

FaithRL: Learning to Reason Faithfully through Step-Level Faithfulness Maximization

Runquan Gui^{1,2*}, Yafu Li^{2†}, Xiaoye Qu², Ziyang Liu¹, Yeqiu Chen¹, Yu Cheng³

¹University of Science and Technology of China, Hefei, China

²Shanghai AI Laboratory, Shanghai, China

³Nanyang Technological University, Singapore

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) has markedly improved the performance of Large Language Models (LLMs) on tasks requiring multi-step reasoning. However, most RLVR pipelines rely on sparse outcome-based rewards, providing little supervision over intermediate steps and thus encouraging over-confidence and spurious reasoning, which in turn increases hallucinations. To address this, we propose **FaithRL**, a general reinforcement learning framework that directly optimizes reasoning faithfulness. We formalize a faithfulness-maximization objective and theoretically show that optimizing it mitigates over-confidence. To instantiate this objective, we introduce a geometric reward design and a faithfulness-aware advantage modulation mechanism that assigns step-level credit by penalizing unfaithful steps while preserving valid partial derivations. Across diverse backbones and benchmarks, FaithRL consistently reduces hallucination rates while maintaining (and often improving) answer correctness. Further analysis confirms that FaithRL increases step-wise reasoning faithfulness and generalizes robustly.

1 Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) has substantially advanced LLM reasoning capabilities in complex domains such as mathematics and coding (DeepSeek-AI, 2025). However, current paradigms relying on sparse outcome-based rewards often neglect intermediate process validity. Since successfully guessing the answer yields a positive reward while explicit refusal is often penalized, models are incentivized toward *over-confidence*, favoring speculative answering rather than acknowledging uncertainty (Chen et al., 2025b; Kalai et al., 2025). Furthermore, the

*Work done during Runquan Gui’s internship at Shanghai AI Laboratory. Yafu Li served as the project lead.

†Corresponding author: yafuly@gmail.com.

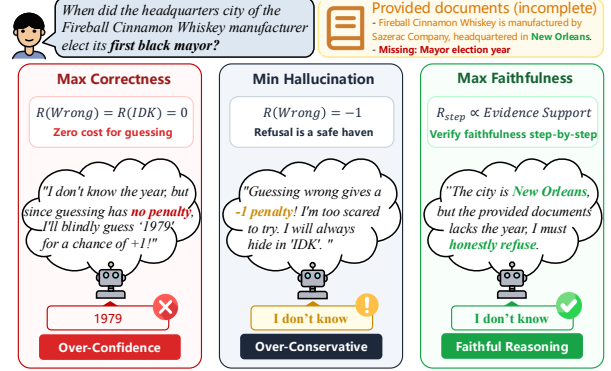


Figure 1: **Illustration of the alignment dilemma.** Maximizing correctness imposes zero cost on guessing, while minimizing hallucinations leads to high refusal rates due to the fear of penalties. FaithRL overcomes this flaw by enforcing step-level faithfulness verification, ensuring honest and logically sound reasoning.

scalar nature of the reward obscures step-level fidelity, reinforcing erroneous steps that coincidentally yield correct answers while penalizing valid partial derivations (Agarwal et al., 2024; Huang et al., 2025b). Consequently, restoring faithfulness necessitates that models act honestly by refusing to predict when evidentiary support is insufficient, and logically by ensuring every reasoning step contributes validly to the final result (Wang et al., 2025b; Chen et al., 2025a).

Recently, numerous approaches have been proposed to improve model faithfulness. A prominent line of work constructs refusal datasets by treating consistent model failures as unanswerable instances and supervising models to generate explicit refusals (Zhang et al., 2024a). However, tying refusal behavior to early-stage failures ignores the fact that model capabilities evolve during training. As a result, such methods can bias models toward *over-conservativeness*, degrading reasoning performance while inflating refusal rates. Complementary efforts focus on verifying intermediate

reasoning steps, typically using factuality-driven metrics (Ren et al., 2025; Li and Ng, 2025). While effective at enforcing factual correctness, these approaches often struggle to guarantee logical soundness: models may produce chains composed of factually correct but irrelevant statements, failing to form a valid deductive path.

To address these challenges, we propose Faithfulness-aware Reinforcement Learning (**FaithRL**), a general reinforcement learning (RL) framework that promotes faithful reasoning. Instead of rewarding only the final answer, it encourages trajectories whose intermediate steps are supported by the required evidence and discourages unsupported steps even when they accidentally lead to a correct outcome. Concretely, we (i) formalize *maximizing reasoning faithfulness* as an objective and show that it avoids degenerate behaviors such as over-confidence and over-conservativeness; (ii) design a *geometric reward* that balances answering and refusal according to baseline capability, preventing both reward-hacking via guessing and blanket refusal; and (iii) introduce *faithfulness-aware advantage modulation*, which assigns step-level credit by filtering positive updates to faithful steps and concentrating penalties on unfaithful ones.

FaithRL achieves consistent performance improvements across multiple benchmarks. When evaluated on the Llama-3.1-8B-Instruct model, FaithRL increases correctness by 5.9 points and reduces the hallucination rate by 4.9 points on the full variants of 2WikiMultiHopQA (Ho et al., 2020), HotpotQA (Yang et al., 2018), and MuSiQue (Trivedi et al., 2022), compared to the average performance of RLVR methods. Furthermore, the approach demonstrates robust generalization capabilities: on out-of-distribution tasks such as MATH500 (Hendrycks et al., 2021) and GSM8k (Cobbe et al., 2021), FaithRL improves average accuracy by over 10.8 points and decreases the hallucination rate by 8.4 points relative to GRPO. Additionally, FaithRL consistently enhances reasoning faithfulness, increasing the proportion of faithful steps by 28.7 points in-domain and 6.5 points out-of-distribution.

The main contributions of this work are summarized as follows. First, we propose the novel optimization objective of *maximizing reasoning faithfulness* and theoretically prove that it prevents the model from exhibiting over-confidence or over-conservativeness. Second, we introduce FaithRL,

a unified RL framework designed to strictly align the incentives of the model with this objective. To the best of our knowledge, this is the first study to implement step-level supervision driven by faithfulness, laying a foundation for future research. Finally, FaithRL consistently outperforms strong baselines across three tasks. On Llama-3.1-8B-Instruct, it achieves an improvement of **28.7** points in the ratio of faithful reasoning compared with GRPO, while also realizing an average reduction of **4.9** points in hallucination and a gain of **5.9** points in correctness compared with RLVR methods.

2 Related Work

Hallucination Mitigation Hallucination mitigation must balance correct answers with reliable refusals for unanswerable queries. Refusal-Aware Instruction Tuning (RAIT) trains models to emit refusal indicators for unknown questions, such as "I don't know" (Cheng et al., 2024; Yang et al., 2024) or uncertainty markers (Zhang et al., 2024a). Methods such as CRaFT (Zhu et al., 2025) reduce over-refusal by fine-tuning on high-certainty and unsolvable samples, but these supervised approaches may still constrain general reasoning. Confidence-based strategies use model uncertainty for inference-time abstention (Jurayj et al., 2025), yet require precise calibration (Sun et al., 2025; Wang et al., 2025a; Damani et al., 2025) and fixed thresholds that can be sensitive to task distributions. Recent RLVR methods encourage refusal by penalizing hallucinations with multi-dimensional (Chen et al., 2025a) or multi-level rewards (Wei et al., 2025), but such coarse outcome rewards offer limited guidance for the intermediate reasoning process.

Reasoning Process Verification In multi-hop QA, improving the structure of the reasoning process has proven effective in enhancing accuracy (Yao et al., 2023; Gui et al., 2025) and efficiency (Gui et al., 2026; Chen et al., 2026). However, structural improvements alone do not ensure faithfulness: because outcome-driven RL supervises only final results, optimized LLMs may still produce reasoning chains that are weakly aligned with the target objective. Existing work on this problem has primarily focused on factual errors (Huang et al., 2025a; Agarwal et al., 2024; Chen et al., 2025b). Accordingly, many verification methods decompose responses into atomic claims for retrieval-based fact-checking, as in HalluMeasure (Akbar et al., 2024), KnowHalu (Zhang et al.,

2024b), and related work (Paul et al., 2024; Ren et al., 2025; Li and Ng, 2025). Although these claim-centric methods can identify factual inaccuracies, they are less suited to detecting invalid logical processes.

3 Preliminary

3.1 Problem Formulation

We define the multi-hop QA task over a universal knowledge base $\mathcal{S}$, a set of questions $\mathcal{Q}$, and an answer space $\mathcal{A}$. For each query $q \in \mathcal{Q}$, we designate a specific context $\mathcal{K}(q) \subset \mathcal{S}$, which consists of a set of documents associated with the query. An LLM $\mathcal{M}$ takes the pair $(q, \mathcal{K}(q))$ as input to generate a sequence of tokens $\tau = (t_1, \dots, t_L)$. To facilitate step-level analysis, we structurally decompose the trajectory into discrete reasoning steps and a final answer, denoted as $\tau = (s_1, \dots, s_k, a)$, where $s_{1:k}$ represents the reasoning process. Let $A^* \in \mathcal{A}$ be the ground truth.

We postulate that $\mathcal{S}$ is informationally complete:

Assumption 3.1 (Knowledge Completeness). For any $q \in \mathcal{Q}$, $\mathcal{S}$ contains sufficient information to derive the correct answer A^* .

Based on this assumption, we define the evidence set $\mathcal{E}(q)$ as the minimal subset of $\mathcal{S}$ required to answer q . In standard multi-hop reasoning tasks, the reasoning chain is universally designed to be unique, thereby ensuring the uniqueness of $\mathcal{E}(q)$. From this definition, we derive the following corollary regarding the answerability of a question:

Corollary 3.2 (Answerability Partition). *Based on the evidence availability, the query set $\mathcal{Q}$ is partitioned into two disjoint subsets: the Answerable Set $\mathcal{Q}_{ans}$ and the Unanswerable Set $\mathcal{Q}_{unans}$.*

$$\mathcal{Q}_{ans} = \{q \in \mathcal{Q} \mid \mathcal{E}(q) \subseteq \mathcal{K}(q)\} \quad (1)$$

$$\mathcal{Q}_{unans} = \mathcal{Q} \setminus \mathcal{Q}_{ans} \quad (2)$$

For queries identified as unanswerable ($q \in \mathcal{Q}_{unans}$), we explicitly modify the ground truth to $A^* = \text{"IDK"}$ and the evidence set to $\mathcal{E}(q) = \mathcal{E}(q) \cup \{\text{IDK}\}$.

3.2 Process-Outcome Consistency

Let $\mathcal{T}$ denote the space of reasoning trajectories. We define two operators for any $\tau \in \mathcal{T}$:

1. **Knowledge extraction** $\phi : \mathcal{T} \rightarrow 2^{\mathcal{S}}$ (the power set of $\mathcal{S}$) identifies the set of knowledge items used in τ .

2. **Answer Parsing** $\psi : \mathcal{T} \rightarrow \mathcal{A} \cup \{\text{IDK}\}$ extracts the final answer.

We define the set of *hallucinated answers* as $\mathcal{H} = \mathcal{A} \setminus \{A^*, \text{IDK}\}$. Modeling the LLM as a stochastic policy π_θ , we connect evidence use to answer outcomes through two empirically grounded assumptions. First, correctness sufficiency states that if a trajectory covers the required evidence set, the model has enough information to derive the correct answer, even with redundant steps:

Assumption 3.3 (Correctness Sufficiency).

$$\mathcal{E}(q) \subseteq \phi(\tau) \implies \psi(\tau) = A^* \quad (3)$$

Conversely, hallucination prevention states that if a trajectory relies only on verified evidence, unsupported answer candidates should be suppressed:

Assumption 3.4 (Hallucination Prevention).

$$\phi(\tau) \subseteq \mathcal{E}(q) \implies \psi(\tau) \notin \mathcal{H} \quad (4)$$

These assumptions capture the primary process–outcome relationships: evidence-complete reasoning generally promotes correctness, whereas evidence-grounded reasoning mitigates hallucination. Validation on the MuSiQue-Full dataset (Appendix A) corroborates this perspective, demonstrating mean ratios for correctness sufficiency and hallucination prevention of **82.0%** and **92.8%**, respectively. These findings motivate the optimization toward reasoning that is both evidence-complete and evidence-grounded.

