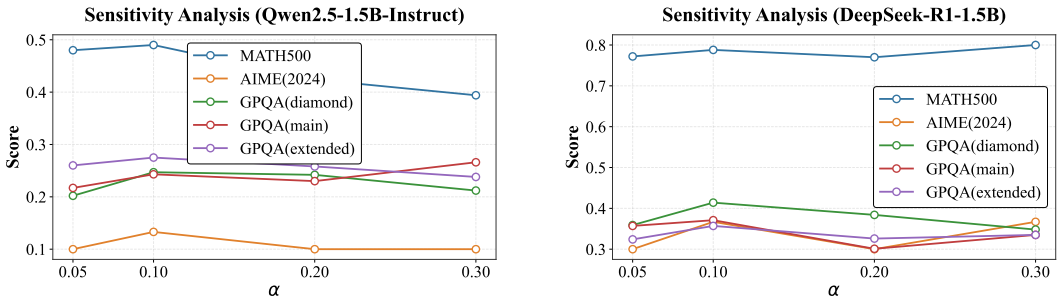


Figure 8: We conduct a sensitivity analysis on the weight of the reasoning score reward in GRPO-R, evaluating its impact on the accuracy metric. Experiments are carried out on both Qwen2.5-1.5B-Instruct and DeepSeek-R1-1.5B by varying the weight parameter α.

Table 6: Accuracy of distilled models across benchmarks using different sampling strategies. Distillation is performed on Qwen2.5-1.5B-Instruct using reasoning traces from R1-14B.

Method	MATH500	AIME (2024)	GPQA (diamond)	GPQA (main)	GPQA (extended)
Qwen2.5-1.5B-Instruct	0.466	0.100	0.202	0.197	0.211
Random 20%	0.504	0.100	0.247	**0.230**	0.242
RHD 20%	**0.520**	0.100	**0.263**	0.210	**0.249**
Random 50%	0.488	0.033	0.187	0.248	**0.266**
RHD 50%	**0.516**	**0.200**	**0.247**	**0.250**	0.242
100%	0.488	0.100	0.217	0.210	0.214

M Sensitivity Analysis of Reasoning Score Weight in GRPO-R

To investigate the sensitivity of the reasoning score reward weight α in the GRPO-R objective, we conduct experiments on both `DeepSeek-R1-1.5B` and `Qwen2.5-1.5B-Instruct`. We vary α in the range $[0.05, 0.1, 0.2, 0.3]$ and evaluate the models' performance accordingly.

Experimental results in Figure 8 indicate that both models achieve the best average performance when $\alpha = 0.1$. As α increases beyond this value, we observe a gradual decline in performance. These results suggest that incorporating the reasoning score reward can effectively mitigate reasoning hallucinations without compromising accuracy, as long as it remains a secondary signal. However, overemphasizing the reasoning score (i.e., assigning it a large weight) can lead to a degradation in the model's ability to optimize for correctness, indicating that the reasoning signal should not dominate the outcome-based reward objective.

N RHD-Guided Reasoning Distillation

Distilling long-chain-of-thought data from large reasoning models to fine-tune smaller LLMs has become a widely adopted strategy for improving reasoning capabilities [10]. However, directly fine-tuning small LLMs on raw LRM-generated data risks transferring undesirable reasoning behaviors such as shallow pattern matching or overthinking, potentially introducing reasoning hallucinations into the smaller models. To address this issue, we propose using the RHD score to rank distillation data and select more truthful samples for training.

The distillation setup uses a learning rate of 5.0×10^{-5}, batch size of 8, and LoRA applied to all linear layers with parameters `lora_r` = 16 and `lora_alpha` = 16. We use the training data from the hallucination mitigation experiment where `R1-14B` produces correct answers, along with their corresponding reasoning traces and final answers. We then score each reasoning trace using the RHD metric and sort the data in descending order. The top 20% and 50% of ranked samples are distilled into a smaller model, `R1-1.5B`, and compared against randomly sampled subsets of 20%, 50%, and 100% of the same data.

Results, as shown in Table 6, demonstrate that RHD-guided distillation consistently yields better performance across most evaluation benchmarks. In contrast, distillation using 100% of the raw data results in degraded performance, likely due to noise introduced by hallucinated or low-quality samples. These findings validate the effectiveness of RHD in selecting high-quality data and mitigating reasoning hallucinations in downstream small LLMs during the distillation process.