3.3 Group Relative Policy Optimization

GRPO (Shao et al., 2024) forms the basis of our RL framework by replacing the value function with group-based advantage estimation. For a query q , GRPO samples N outputs $\{\tau_1, \dots, \tau_N\}$ from the old policy $\pi_{\theta_{old}}$. A reward function $r(\tau_i)$ scores the final outcome, usually with a sparse binary signal:

$$r(\tau_i) = \begin{cases} 1, & \text{if } \psi(\tau_i) = A^* \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

The advantage A_i for the i -th output normalizes rewards within the group: $A_i = (r(\tau_i) - \mu_r) / \sigma_r$, where μ_r and σ_r denote the mean and standard deviation of group rewards.

GRPO adapts the PPO surrogate objective to this group setting as the expectation over the token-level clipped surrogate loss:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}_{q, \{\tau_i\}} \left[\frac{1}{N} \sum_{i=1}^N \frac{1}{|\tau_i|} \sum_{t=1}^{|\tau_i|} \mathcal{L}_{i,t}^{CLIP}(\theta) \right] \quad (6)$$

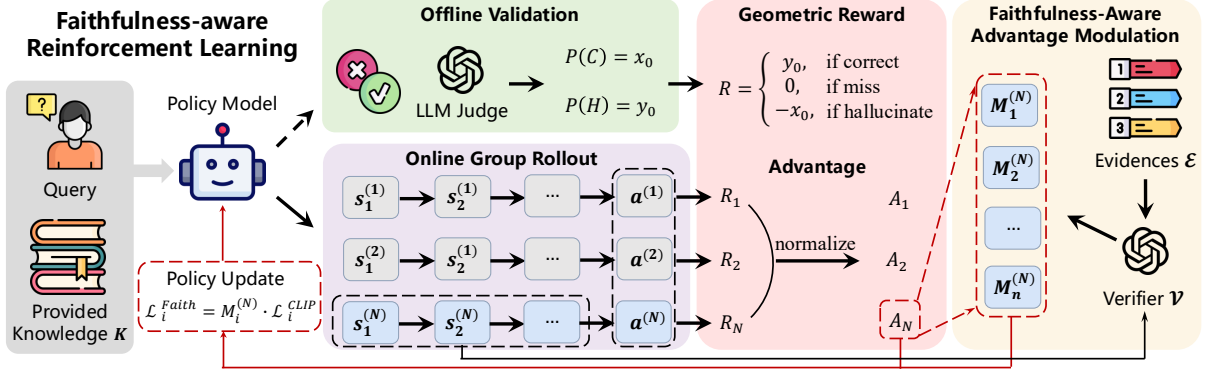


Figure 2: Overall framework of FaithRL. Taking a query and knowledge $\mathcal{K}$, the model first uses offline validation to set baseline rates (x_0, y_0) . During online group rollout, these rates form a geometric reward based on answer correctness. Simultaneously, the verifier $\mathcal{V}$ checks alignment with evidence $\mathcal{E}$ to derive token-level advantage modulation for the policy update.

Here, $\mathcal{L}_{i,t}^{CLIP}(\theta)$ is the standard clipped objective term: $\mathcal{L}_{i,t}^{CLIP}(\theta) = \min(\rho_{i,t} A_i, \text{clip}(\rho_{i,t}, 1 - \epsilon, 1 + \epsilon) A_i)$, where $\rho_{i,t} = \frac{\pi_{\theta}(\tau_{i,t}|q, \tau_{i,t} < t)}{\pi_{\theta_{old}}(\tau_{i,t}|q, \tau_{i,t} < t)}$ is the probability ratio between current and reference policies. The KL regularization term is omitted for brevity.

4 Methodology

In this section, we introduce the FaithRL framework in detail, with the overall process illustrated in Figure 2.

4.1 Optimization Objectives

We assess model performance by partitioning the trajectory space $\mathcal{T}$ into three outcomes. The *correct set* $S_c = \{\tau \in \mathcal{T} \mid \psi(\tau) = A^*\}$ includes trajectories where the model provides the correct answer or properly refuses an unanswerable query. The *miss set* $S_m = \{\tau \in \mathcal{T} \mid \psi(\tau) = \text{IDK} \wedge A^* \neq \text{IDK}\}$ captures false refusals, where the model incorrectly declines to answer a valid question. All remaining errors form the *hallucination set* $S_h = \{\tau \in \mathcal{T} \mid \psi(\tau) \in \mathcal{H}\}$. We denote the probabilities of these outcomes for a policy π_{θ} as correctness $P(C)$, miss rate $P(M)$, and hallucination rate $P(H)$.

Conventional QA optimization typically prioritizes one of two objectives:

- **Objective A (maximize correctness):** Maximize $P(C)$. This is the standard objective for general QA (Li et al., 2024; Huang et al., 2025a).
- **Objective B (minimize hallucination):** Minimize $P(H)$. This is often prioritized in safety-critical applications (Wei et al., 2025).

We instead propose **objective C (maximize reasoning faithfulness)**. It targets the *reasoning-faithful set* $S_f = \{\tau \in \mathcal{T} \mid \phi(\tau) = \mathcal{E}(q)\}$, where the used knowledge strictly matches the required evidence. We maximize $P(F) = P(\tau \in S_f)$. Under Assumptions 3.3 and 3.4, increasing $P(F)$ should improve accuracy and honesty together: evidence-complete trajectories are more likely to reach the correct answer, while evidence-grounded trajectories are less likely to hallucinate. This gives the optimizer a process-level target, favoring deductive steps over final-answer guessing.

The theorem below contrasts these objectives: standard objectives push toward extreme refusal rates, whereas reasoning faithfulness ties refusal to evidence-grounded derivation.

Theorem 4.1 (Asymptotic stability of refusal strategy). *Consider a model with imperfect intrinsic capability. Optimizing the three objectives yields distinct asymptotic behaviors for the miss rate $P(M)$:*

1. **Objective A (maximize correctness)** forces $P(M) \rightarrow 0$, causing **over-confidence**: the model speculates on all queries to maximize rewards.
2. **Objective B (minimize hallucination)** drives $P(M) \rightarrow 1$, causing **over-conservativeness**: the model refuses answerable queries to avoid hallucination risks.
3. **Objective C (maximize reasoning faithfulness)** ensures **stability**, maintaining a miss-rate equilibrium determined by the model’s intrinsic capability and preventing the collapse observed under objectives A and B.

Proofs are provided in Appendix B.1.

4.2 Faithfulness-aware Reinforcement Learning

We propose FaithRL, an RL framework designed to maximize the faithfulness objective $P(F)$. To achieve this, FaithRL integrates a geometric reward mechanism that aligns outcome-level optimization with the model’s intrinsic capability, and a faithfulness-aware advantage modulation strategy that enforces step-level process supervision.

4.2.1 Geometric Reward Design

Existing strategies for optimizing reasoning tasks often rely on independent metrics for correctness and hallucination, which may fail to accurately quantify the genuine performance gain after optimization. To simultaneously address helpfulness and honesty without relying on arbitrary hyperparameters, we adopt the **truthful helpfulness score (THS)** (Zhu et al., 2025) as a higher-order metric that provides a geometrically intuitive fusion of utility and honesty.

As seen in Figure 3, we represent model capability on a two-dimensional plane, where the x-axis denotes correctness and the y-axis denotes hallucination. Let $E_0 = (x_0, y_0)$ be the baseline point, E_1 the optimized point, O the origin, and $E^*(1, 0)$ the ideal point. THS is defined as the **normalized signed area**: $\vec{S}_{\Delta OE_0 E_1} / \vec{S}_{\Delta OE_0 E^*}$. Detailed mathematical definitions are provided in Appendix B.2.

Guided by THS, we derive a reward mechanism whose coefficients come directly from the baseline coordinates (x_0, y_0) :

$$\mathcal{R}_{geo}(\tau) = \begin{cases} +y_0, & \text{if } \tau \in S_c \\ 0, & \text{if } \tau \in S_m \\ -x_0, & \text{if } \tau \in S_h \end{cases} \quad (7)$$

This capability-dependent reward gives correctness and hallucination penalties according to the model’s starting point, aligning optimization with the THS trajectory.

Theorem 4.2 (Geometric alignment with THS).

Let $J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[\mathcal{R}_{geo}(\tau)]$ be the expected return under the proposed geometric reward. The optimization gradient is strictly proportional to the gradient of THS with a positive coefficient C :

$$\nabla_\theta J(\theta) = C \cdot \nabla_\theta THS(\pi_\theta), \quad \text{where } C > 0$$

This implies that maximizing the geometric reward is equivalent to maximizing the signed area improvement relative to the baseline capability.

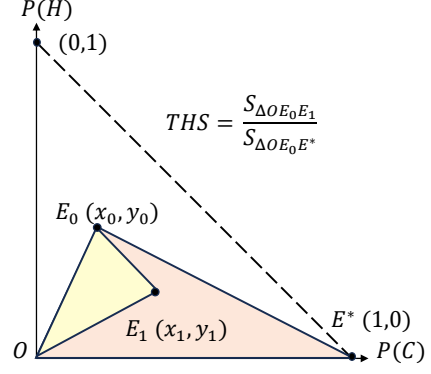


Figure 3: Truthful Helpfulness Score (THS).

The complete proof is provided in Appendix B.3.

4.2.2 Faithfulness-Aware Advantage Modulation

Geometric rewards optimize the final outcome, but they treat the reasoning process as a black box. To ensure faithfulness, we must verify that every single step contributes validly to the result. This requires supervision at the process level. We introduce **faithfulness-aware advantage modulation (FAAM)** to strictly enforce this step-wise fidelity.

Adopting the structure from Section 3.1, we decompose the trajectory τ into discrete reasoning steps $(s_1, \dots, s_k)$ and a final answer. We then employ a verifier $\mathcal{V}$ to assess whether each step s_j is strictly supported by the evidence $\mathcal{E}(q)$. Formally, we define $\mathcal{V}(s_j) = \mathbb{I}[\phi(s_j) \subseteq \mathcal{E}(q)]$, where 1 signifies a faithful step and 0 denotes an unfaithful one. This verification is merely an objective check for the utilization of evidence from $\mathcal{E}(q)$ in the step.

We integrate this verification mechanism into the GRPO framework. First, we calculate the advantage A_i for each trajectory using the geometric rewards $\mathcal{R}_{geo}(\tau_i)$ from Equation 7. To apply supervision at the step level, we introduce a modulation scalar $M_{i,t}$ for each token. Let $\alpha \in [0, 1]$ be the faithfulness coefficient. For any token t within step s_j , we define the modulation term as:

$$M_{i,t} = \begin{cases} (1 - \alpha)\mathcal{V}(s_j) + \alpha, & \text{if } A_i > 0 \\ (1 - \alpha)(1 - \mathcal{V}(s_j)) + \alpha, & \text{if } A_i \leq 0 \end{cases}$$

This design functions as a credit assignment filter. For trajectories with a positive advantage ($A_i > 0$), we reinforce valid reasoning: faithful steps get the full reward ($M_{i,t} = 1$), and unfaithful steps a discounted reward ($M_{i,t} = \alpha$). For trajectories with a negative advantage, by contrast,

Table 1: Main results on Llama3.1-8B-Instruct comparing **FaithRL** against strong baselines across three multi-hop QA benchmarks. We report **correct rate (C)** and **truthful helpfulness score (THS)** where higher is better ($\uparrow$), and **hallucination rate (H)** where lower is better ($\downarrow$). The best performance is highlighted in **bold**, and the second best is underlined.

Method	2WikiMultiHopQA-Full			HotpotQA-Full			MuSiQue-Full			Average		
	C ($\uparrow$)	H ($\downarrow$)	THS ($\uparrow$)	C ($\uparrow$)	H ($\downarrow$)	THS ($\uparrow$)	C ($\uparrow$)	H ($\downarrow$)	THS ($\uparrow$)	C ($\uparrow$)	H ($\downarrow$)	THS ($\uparrow$)
<i>Llama3.1-8B-Instruct</i>												
PROMPTING BASELINE												
Prompting	69.2	24.4	0.0	63.7	33.9	0.0	54.1	32.8	0.0	62.3	30.4	0.0
REFUSAL-AWARE SFT												
R-Tuning	56.9	12.7	20.9	64.2	<u>10.9</u>	43.7	65.8	12.4	45.3	62.3	12.0	37.7
CRaFT	49.1	14.9	6.8	63.0	10.0	44.2	59.2	6.2	49.0	57.1	<u>10.4</u>	35.8
CONFIDENCE-BASED ABSTENTION												
CTSF	63.5	14.8	21.5	66.6	10.8	46.3	56.8	9.0	42.0	62.3	11.5	38.7
RLCR	60.6	35.8	-40.9	59.3	40.0	-15.9	50.6	41.5	-17.8	56.8	39.1	-23.3
GRPO + CTSF	50.6	46.7	-81.8	68.0	28.3	14.8	58.8	30.7	8.2	59.1	35.2	-13.0
RLVR METHODS												
GRPO	<u>86.2</u>	12.2	51.6	76.1	22.9	33.1	79.7	13.1	58.1	80.7	16.1	47.7
TruthRL	85.3	11.6	<u>52.4</u>	78.1	18.0	44.3	80.1	9.7	<u>64.1</u>	<u>81.2</u>	13.1	<u>54.4</u>
FSPO	77.7	9.2	51.6	78.3	14.0	52.0	63.2	12.4	42.7	73.0	11.8	48.8
KnowRL	78.8	19.5	23.5	<u>81.9</u>	16.7	<u>50.5</u>	75.3	16.9	47.4	78.7	17.7	42.4
FaithRL (Ours)	87.5	9.1	61.7	85.5	11.4	64.1	<u>80.0</u>	<u>8.8</u>	65.5	84.3	9.8	64.2

unfaithful steps face the full penalty, while faithful steps incur a reduced one. This prevents the model from being unfairly penalized for correct partial reasoning steps, even if the final outcome is wrong. Appendix C.1 further clarifies how FAAM differs from generic step-level rewards.

Building upon the GRPO surrogate objective (Eq. 6), we incorporate this modulation to define the FaithRL objective:

$$\mathcal{J}_{\text{FaithRL}}(\theta) = \mathbb{E}_{q, \{\tau_i\}} \left[\frac{1}{N} \sum_{i=1}^N \frac{1}{|\tau_i|} \sum_{t=1}^{|\tau_i|} M_{i,t} \cdot \mathcal{L}_{i,t}^{\text{CLIP}}(\theta) \right]$$

This objective effectively adjusts the importance of each step during training. By dynamically re-weighting the learning signal, FaithRL ensures that the model is reinforced primarily for reasoning that is both correct and faithful.

5 Experiments

5.1 Experimental Setup

Datasets We conduct evaluations on three multi-hop reasoning benchmarks: 2WikiMultiHopQA (Ho et al., 2020), HotpotQA (Yang et al., 2018), and MuSiQue-Full (Trivedi et al., 2022). MuSiQue-Full naturally includes a mix of answerable and unanswerable questions. In contrast, the 2WikiMultiHopQA and HotpotQA datasets only contain answerable queries. To bridge this gap, we synthesize unanswerable samples for them, resulting in the 2WikiMultiHopQA-Full and HotpotQA-Full datasets. Specifically, for each question q , we remove exactly one evidence document from $\mathcal{K}(q)$, which is the set of discrete documents. This manipulation ensures that the reasoning chain is

logically broken, while the remaining knowledge retains high semantic relevance to q . This design strictly compels the model to determine answerability through genuine reasoning, effectively preventing it from exploiting superficial heuristics (e.g., keyword matching) or reward hacking to shortcut the judgment process. We strictly filter these samples to ensure validity (see Appendix D.1). We train our model on the MuSiQue-Full dataset and evaluate performance on the test sets of all three *full* benchmarks.

Metrics We use correctness rate $P(C)$ and hallucination rate $P(H)$ as primary outcome metrics, together with THS (defined in Section 4.2) to summarize their trade-off. The faithful step ratio is an auxiliary process metric: because it is closely related to the training-time faithfulness signal, we use it only to analyze reasoning behavior rather than as independent evidence of outcome improvement.

Models and baselines We evaluate FaithRL against three baseline categories: (1) Refusal-aware SFT (R-Tuning (Zhang et al., 2024a), CRaFT (Zhu et al., 2025)), which learn abstention via refusal data; (2) Confidence-based abstention, which applies CTSF (Jurayj et al., 2025) to the backbone, GRPO, and RLCR models (Damani et al., 2025); (3) RLVR methods, which include outcome-based approaches (GRPO, TruthRL (Wei et al., 2025)) that optimize final correctness, and factuality-driven approaches (FSPO (Li and Ng, 2025), KnowRL (Ren et al., 2025)) that guide intermediate reasoning.

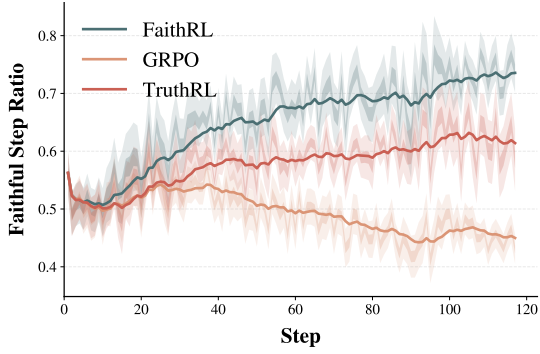


Figure 4: Comparison of the faithful step ratio among FaithRL, TruthRL, and GRPO during training.

5.2 Main Results

Table 1 reports the main Llama-3.1-8B-Instruct results, comparing FaithRL with strong baselines. Qwen-2.5-7B-Instruct results are provided in Appendix Table 9.

FaithRL consistently achieves higher THS than the baselines. It improves the average THS over other RLVR methods (GRPO, TruthRL, FSPO, and KnowRL) by **15.9** points. This advantage is especially clear on HotpotQA-Full, where FaithRL leads the next best method by **12.1** points. These results demonstrate the efficacy of our geometric reward design in balancing utility and honesty.

Beyond THS, FaithRL achieves the highest average correctness while maintaining the lowest hallucination rate. Compared with the average of RLVR methods, FaithRL improves correctness by **5.9** points and reduces hallucinations by **4.9** points. Averaged across all tasks, FaithRL records 84.3% correctness and a 9.8% hallucination rate, surpassing GRPO by 3.6 points in correctness and reducing hallucinations by 6.3 points. As detailed in Appendix E.1, the aggregate correctness gain over GRPO is primarily driven by more accurate refusal on unanswerable questions; on answerable subsets, FaithRL is slightly less aggressive in answering. We therefore interpret the result as an improved honesty–utility trade-off rather than a uniform gain on every subset. We provide more detailed experimental results in Appendix E.1.

5.3 FaithRL Enhances Reasoning Faithfulness

Dynamic Evolution of Reasoning Faithfulness during Training. We first investigate the ability of FaithRL to dynamically refine reasoning patterns throughout the optimization process. In this exper-

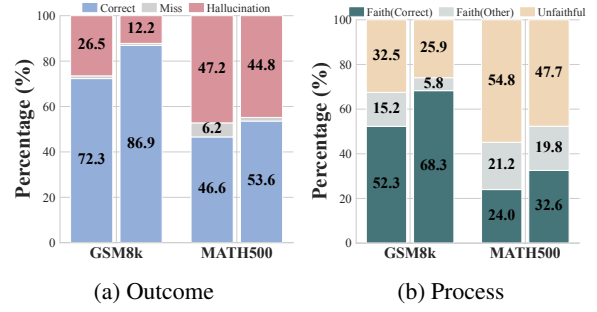


Figure 5: Model performance on OOD tasks. (a) illustrates the outcome correctness distribution across datasets. (b) displays the faithfulness of the reasoning process. For each dataset, the **left** bar in each pair represents the GRPO baseline, and the **right** bar represents FaithRL. *Faithful (Correct)* denotes faithful reasoning steps within correctly answered samples.

iment, we train the model on the MuSiQue-Full dataset with 15,000 samples. For every training step, we monitor the reasoning trajectories of each sample and calculate the *faithful step ratio*, defined as the proportion of reasoning steps strictly supported by evidence relative to the total steps.

As illustrated in Figure 4, FaithRL is the only method that drives a significant and sustained increase in reasoning faithfulness. Specifically, the overall ratio of faithful steps increases from an initial baseline of 51.4% to 74.6%. In contrast, standard RL baselines (GRPO and TruthRL) show no meaningful improvement. This disparity highlights the fundamental advantage of our approach: unlike traditional outcome-oriented optimization, FaithRL provides precise step-level guidance to effectively shape the model’s underlying reasoning process. We provide a comprehensive breakdown of training dynamics and more baselines in Appendix E.2.

Generalization to out-of-distribution domains.

To demonstrate that the faithfulness learned by FaithRL is a robust capability rather than a dataset-specific artifact, we evaluate the trained policies on two out-of-distribution (OOD) benchmarks, GSM8k (Cobbe et al., 2021) and MATH500 (Hendrycks et al., 2021). Policy inference uses only the benchmark questions and requires neither evidence annotations nor a verifier. Ground-truth solution steps are provided to an LLM only after generation to compute the auxiliary faithful step ratio; they do not affect model outputs or the primary outcome metrics. Figure 5 compares FaithRL with GRPO, while Appendix E.3 additionally reports TruthRL. See Appendix D.3

Method	Cor.	Hal.	THS
Prompting	67.8	16.2	0.0
GRPO	82.8	14.4	22.5
+ $\mathcal{R}_{geo}$	79.8	8.8	43.0
+ FAAM	77.8	12.0	27.5
Ours	82.4	7.3	51.8

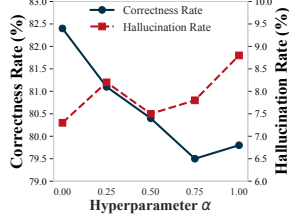


Figure 6: **Ablation studies.** (Left) Component analysis confirming the effectiveness of each module compared to baselines. (Right) Sensitivity analysis of α , showing optimal performance at $\alpha = 0$.

for experimental details.

As shown in Figure 5, FaithRL consistently outperforms GRPO. In terms of performance, FaithRL achieves significantly higher accuracy, surpassing GRPO by 14.6% on GSM8k, while simultaneously maintaining a substantially lower hallucination rate. Regarding faithfulness, FaithRL exhibits a marked improvement in the overall faithful step ratio. Breakdown analysis shows this increase is primarily driven by improved faithfulness within correctly answered samples (rising from 52.3% to 68.3% on GSM8k). This suggests that the performance gains stem from genuine reasoning improvements rather than lucky guesses. We provide detailed results in Appendix E.3.

To test whether this improvement depends on a particular reference solution, we additionally evaluate both policies against alternative reasoning paths generated for all 1,819 GSM8k and MATH500 questions. These paths intentionally differ from the original references in structure, intermediate steps, and notation. FaithRL still improves the overall faithful step ratio from 57.8% to 65.1% over GRPO. This result suggests that the observed gain is not tied to the surface form of a single reference path; full results are provided in Appendix C.3.

5.4 Ablations

Component Analysis. We verify the effectiveness of the two core components in FaithRL: the geometric reward ($\mathcal{R}_{geo}$) and the faithfulness-aware advantage modulation (FAAM). As detailed in Figure 6 (Left), incorporating either $\mathcal{R}_{geo}$ or FAAM individually yields improvements in THS compared to the GRPO baseline. Specifically, adding $\mathcal{R}_{geo}$ alone results in a substantial THS gain of 20.5 points. However, deploying either component in isolation leads to a marked decline in correctness. In contrast, combining both modules enables FaithRL to achieve the lowest hallucination rate

Table 2: Verifier-size ablation. Human alignment is measured on 100 sampled reasoning traces.

Verifier	C	H	THS	Human Align.
Llama-3.3-70B-Inst	84.3	9.8	64.2	88%
Qwen3-4B-Inst-2507	81.9	9.6	62.2	85%
HHEM-2.1	81.4	12.1	56.6	66%

while maintaining high correctness, ultimately securing the highest THS. This demonstrates that $\mathcal{R}_{geo}$ and FAAM work synergistically to enforce faithful reasoning. For detailed experimental configurations and specific numerical results, please refer to Appendix E.5.

Hyperparameter Sensitivity. We further evaluate the sensitivity of our method to the hyperparameter α by varying it across $\{0, 0.25, 0.5, 0.75, 1.0\}$. The results, visualized in Figure 6 (Right), show that decreasing α consistently mitigates hallucinations. In particular, $\alpha = 0$ yields the optimal performance profile, effectively minimizing the hallucination rate compared to the vanilla RL baseline ($\alpha = 1.0$) while preserving high correctness. We provide detailed results in Appendix E.4.

Verifier Size Sensitivity. We further investigate whether FaithRL relies on a large-scale step-level verifier by replacing Llama-3.3-70B-Instruct with smaller alternatives for faithfulness verification. All resulting policies are evaluated with the Llama-3.3-70B-Instruct judge, so the Qwen3 training verifier also provides a cross-family evaluation. As shown in Table 2, the 4B verifier yields performance comparable to that of the 70B verifier, achieving a THS of 62.2 compared to 64.2, alongside similar human alignment (85% versus 88%). Even with the lightweight 110M HHEM-2.1 model (Hughes and Bae, 2023), FaithRL attains a THS of 56.6, representing an absolute improvement of 34.1 over the THS of 22.5 achieved by the GRPO baseline. These results indicate that step-level faithfulness verification is relatively tractable, and that FaithRL remains effective when utilizing substantially smaller or noisier verifiers.

Robustness to Imperfect Verification. We further test FaithRL under controlled verifier errors by randomly flipping step-level verification labels during training. With 25% label noise, FaithRL retains a THS of 43.3, compared with 50.0 under clean verification; even at 50% noise, it maintains a positive THS of 18.1. These results indicate graceful degradation under imperfect supervision,

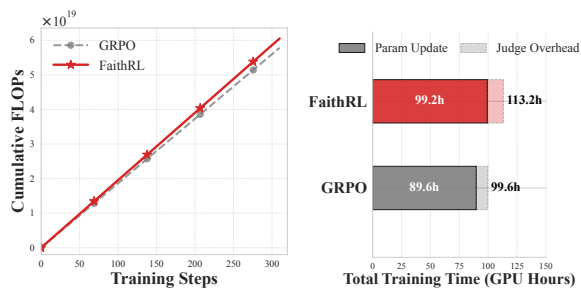


Figure 7: **Training efficiency analysis.** (Left) Comparison of cumulative FLOPs throughout training, showing that FaithRL incurs only a 4.7% increase. (Right) Break-down of total GPU hours, demonstrating that FaithRL limits the overhead to 13.7%.

although randomly flipped labels do not fully represent the structured errors arising from incomplete or retrieved evidence. Full results are provided in Appendix D.5.

5.5 Computational Cost

To quantify the computational overhead introduced by our method, we compared FaithRL against GRPO over one training epoch. We employed FLOPs and GPU hours as primary metrics for a fair comparison of resource consumption across the entire training lifecycle. As Figure 7 shows, despite FaithRL incorporating step-level supervision, the additional computational cost remains within 15% relative to GRPO. This efficiency is primarily attributed to batching multiple step-level faithfulness judgments, effective KV-cache reuse, and concurrent processing of verification requests. Appendix F provides detailed experimental setups and an efficiency analysis.

6 Conclusion

We presented FaithRL, a novel RL framework enhancing reasoning faithfulness through step-level verification. By strictly aligning model incentives with the validity of the intermediate reasoning process, this approach effectively bridges the gap between performance capability and factual honesty. Empirical evaluations demonstrate that FaithRL not only improves correctness and mitigates hallucinations, but also successfully internalizes faithful reasoning patterns. We hope this work inspires further research into faithful reasoning, facilitating the development of LMs that are as reliable in logical derivations as they are accurate in final outcomes.

Limitations

Despite the promising results, we acknowledge three primary limitations. First, our theoretical assumptions do not universally hold for every reasoning trajectory or task distribution. However, our empirical validation in Appendix A shows that these assumptions are satisfied at sufficiently high rates in the studied setting. The purpose of our theoretical analysis is therefore not to claim deterministic guarantees, but to explain the root causes of over-confidence and over-conservativeness under common process–outcome relationships, and to show why maximizing reasoning faithfulness can avoid both failure modes. Second, FaithRL requires process-level annotations, such as evidence chains, and a sufficiently reliable verifier during training. Neither is required at inference time, but reducing this training-time supervision remains an important direction for future work. Our noise study in Appendix D.5 suggests graceful degradation under imperfect verification, although naturally occurring evidence errors may be more structured than the simulated label noise. Third, the integration of an external verifier introduces additional computational overhead. However, empirical analysis confirms that this cost is acceptable (within 15%) given the substantial gains in reliability. Moreover, the Verifier Size Sensitivity analysis in Table 2 shows that FaithRL remains effective even with lightweight verifiers, suggesting that strong performance does not strictly require a large-scale verifier. We leave it to future work to further optimize the verification efficiency.

References

- Chirag Agarwal, Sree Harsha Tanneru, and Himabindu Lakkaraju. 2024. Faithfulness vs. plausibility: On the (un) reliability of explanations from large language models. *arXiv preprint arXiv:2402.04614*.
- Shayan Ali Akbar, Md Mosharaf Hossain, Tess Wood, Si-Chi Chin, Erica M Salinas, Victor Alvarez, and Erwin Cornejo. 2024. Hallumeasure: Fine-grained hallucination measurement using chain-of-thought reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15020–15037.
- Xilun Chen, Iliia Kulikov, Vincent-Pierre Berges, Barlas Oğuz, Rulin Shao, Gargi Ghosh, Jason Weston, and Wen-tau Yih. 2025a. Learning to reason for factuality. *arXiv preprint arXiv:2508.05618*.
- Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, and Ethan Perez. 2025b. Reasoning models don’t always say what they think. *arXiv preprint arXiv:2505.05410*.
- Yeqiu Chen, Ziyang Liu, Zhenxin Huang, Runquan Gui, Hong Wang, and Lei Liu. 2026. Arborkv: Structure-aware KV cache management for scaling tree-based LLM reasoning. In *Forty-Third International Conference on Machine Learning (ICML)*.
- Qinyuan Cheng, Tianxiang Sun, Xiangyang Liu, Wenwei Zhang, Zhangyue Yin, Shimin Li, Linyang Li, Zhengfu He, Kai Chen, and Xipeng Qiu. 2024. Can ai assistants know what they don’t know? *arXiv preprint arXiv:2401.13275*.
- 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. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfelf, Leshem Choshen, Yoon Kim, and Jacob Andreas. 2025. Beyond binary rewards: Training lms to reason about their uncertainty. *arXiv preprint arXiv:2507.16806*.
- DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Runquan Gui, Jie Wang, Zhihai Wang, Chi Ma, Jianye Hao, and Feng Wu. 2026. Short chains, deep thoughts: Balancing reasoning efficiency and intra-segment capability via split-merge optimization. In *Forty-Third International Conference on Machine Learning (ICML)*.
- Runquan Gui, Zhihai Wang, Jie Wang, Chi Ma, Huiling Zhen, Mingxuan Yuan, Jianye Hao, Defu Lian, Enhong Chen, and Feng Wu. 2025. Hypertree planning: Enhancing LLM reasoning via hierarchical thinking. In *Forty-second International Conference on Machine Learning (ICML)*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.
- Lei Huang, Xiaocheng Feng, Weitao Ma, Yuchun Fan, Xiachong Feng, Yangfan Ye, Weihong Zhong, Yuxuan Gu, Baoxin Wang, Dayong Wu, Guoping Hu, and Bing Qin. 2025a. Improving contextual faithfulness of large language models via retrieval heads-induced optimization. *arXiv preprint arXiv:2501.13573*.

- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025b. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.
- Simon Hughes and Minseok Bae. 2023. [Vectara hallucination leaderboard](#).
- William Jurayj, Jeffrey Cheng, and Benjamin Van Durme. 2025. Is that your final answer? test-time scaling improves selective question answering. *arXiv preprint arXiv:2502.13962*.
- Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2025. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*.
- Junyi Li and Hwee Tou Ng. 2025. Reasoning models hallucinate more: Factuality-aware reinforcement learning for large reasoning models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Xiang Li, Shizhu He, Fangyu Lei, Jun Yang, Tianhuang Su, Kang Liu, and Jun Zhao. 2024. Teaching small language models to reason for knowledge-intensive multi-hop question answering. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7804–7816.
- Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. 2024. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. *arXiv preprint arXiv:2402.13950*.
- Baochang Ren, Shuofei Qiao, Da Zheng, Huajun Chen, and Ningyu Zhang. 2025. Knowrl: Exploring knowledgeable reinforcement learning for factuality. *arXiv preprint arXiv:2506.19807*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Zhongxiang Sun, Qipeng Wang, Haoyu Wang, Xiao Zhang, and Jun Xu. 2025. Detection and mitigation of hallucination in large reasoning models: A mechanistic perspective. *arXiv preprint arXiv:2505.12886*.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.
- Changyue Wang, Weihang Su, Qingyao Ai, and Yiqun Liu. 2025a. Joint evaluation of answer and reasoning consistency for hallucination detection in large reasoning models. *arXiv preprint arXiv:2506.04832*.
- Yanbo Wang, Yongcan Yu, Jian Liang, and Ran He. 2025b. A comprehensive survey on trustworthiness in reasoning with large language models. *arXiv preprint arXiv:2509.03871*.
- Zhepei Wei, Xiao Yang, Kai Sun, Jiaqi Wang, Rulin Shao, Sean Chen, Mohammad Kachuee, Teja Gollapudi, Tony Liao, Nicolas Scheffer, Rakesh Wanga, Anuj Kumar, Yu Meng, Wen-tau Yih, and Xin Luna Dong. 2025. Truthrl: Incentivizing truthful llms via reinforcement learning. *arXiv preprint arXiv:2509.25760*.
- Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2024. Alignment for honesty. *Advances in Neural Information Processing Systems*, 37:63565–63598.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. 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.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*.
- Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2024a. R-tuning: Instructing large language models to say ‘i don’t know’. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7106–7132.
- Jiawei Zhang, Chejian Xu, Yu Gai, Freddy Lecue, Dawn Song, and Bo Li. 2024b. Knowhalu: Hallucination detection via multi-form knowledge based factual checking. *arXiv preprint arXiv:2404.02935*.
- Runchuan Zhu, Zhipeng Ma, Jiang Wu, Junyuan Gao, Jiaqi Wang, Dahua Lin, and Conghui He. 2025. Utilize the flow before stepping into the same river twice: Certainty represented knowledge flow for refusal-aware instruction tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26157–26165.

Appendix Contents

A	Experimental Validation of Assumptions	12
B	Theoretical and Metric Details	12
B.1	Proofs of Theorem 1	12
B.2	Metric Details: Truthful Helpfulness Score (THS)	14
B.3	Proof of Theorem 2	14
C	Methodological Clarifications	15
C.1	Faithfulness Modulation vs. Step-level Rewards	15
C.2	Static Geometric Coefficients and THS Training Trajectory	15
C.3	Faithfulness Evaluation with Alternative Reasoning Paths	16
D	Experimental Details	16
D.1	Datasets	16
D.2	Implementation Details of the Baselines	17
D.3	Training Details	18
D.4	Verifier Robustness across Task Difficulty	19
D.5	Robustness to Noisy Verification	19
E	Extended Experimental Results	19
E.1	Additional Results for Main Experiments	19
E.2	Training Dynamics and Step-Level Verification Baselines	19
E.3	Extended Results on OOD Generalization	19
E.4	Extended Results on Hyperparameter Sensitivity	20
E.5	Extended Results on Component Analysis	20
F	Computational Cost	21
G	Case Study	23
G.1	Analysis of Answerable Scenarios	23
G.2	Analysis of Unanswerable Scenarios	23
G.3	Reasoning Faithfulness and Step-wise Contribution	23
H	Prompt Template	24

A Experimental Validation of Assumptions

To empirically validate the theoretical assumptions in Section 3, we analyze the MuSiQue-Full dataset ($N = 39,876$). To fully capture the dynamic satisfaction of these assumptions throughout the training lifecycle, we evaluate both the initial pre-trained model Qwen-2.5-7B-Instruct and the final **FaithRL**-optimized model derived from it.

We design two experimental protocols to verify the causal links between evidence sufficiency, reasoning faithfulness, and outcome validity. For **Assumption 3.3 (Correctness Sufficiency)**, we employ an LLM Judge to assess evidence utilization, isolating trajectories that fully incorporate the complete evidence set $\mathcal{E}(q)$ to measure answer accuracy. For **Assumption 3.4 (Hallucination Prevention)**, we evaluate the logical consistency of reasoning steps against the provided context, filtering for strictly faithful trajectories to measure the hallucination rate. Llama-3.3-70B-Instruct serves as the unified judge. For inference, we set the temperature to 0 and the maximum output length to 4096. All prompt templates used for evaluation are detailed in Appendix H.

The results are summarized in Table 3.

As demonstrated in Table 3, the empirical results strongly support our theoretical framework. For the pre-trained model, the adherence ratios are 72.9% and 89.5%. Post-training with FaithRL, these ratios improve significantly to 91.2% and 96.1%, respectively. Averaging these states to approximate the conditions during training, we find a mean Correctness Sufficiency ratio of **82.0%** and a mean Hallucination Prevention ratio of **92.8%**. These high aggregate metrics confirm that our theoretical assumptions hold statistically significant validity within the domain of knowledge-intensive reasoning.

B Theoretical and Metric Details

B.1 Proofs of Theorem 1

Let Π be the space of stochastic policies parameterized by θ . For a given input q , π_θ induces a distribution over trajectories $\mathcal{T}$. Consistent with Section 4.1, the sample space is partitioned into disjoint sets: S_c (Correct), S_m (Miss), S_h (Hallucination). We analyze the asymptotic behavior of the Miss Rate $P(M) = P(\tau \in S_m)$. Note that S_m is only defined for answerable queries (Q_{ans}), as refusal is correct for Q_{unans} .

Table 3: Statistical Validation of Theoretical Assumptions on MuSiQue-Full.

Model	Assumption	Condition Met	Result Satisfied	Ratio
Qwen-2.5-7B-Inst	Asm. 3.3 (Correctness)	10,288	7,497	72.9%
	Asm. 3.4 (Hallucination)	14,355	12,840	89.5%
FaithRL (Ours)	Asm. 3.3 (Correctness)	15,392	14,036	91.2%
	Asm. 3.4 (Hallucination)	10,208	9,809	96.1%

B.1.1 Analysis of Objective A: Maximize Correctness

Proposition: *Optimizing solely for correctness ($J_A(\theta) = P_\theta(S_c)$) drives the Miss Rate $P(M)$ asymptotically to 0.*

Proof. Consider the reward structure: $r(\tau) = 1$ if $\tau \in S_c$, and 0 otherwise. For any answerable query $q \in \mathcal{Q}_{ans}$:

- **Action: Refuse** ($\tau \in S_m$). By definition of Miss, the answer is IDK but $A^* \neq \text{IDK}$. Thus $\tau \notin S_c$. Reward $r = 0$.
- **Action: Attempt.** The model generates a candidate answer. Even if the model is uncertain, there exists a non-zero probability that the generated answer matches A^* (either through knowledge or lucky guessing). Thus, the expected reward $E[r|\text{Attempt}] = P(\text{Correct}|\text{Attempt}) > 0$.

Since $E[r|\text{Attempt}] > E[r|\text{Refuse}] = 0$, the policy gradient consistently pushes the probability mass away from refusal (S_m) towards attempting. Asymptotically, $P(M) \rightarrow 0$. **Consequence:** This creates Over-Confidence. The model learns that "trying" is always strictly better than "missing", encouraging hallucination on difficult or unanswerable queries (where it might also get lucky). $\square$

B.1.2 Analysis of Objective B: Minimize Hallucination

Proposition: *Optimizing solely to minimize hallucination ($J_B(\theta) = -P_\theta(S_h)$) drives the Miss Rate $P(M)$ asymptotically to 1 (Degenerate Solution).*

Proof. The objective is to minimize $P(H)$, or equivalently maximize likelihood of Non-Hallucination ($S_c \cup S_m$). For any answerable query $q \in \mathcal{Q}_{ans}$:

- **Action: Refuse** ($\tau \in S_m$). By definition, S_m involves outputting "IDK". Since "IDK" is not in the Hallucination set $\mathcal{H}$, the risk of hallucination is strictly 0.

- **Action: Attempt.** Due to model imperfection, any attempt to answer carries a risk $\epsilon > 0$ of generating an incorrect answer ($\tau \in S_h$).

To strictly minimize the hallucination rate to 0, the optimal policy is to choose the action with 0 risk. Refusal (S_m) is a "safe haven" that guarantees zero penalty under Objective B. Thus, the gradient favors shifting probability mass from Attempt to Refuse. Asymptotically, the model learns to refuse all queries to achieve perfect safety, driving $P(M) \rightarrow 1$. **Consequence:** This creates Over-Conservativeness. Usefulness is sacrificed for safety. $\square$

B.1.3 Analysis of Objective C: Maximize Reasoning Faithfulness

Proposition: *Maximizing reasoning faithfulness ($J_C(\theta) = P_\theta(S_f)$) stabilizes the Miss Rate $P(M)$.*

Proof. Objective C rewards membership in $S_f = \{\tau \mid \phi(\tau) = \mathcal{E}(q)\}$. We analyze the gradient direction for answerable queries $q \in \mathcal{Q}_{ans}$ (where S_m exists):

- **Refusal (S_m):** If the model refuses, it outputs "IDK". However, for $q \in \mathcal{Q}_{ans}$, the evidence set $\mathcal{E}(q)$ consists of specific knowledge facts required to derive the answer. A refusal trajectory generally fails to utilize/cite these specific facts (or cites them but fails to derive the answer). Thus $\phi(\tau) \neq \mathcal{E}(q)$, so $\tau \notin S_f$. Reward $r = 0$.
- **Faithful Attempt (S_f):** If the model correctly uses evidence $\mathcal{E}(q)$ to derive A^* , then $\tau \in S_f$. Reward $r = 1$.

For these queries, Objective C behaves like Objective A: it penalizes Misses ($r = 0$) and rewards Faithful Attempts ($r = 1$). Thus, it exerts a downward pressure on $P(M)$, preventing the Over-Conservativeness of Objective B.

However, unlike Objective A, Objective C does not drive the model to speculative extremes. For

queries where the model cannot find evidence (e.g., due to capability limits or unanswerable nature), it cannot satisfy the condition $\phi(\tau) = \mathcal{E}(q)$. In such cases, attempting to answer often leads to S_{extra} (spurious) or S_h , which are not rewarded ($r = 0$) or penalized. By strictly tying reward to evidence presence, Objective C ensures that $P(M)$ is reduced *only* when the model is capable of faithful reasoning, thereby stabilizing the refusal strategy at a level consistent with the model’s actual capability. $\square$

B.2 Metric Details: Truthful Helpfulness Score (THS)

Due to the shortcomings of existing metrics, we employ the Truthful Helpfulness Score (THS).

Geometric Definition. To rigorously ground the definition of THS, we consider the polar characteristics of the performance plane. The angle θ of a capability vector relative to the positive x-axis, representing the Correctness Rate, inversely correlates with model quality. A smaller angle implies a lower ratio of hallucinations to correct answers. The ray $O\vec{E}_0$ thus establishes an "iso-capability" threshold. Geometrically, the perpendicular distance of a new state E_1 from this ray serves as a direct measure of capability enhancement. Since the area of $\triangle OE_0E_1$ is proportional to this altitude given a fixed base $O\vec{E}_0$, the signed area naturally quantifies the magnitude of improvement. Points to the right of the ray yield a positive area, indicating capability gain, while those to the left yield a negative area, signaling degradation.

Operationally, we calculate this signed area using the cross product in 2D space. Let $E_0 = (x_0, y_0)$ and $E_1 = (x_1, y_1)$. The THS is computed as the ratio of the achieved signed area to the maximum potential area:

$$\begin{aligned} \text{THS} &= \frac{O\vec{E}_1 \times O\vec{E}_0}{O\vec{E}^* \times O\vec{E}_0} \\ &= \frac{x_1y_0 - x_0y_1}{1 \cdot y_0 - x_0 \cdot 0} = \frac{x_1y_0 - x_0y_1}{y_0} \end{aligned} \quad (8)$$

This formulation ensures that the metric is strictly monotonic with respect to the geometric distance from the baseline.

As referenced in Figure 3, the area of the triangle $\triangle OE_0E_1$ represents the magnitude of capability improvement. The THS effectively normalizes this area, providing a metric with a theoretical upper

bound of 1, where positive values indicate improvement over the baseline.

Effectiveness Analysis. To illustrate the necessity of THS, consider the specific example involving two models: Model A with $P_c = 0.7, P_h = 0.1$ and Model B with $P_c = 0.8, P_h = 0.2$. While Model B has higher accuracy, its hallucination rate is double that of Model A. By establishing Model A as the baseline E_0 , Model B yields a THS of -60%, clearly indicating a negative impact on trustworthiness despite the accuracy gain.

We further validate the metric through controlled variable analysis to demonstrate its robustness. When the Hallucination Rate is held constant, an increase in the Correctness Rate naturally expands the signed area of the capability triangle, resulting in a higher THS. Conversely, when the Correctness Rate is fixed, a reduction in the Hallucination Rate similarly increases the signed area and improves the score. This confirms that maximizing THS is theoretically consistent with the dual goals of enhancing utility and reducing hallucination, without conflict.

For more comprehensive details, please refer to Appendix C.3 of CRaFT (Zhu et al., 2025).

B.3 Proof of Theorem 2

Proof. Let the performance of the model be represented in a 2D Cartesian plane where the x-axis denotes the Correctness Rate and the y-axis denotes the Hallucination Rate. We define the baseline capability vector as $\mathbf{v}_0 = (x_0, y_0)$ and the current policy’s performance vector as $\mathbf{v}_\pi = (P_\pi(C), P_\pi(H))$.

1. Geometric Interpretation of the Objective

The Truthful Helpfulness Score (THS) measures the capability gain as the ratio of the signed area swept by the performance vector relative to the baseline. The signed area of the triangle formed by the origin O , the baseline $\mathbf{v}_0$, and the current state $\mathbf{v}_\pi$ is given by half the magnitude of the 2D cross product:

$$\begin{aligned} \text{Area}(O, \mathbf{v}_0, \mathbf{v}_\pi) &= \frac{1}{2}(\mathbf{v}_\pi \times \mathbf{v}_0) \\ &= \frac{1}{2}(P_\pi(C) \cdot y_0 - P_\pi(H) \cdot x_0) \end{aligned} \quad (9)$$

By definition, THS is the normalized area relative to the ideal area S_{ideal} (determined by the perfect

state $\mathbf{v}^*$):

$$\text{THS}(\pi_\theta) = \frac{\text{Area}(O, \mathbf{v}_0, \mathbf{v}_\pi)}{S_{ideal}} \quad (10)$$

Crucially, S_{ideal} is determined solely by the baseline $\mathbf{v}_0$ and the task ground truth, making it a constant positive scalar with respect to the policy parameters θ .

2. Formulation of the Expected Return

The expected return $J(\theta)$ under our proposed reward scheme $r(\tau)$ is calculated by summing the probability-weighted rewards for each outcome set S_c, S_m, S_h :

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[r(\tau)] \quad (11)$$

$$= P_\pi(S_c) \cdot (+y_0) + P_\pi(S_m) \cdot 0 + P_\pi(S_h) \cdot (-x_0) \quad (12)$$

$$= y_0 P_\pi(C) - x_0 P_\pi(H) \quad (13)$$

3. Equivalence and Gradient Alignment Comparing the expressions, we observe that the expected return is exactly double the signed area:

$$J(\theta) = 2 \cdot \text{Area}(O, \mathbf{v}_0, \mathbf{v}_\pi) \quad (14)$$

Substituting the definition of THS:

$$J(\theta) = (2S_{ideal}) \cdot \text{THS}(\pi_\theta) \quad (15)$$

Since $S_{ideal} > 0$, the coefficient $C = 2S_{ideal}$ is a strictly positive constant. Calculating the gradient with respect to θ yields:

$$\nabla_\theta J(\theta) = C \cdot \nabla_\theta \text{THS}(\pi_\theta) \quad (16)$$

This establishes a strict positive proportionality (collinearity). It implies that the gradient vector of our reward function points in the exact same direction as the gradient of the Truthful Helpfulness Score, ensuring identical optimization paths. $\square$

C Methodological Clarifications

C.1 Faithfulness Modulation vs. Step-level Rewards

FAAM is not intended as a generic reuse of classical step-level rewards. Its supervision target is *faithfulness*: each step is checked for semantic support from the task-specific evidence $\mathcal{E}(q)$, so a step that is factually plausible but unsupported by the provided evidence is still treated as unfaithful. This differs from process reward models that primarily

Table 4: Comparison between static and dynamically updated geometric reward coefficients on MuSiQue-Full.

Strategy	C	H	THS
Static	82.4	7.3	51.8
Dynamic	82.5	7.5	51.1

score whether an intermediate step appears correct or likely to lead to a correct final answer.

The acquisition mechanism is also different. FAAM does not require a dense reward model or a calibrated estimate of future success for every partial trajectory. It only needs a binary semantic-consistency signal $\mathcal{V}(s_j) \in \{0, 1\}$ against the evidence, which makes the supervision easier to obtain and robust even with smaller verifiers, as shown in Table 2.

Finally, FAAM modulates the trajectory-level GRPO advantage rather than replacing the RL objective with independent per-step rewards. Positive advantages are filtered toward faithful steps, while negative advantages are concentrated on unfaithful steps. This preserves the outcome-level optimization signal from the geometric reward while preventing unsupported intermediate reasoning from being reinforced merely because the final answer happens to be correct.

C.2 Static Geometric Coefficients and THS Training Trajectory

We use static geometric reward coefficients to provide a consistent optimization anchor, optimizing performance relative to the model’s initial state. To examine whether dynamically updating these coefficients would change the final outcome, we conducted an ablation experiment using the Qwen-2.5-7B-Instruct backbone, where the coefficients were dynamically updated every 10 steps.

As shown in Table 4, both strategies yield highly comparable final performance. This confirms that our static configuration is already robust enough to reach the optimal region, while avoiding the computational overhead of continuous re-evaluation during training.

To further visualize the model’s movement in the THS plane, Figure 8 plots representative checkpoints sampled during training, with the underlying values listed in Table 5. The trajectory steadily moves toward higher correctness and lower hallucination, indicating that the optimization process follows the intended helpfulness–honesty direction.

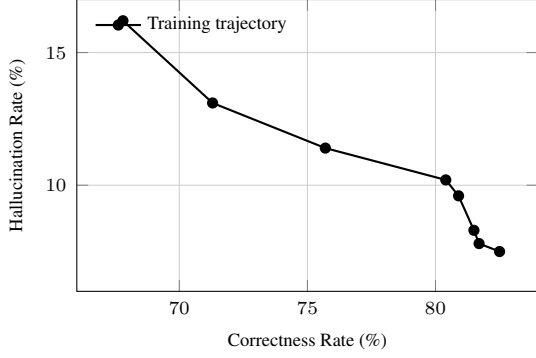


Figure 8: Representative training trajectory in the THS plane.

Table 5: Representative checkpoints used for the THS training trajectory.

Metric	1	2	3	4	5	6	7	8
C	67.8	71.3	75.7	80.4	80.9	81.5	81.7	82.5
H	16.2	13.1	11.4	10.2	9.6	8.3	7.8	7.5

C.3 Faithfulness Evaluation with Alternative Reasoning Paths

In the OOD math evaluation (Section 5.3), we instruct the verifier to base its faithfulness judgments on the provided reference solution. A natural question arises: *how does the faithfulness metric handle alternative correct reasoning paths that differ from the reference solution?*

Since we favor lightweight verifiers (e.g., Qwen-4B or the 110M HHEM) for computational efficiency, these models lack the capacity to independently judge whether a novel reasoning path is mathematically valid. Our implementation therefore anchors verification to the provided reference. This design raises a natural concern: does FaithRL learn transferable logical reasoning, or does it merely overfit to the surface form of the training references?

To answer this question, we designed a stress-test experiment. For all test questions in GSM8k and MATH500 (1,819 questions in total), we prompted an advanced LLM (GPT-4o-mini) to generate *alternative* correct reasoning paths that intentionally differ from the original reference solutions in structure, intermediate steps, and notation. We then re-evaluated the reasoning paths generated by both GRPO and FaithRL against these divergent reference paths.

As shown in Table 6, even when evaluated against entirely different reference paths, FaithRL maintains a higher faithful step ratio than GRPO

Table 6: Faithful step ratio (%) when evaluated against alternative reference solutions that intentionally diverge from the original ground-truth paths.

Method	GSM8k	MATH500	Total
GRPO	68.1	39.3	57.8
FaithRL	73.2	46.3	65.1

(+7.3 points overall). This gap indicates that the faithfulness improvements are not tied to a particular reference format: a model that merely memorized the phrasing of the original solutions would show no advantage when the reference changes. The consistent margin under divergent references suggests that FaithRL encourages the model to internalize valid reasoning patterns transferable across solution strategies, corroborating the OOD findings in Section 5.3.

D Experimental Details

D.1 Datasets

D.1.1 Dataset Overview

We evaluate our framework on three multi-hop QA benchmarks selected for their distinct complexity profiles.

HotpotQA is a large-scale multi-hop QA dataset with 112,779 Wikipedia-based question-answer pairs, constructed via crowdsourcing. Each sample includes sentence-level supporting facts to enable explainable reasoning, and all questions require multi-hop inference across multiple documents. It features diverse questions (covering entities, dates, locations, etc.) and a novel type of factoid comparison questions (including yes/no and numerical comparison tasks), without being constrained by pre-existing knowledge bases or schemas, supporting comprehensive evaluation of models’ multi-hop reasoning and explainability.

2WikiMultiHopQA is a large-scale multi-hop QA dataset with 192,606 samples, constructed by integrating unstructured text from Wikipedia and structured triples from Wikidata. Each sample includes explicit evidence in the form of structured triples to illustrate the reasoning path, ensuring all questions require multi-hop inference to answer. It covers four question types (comparison, inference, compositional, bridge comparison) generated via predefined templates and logical rules, facilitating comprehensive evaluation of models’ reasoning capabilities.

MuSiQue-Full is a challenging multi-hop QA

dataset with 25K 2–4 hop questions, constructed via a bottom–up approach by composing single-hop questions from 5 Wikipedia-based datasets. It enforces strict connected reasoning (each step relies on previous outputs) through stringent filters, minimizing train-test leakage and including hard-to-identify distractors. Featuring six reasoning graph structures, it has a 3× larger human–machine gap and lower cheatability (single-hop models drop 30 F1 points) than existing datasets, with natural question decompositions to support interpretable reasoning model development.

D.1.2 Evidence Construction

For the MuSiQue-Full dataset, each sample includes a ‘question_decomposition’ field representing prior knowledge. We leverage this structured data to construct the target evidence set $\mathcal{E}(q)$. For example, a raw decomposition might appear as:

```
[
  {"id": 42543,
   "question": "who wrote crazy little thing called love",
   "answer": "Freddie Mercury"},
  {"id": 20093,
   "question": "In what year did #1 die?",
   "answer": "1991"}
]
```

To transform these pairs into a coherent evidence set, we utilize Llama-3.3-70B-Instruct with the prompt defined in Appendix H. The model resolves references (e.g., “#1”) and synthesizes declarative statements, resulting in the final $\mathcal{E}(q)$:

```
['Freddie Mercury wrote crazy little thing called love original artist', 'Freddie Mercury died in year 1991.']
```

D.1.3 Dataset Construction for 2WikiMultihopQA-Full and HotpotQA-Full

To construct the unanswerable subsets for our Out-of-Domain test sets (2WikiMultiHopQA-Full and HotpotQA-Full), we employ a two-stage process consisting of strategic pruning and rigorous filtering.

Phase 1: Evidence Pruning Both datasets provide a supporting_facts field for each query q , formatted as a list of indices (e.g., [“Arthur’s Magazine”, 0], [“First for Women”, 0]) that point to specific sentences within the context documents. We utilize this ground truth to identify the complete evidence set $\mathcal{E}(q)$ within the retrieved context $\mathcal{K}(q)$. The initial generation of unanswerable samples involves randomly pruning documents referenced by these indices, thereby creating reasoning gaps.

Phase 2: Rigorous Quality Control To ensure the constructed samples are genuinely unanswerable due to missing logic rather than trivial irrelevance, and to prevent model shortcuts (reward hacking), we enforce three strict technical constraints during a secondary filtering stage:

1. **Context Density Maintenance:** If the number of remaining evidence documents in $\mathcal{E}(q)$ exceeds 3 after the initial pruning, we iteratively prune an additional document. This ensures that the remaining context $\mathcal{K}(q)$ maintains high semantic density and relevance to the query, preventing the model from identifying unanswerable questions simply by detecting a lack of relevant keywords (a common form of reward hacking).
2. **Hop-Sensitive Pruning:** We strictly prohibit the pruning of documents corresponding to the first hop of the reasoning chain. Instead, we randomly target intermediate or final hops. This is because first-hop entities are often explicitly mentioned in the question itself, making them easily recoverable or hallucinated. Removing later hops ensures the break in reasoning occurs deep within the logic chain.
3. **Empirical Solvability Verification:** We employ a verification step using Qwen2.5-7B-Instruct with a *Best-of-32* inference strategy. For every generated unanswerable candidate, we sample 32 reasoning paths. If the model succeeds in deriving the correct answer in *any* of the 32 attempts (indicating the reasoning gap was bypassed via hallucination or leakage), the sample is discarded. We retain only those samples where all 32 attempts fail, ensuring the target output is strictly “I don’t know”.

D.2 Implementation Details of the Baselines

Below, we provide implementation details for the baseline methods:

- **R-Tuning:** We apply R-Tuning (Zhang et al., 2024a) on the MuSiQue-Full training set using the backbone model for inference. We append “I am sure” to the model’s response for correct reasoning trajectories and “I am unsure” for incorrect ones. The backbone model is then supervised fine-tuned (SFT) on this augmented dataset using full-parameter updates with a cosine learning rate schedule for 3

epochs. During evaluation, responses containing "I am unsure" are mapped to "IDK", while answers are extracted from those marked "I am sure".

- **CRaFT:** Following [Zhu et al. \(2025\)](#), we first sample 10 responses for each training query to compute accuracy and certainty scores. We select samples in the top 10% of both metrics for Rehearsal Training to obtain a rehearsal model. This model is then employed for a two-stage data filtering process, and the final model is obtained by SFT on the filtered dataset.
- **CTSF:** We implement the Confidence-Based Thresholding strategy ([Jurayj et al., 2025](#)) using the cumulative log-probability of the generated sequence as the confidence metric. We set the refusal threshold to -15; responses falling below this value are converted to "IDK".
- **RLCR:** We calibrate the initial model via RL using the reward function $R = 1 - (cr - conf)^2$, where $cr \in \{0, 1\}$ denotes correctness (1 for Correct, 0 for Error/Miss) and $conf \in [0, 1]$ is extracted from a generated <confidence> tag ([Damani et al., 2025](#)). Post-training, we use a fixed confidence threshold of 0.5 to determine refusal.
- **TruthRL:** We train the model using PPO with the ternary reward scheme proposed by [Wei et al. \(2025\)](#), assigning rewards of +1 for correct answers, -1 for hallucinations, and 0 for other outcomes.
- **FSPO:** We fully reproduce the original implementation by directly adopting the official code of Fine-Grained Step-Level Policy Optimization ([Li and Ng, 2025](#)), with only the train batch size adjusted to 128 to enhance the stability and reproducibility of the training process.
- **KnowRL:** Following [Ren et al. \(2025\)](#), we design a composite reward function comprising format, factuality, and correctness components. We set the weighting coefficients for these three terms to a ratio of 1:1:1.

D.3 Training Details

Implementation & Hardware. We implement our framework using the `verl` library on a com-

putational node equipped with 4 NVIDIA H200 GPUs (141GB VRAM each). Models undergo full-parameter fine-tuning with a global training batch size of 128 and a validation batch size of 512. Maximum sequence lengths are set to 8192 tokens for prompts and 2048 tokens for responses.

Optimization Hyperparameters. Optimization utilizes a cosine learning rate scheduler with a peak learning rate of 1×10^{-6} , a warmup ratio of 0.1, and a minimum learning rate ratio of 0.1. For PPO training, we set the mini-batch size to 64 and the micro-batch size per GPU to 16. The KL divergence penalty is disabled (`use_kl_loss=False`), and the rollout sampling temperature is set to 1.0.

Oracle & Verification. To accurately categorize outcomes into Correct (S_c), Miss (S_m), and Hallucination (S_h) as required by our geometric reward, we implement a hierarchical evaluation protocol. We first employ rigorous string matching to detect explicit refusals (e.g., "I don't know"), classifying such instances as Miss (S_m) in answerable queries and Correctness (S_c) in unanswerable queries. For non-refusal responses, Llama-3.3-70B-Instruct serves as the oracle judge to distinguish between Correct (S_c) and Hallucination (S_h). The same model is the step-wise evidence verifier ($\mathcal{V}$) in our main FaithRL experiments, where we segment the reasoning process into steps based on sentence boundaries. We select a strong model for this verification primarily to leverage shared context caching with the outcome judge, thereby minimizing additional overhead (see Section F). Crucially, this task is distinct from reasoning distillation; it constitutes an objective check of evidence attribution rather than a complex evaluation of logical validity, ensuring the task is computationally simpler and more robust than the reasoning process itself. The verifier is deployed on a dedicated server utilizing 2 GPUs with a temperature of 0.0. We present the specific prompt templates in Appendix H.

Evidence for OOD Benchmarks. We conduct inference-only policy evaluations on GSM8k and MATH500 without evidence annotations or verifier access. Correctness, hallucination, and THS are computed from final outcomes. Only after generation, we use the ground-truth solution steps as evidence to compute the auxiliary faithful step ratio. Thus, reference solutions neither condition generation nor provide a test-time training signal. The post-hoc verification prompts are detailed in

Table 7: Verifier human alignment across task difficulty levels on MuSiQue-Full.

Task Difficulty	Proportion	Human Alignment
Simple	72.1%	83%
Medium	22.0%	94%
Hard	5.9%	80%

Appendix H.

D.4 Verifier Robustness across Task Difficulty

To systematically investigate whether the verifier can maintain accurate judgment as reasoning difficulty increases, we categorize the MuSiQue-Full dataset by the required number of reasoning hops: 2-hop, 3-hop, and 4-hop. These tiers account for 72.1%, 22.0%, and 5.9% of the data, respectively. Using the Qwen3-4B-Instruct-2507 verifier, we sampled 100 reasoning traces from each tier and measured the human alignment of its judgments.

As shown in Table 7, while verifier accuracy is naturally affected by task difficulty, the verifier demonstrates robust behavior, consistently maintaining an alignment of over 80% across all levels. An analysis of the minority of failure cases reveals two distinct error modes. For simple tasks, sparse evidence can lead to low precision: the verifier occasionally relies on parametric memory to judge factual correctness, mistakenly accepting unfaithful steps. For hard tasks, larger evidence volume can lead to low recall: the verifier occasionally struggles to match claims with the correct snippets, falsely rejecting genuinely faithful steps. To further mitigate these edge cases, future work could consider training a specialized verifier explicitly tailored for step-level faithfulness verification.

D.5 Robustness to Noisy Verification

We simulate imperfect evidence verification by independently flipping the verifier’s binary step labels during training. Table 8 reports results on MuSiQue-Full using the Qwen-2.5-7B-Instruct policy and Qwen3-4B-Instruct-2507 verifier. Performance degrades gradually as noise increases: even when 50% of labels are flipped, FaithRL retains a positive THS. This controlled experiment complements the verifier-size results, while not claiming to capture every structured error that may occur with incomplete or retrieved evidence.

Table 8: Robustness to randomly flipped verification labels on MuSiQue-Full.

Setting	C	H	THS
FaithRL (clean)	78.9	6.9	50.0
FaithRL (25% noise)	77.2	8.1	43.3
FaithRL (50% noise)	71.7	12.8	18.1

E Extended Experimental Results

E.1 Additional Results for Main Experiments

We present a more granular analysis of Multi-hop QA performance, specifically examining the model’s behavior on answerable versus unanswerable subsets of the test data in Table 10.

E.2 Training Dynamics and Step-Level Verification Baselines

We visualize the training dynamics of step-wise faithfulness in Figure 9. We track the evolution of reasoning steps for TruthRL, GRPO, and FaithRL, dividing the metrics into four categories based on the sample polarity (Positive vs. Negative) and step validity (Faithful vs. Unfaithful). As illustrated, FaithRL demonstrates superior reasoning alignment capabilities: it achieves the highest proportion of faithful steps on positive samples (Figure 9a) while successfully suppressing unfaithful reasoning steps on negative samples to the lowest level among all baselines (Figure 9d).

We further compare FaithRL with methods that explicitly leverage step-level verification signals, including FSPO and Know-RL, to provide a fairer assessment of reasoning faithfulness. The faithful step ratio results on MuSiQue-Full are reported in Table 11. FaithRL achieves the highest faithful step ratio, outperforming both outcome-oriented RL baselines and step-level verification baselines by a clear margin.

E.3 Extended Results on OOD Generalization

We also provide a comprehensive breakdown of the OOD generalization experiments on GSM8k and MATH500. Table 12 reports outcome and post-hoc process metrics for FaithRL, GRPO, and the stronger TruthRL baseline. FaithRL improves correctness over TruthRL by 12.8 points on GSM8k and 7.4 points on MATH500, while also increasing the faithful-correct step ratio by 12.6 and 7.8 points, respectively.

Table 9: Main results on Qwen2.5-7B-Instruct comparing **FaithRL** against strong baselines across three multi-hop QA benchmarks. We report **correct rate (C)** and **truthful helpfulness score (THS)** where higher is better ($\uparrow$), and **hallucination rate (H)** where lower is better ($\downarrow$). The best performance is highlighted in **bold**, and the second best is underlined.

Method	2WikiMultihopQA-Full			HotpotQA-Full			MuSiQue-Full			Average		
	C ($\uparrow$)	H ($\downarrow$)	THS ($\uparrow$)	C ($\uparrow$)	H ($\downarrow$)	THS ($\uparrow$)	C ($\uparrow$)	H ($\downarrow$)	THS ($\uparrow$)	C ($\uparrow$)	H ($\downarrow$)	THS ($\uparrow$)
<i>Qwen2.5-7B-Instruct</i>												
PROMPTING BASELINE												
Prompting	71.5	13.1	0.0	72.7	19.3	0.0	59.2	16.3	0.0	67.8	16.2	0.0
REFUSAL-AWARE SFT												
R-Tuning	59.2	21.9	-60.3	64.8	15.2	7.5	65.5	13.8	27.2	63.2	16.9	-7.5
CRaFT	51.3	10.9	-8.2	64.5	10.3	25.7	58.2	5.3	43.5	58.0	<u>8.8</u>	21.2
CONFIDENCE-BASED ABSTENTION												
CTSF	55.7	24.1	-75.8	63.3	11.4	20.4	53.8	7.5	33.0	57.6	14.3	-2.2
RLCR	64.9	27.0	-82.5	67.1	29.7	-44.8	52.0	29.1	-53.7	61.3	28.5	-58.0
GRPO + CTSF	64.8	29.2	-94.6	76.1	15.3	18.5	63.5	13.2	26.9	68.1	19.2	-12.3
RLVR METHODS												
GRPO	<u>84.9</u>	13.1	13.4	84.2	14.8	28.4	79.4	15.3	37.0	82.8	14.4	22.5
TruthRL	83.8	<u>9.8</u>	<u>30.3</u>	<u>85.1</u>	<u>10.2</u>	<u>46.7</u>	<u>77.7</u>	8.0	<u>55.5</u>	82.2	9.3	<u>43.3</u>
FSPO	72.4	14.9	-8.9	67.2	26.1	-31.1	58.6	13.9	8.1	66.0	18.3	-10.6
KnowRL	73.8	23.9	-5.7	79.7	18.9	8.5	73.8	17.6	9.9	75.8	20.1	-8.3
FaithRL (Ours)	85.4	7.9	42.3	86.1	8.0	56.0	75.7	<u>5.9</u>	59.3	<u>82.4</u>	7.3	51.8

Table 10: Fine-grained results comparing **FaithRL** against strong baselines across three multi-hop QA benchmarks. We report **Correct Rate (C)** and **Truthful Helpfulness Score (THS)** where higher is better ($\uparrow$), and **Hallucination Rate (H)** where lower is better ($\downarrow$). The performance is reported separately for Answerable (Ans.) and Unanswerable (Unans.) subsets. The best performance is highlighted in **bold**, and the second best is underlined.

Method	2WikiMultihopQA-Full						HotpotQA-Full						MuSiQue-Full						Average		
	Ans.			Unans.			Ans.			Unans.			Ans.			Unans.			A	H	T
	C	H	T	C	H	T	C	H	T	C	H	T	C	H	T	C	H	T			
<i>Llama3.1-8B-Instruct</i>																					
PROMPTING BASELINE																					
Prompting	56.6	30.4	0.0	81.7	18.3	0.0	70.5	24.6	0.0	56.8	43.2	0.0	41.6	32.3	0.0	66.6	33.4	0.0	62.3	30.4	0.0
REFUSAL-AWARE SFT																					
R-Tuning	18.7	20.5	-19.4	95.1	4.9	73.2	34.2	15.8	-11.1	94.1	5.9	86.3	35.5	21.0	8.5	96.3	3.7	88.9	62.3	12.0	37.7
CRaFT	9.7	18.2	-24.2	88.5	11.5	37.2	34.5	11.4	1.8	<u>91.5</u>	<u>8.5</u>	<u>80.3</u>	22.2	8.4	11.4	<u>96.1</u>	<u>3.9</u>	<u>88.3</u>	57.1	<u>10.4</u>	35.8
CONFIDENCE-BASED ABSTENTION																					
CTSF	37.4	19.1	1.8	89.5	10.5	42.6	43.4	11.4	10.7	89.8	10.2	76.4	20.6	10.9	6.6	93.0	7.0	79.0	62.3	11.5	38.7
RLCR	64.4	28.3	11.7	56.7	43.3	-136	72.3	26.3	-3.1	46.3	53.7	-24.3	45.1	39.0	-5.1	56.1	43.9	-31.4	56.8	39.1	-23.3
GRPO + CTSF	67.6	27.0	17.3	33.6	66.4	-262	77.0	15.5	32.5	58.9	41.1	4.9	61.9	17.1	39.9	55.7	44.3	-32.6	59.1	35.2	-13.0
RLVR METHODS																					
GRPO	83.5	13.3	<u>58.7</u>	88.8	11.2	38.8	88.1	<u>9.8</u>	<u>60.0</u>	64.1	35.9	16.9	68.9	16.7	47.4	90.5	9.5	71.6	<u>80.7</u>	16.1	47.7
TruthRL	81.8	12.0	59.5	88.8	11.2	38.8	84.5	7.5	63.0	71.6	28.4	34.3	70.0	<u>9.5</u>	57.8	90.2	9.8	70.7	81.2	13.1	<u>54.4</u>
FSPO	62.8	10.8	42.7	92.5	7.5	59.0	73.1	11.3	40.7	83.4	16.6	61.6	36.1	15.1	16.7	90.3	9.7	71.0	73.0	11.8	48.8
KnowRL	<u>82.6</u>	13.8	56.9	74.9	25.1	-37.2	<u>87.1</u>	10.0	58.4	76.7	23.3	46.1	67.4	17.0	45.5	83.2	16.8	49.7	78.7	17.7	42.4
FaithRL (Ours)	81.7	<u>11.6</u>	60.1	<u>93.3</u>	6.7	<u>63.4</u>	83.9	<u>9.8</u>	55.8	87.0	13.0	69.9	65.3	12.4	<u>49.3</u>	94.7	5.3	84.1	84.3	9.8	64.2
<i>Qwen2.5-7B-Instruct</i>																					
PROMPTING BASELINE																					
Prompting	50.4	18.8	0.0	<u>92.6</u>	<u>7.4</u>	0.0	66.2	17.7	0.0	79.2	20.8	0.0	31.2	19.7	0.0	87.2	12.8	0.0	67.8	16.2	0.0
REFUSAL-AWARE SFT																					
R-Tuning	26.6	35.4	-68.3	91.7	8.3	-12.2	36.8	23.2	-50.0	92.8	7.2	65.4	34.9	23.6	-2.5	96.1	3.9	69.5	63.2	16.9	-7.5
CRaFT	9.8	14.5	-29.1	92.7	7.3	1.4	38.0	11.5	-5.0	90.9	9.1	56.3	19.8	<u>7.2</u>	8.4	<u>96.6</u>	<u>3.4</u>	<u>73.4</u>	58.0	<u>8.8</u>	21.2
CONFIDENCE-BASED ABSTENTION																					
CTSF	43.8	15.7	1.71	67.5	32.5	-339	39.3	10.0	1.89	87.2	12.8	38.5	15.4	7.1	4.16	92.1	7.9	38.3	57.6	14.3	-2.2
RLCR	60.5	23.2	-1.7	69.3	30.7	-315	71.9	21.6	-8.9	62.3	37.7	-81.3	33.6	28.5	-11.5	70.4	29.6	-131	61.3	28.5	-58.0
GRPO + CTSF	71.0	16.9	25.7	58.5	41.5	-460	71.9	10.7	31.9	80.2	19.8	4.81	42.6	10.7	25.7	84.3	15.7	-22.7	68.1	19.2	-12.3
RLVR METHODS																					
GRPO	82.0	14.1	44.2	87.9	12.1	-63.5	86.8	11.1	45.3	81.6	18.4	11.5	65.9	23.5	28.7	92.9	7.1	44.5	82.8	14.4	22.5
TruthRL	<u>78.1</u>	<u>8.9</u>	<u>54.2</u>	89.4	10.6	-43.2	<u>82.1</u>	<u>8.4</u>	50.7	88.1	11.9	42.8	<u>59.6</u>	11.7	41.1	95.7	4.3	66.4	82.2	9.3	<u>43.3</u>
FSPO	56.4	18.1	7.9	88.3	11.7	-58.1	70.4	16.2	9.8	64.0	36.0	-73.1	31.1	13.7	9.4	86.0	14.0	-9.4	66.0	18.3	-10.6
KnowRL	70.4	24.8	3.9	77.1	22.9	-209	82.0	15.1	25.5	77.4	22.6	-8.7	56.5	26.3	14.8	91.1	8.9	30.5	75.8	20.1	-8.3
FaithRL (Ours)	78.0	8.6	54.9	92.7	7.3	1.4	79.8	8.3	<u>48.8</u>	<u>92.4</u>	<u>7.6</u>	<u>63.5</u>	54.2	9.1	<u>39.8</u>	97.2	2.8	78.1	<u>82.4</u>	7.3	51.8

E.4 Extended Results on Hyperparameter Sensitivity

Table 13 reports the per-benchmark results of the hyperparameter sensitivity analysis for α on Qwen-2.5-7B-Instruct, showing how the trade-off between standard rewards and faithfulness constraints affects performance as α varies from 0 to 1.

E.5 Extended Results on Component Analysis

We utilize the Qwen-2.5-7B-Instruct model as the backbone for these ablation studies. Table 14 details the performance metrics across all three multi-hop QA benchmarks, illustrating the contribution of each module to the overall performance.

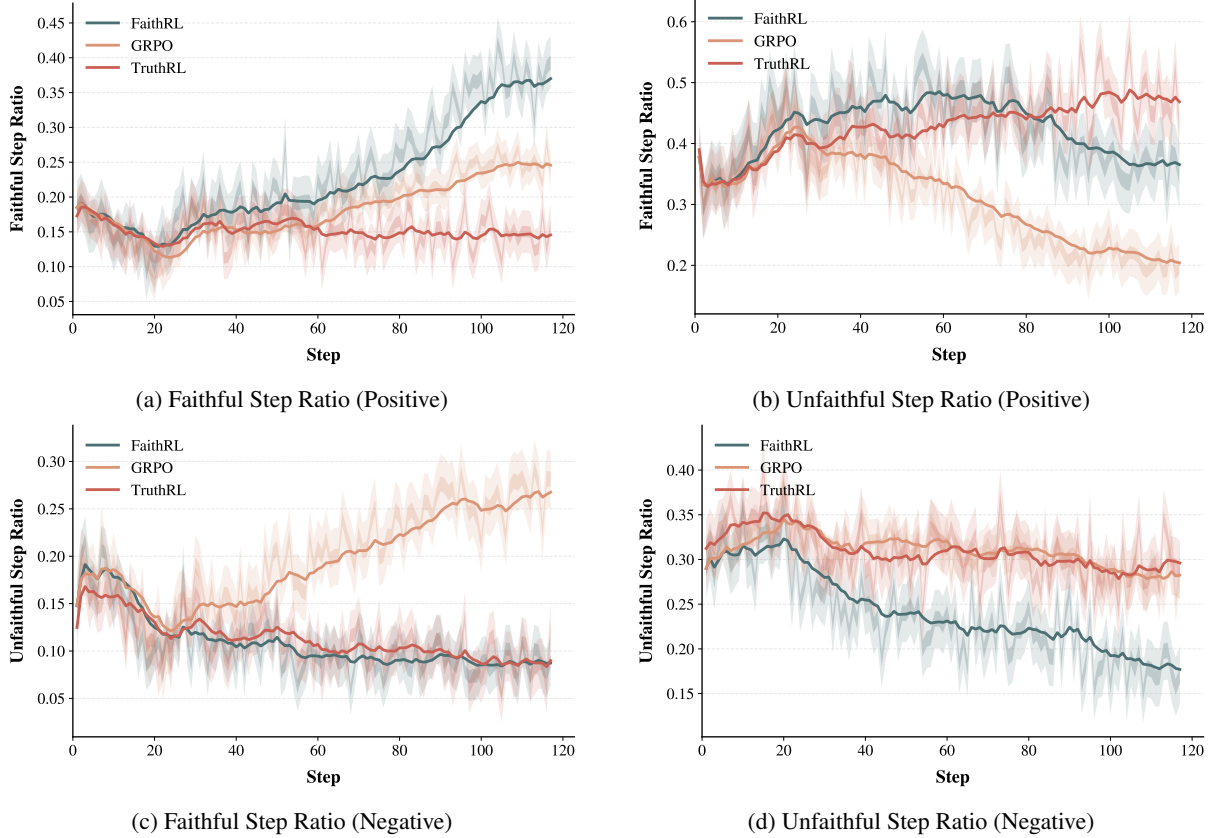


Figure 9: Analysis of Faithful and Unfaithful Step Ratios on positive and negative samples. TruthRL, GRPO, and FaithRL are compared across different steps.

Table 11: Faithful step ratio (%) on MuSiQue-Full.

Method	MuSiQue-Full
GRPO	45.9
TruthRL	61.2
FSPO	57.0
Know-RL	53.4
FaithRL	74.6

F Computational Cost

Experimental Setup Details. To ensure a rigorous and fair evaluation, we tracked resource consumption across the full training pipeline. For **FLOPs**, we recorded the ‘flops/total’, representing the cumulative floating-point operations since the start of training. This metric encompasses all major stages: actor/critic updates, log-probability/value forward passes, vLLM rollout generation, and server-side LLM-judge computations (estimated via API usage tokens). For **GPU Hours**, we calculated the total usage of the 4 NVIDIA H200 GPUs dedicated to model training and gradient updates. Additionally, to account for the overhead of the external reward model, we included the consump-

tion of the 2 H200 GPUs hosting the LLM server, scaled by their average Streaming Multiprocessor (SM) Utilization. This approach ensures that the reported cost reflects the actual hardware burden rather than just wall-clock time.

Extended Results on GPU Hours. We provide a detailed breakdown of the computational cost comparison between GRPO and FaithRL. The total GPU hours are composed of two parts: the time spent on model parameter updates (using 4 GPUs) and the equivalent time cost for the LLM Judge server.

Table 15 presents the detailed calculation for the total GPU hours consumed by both methods.

The "Update Time" refers to the wall-clock time for training on 4 GPUs. The "Judge Server Cost" is calculated by converting the active judge server usage (percentage of training time $\times$ number of judge GPUs, here assumed to be 2) into equivalent GPU hours. FaithRL incurs slightly higher costs due to the additional verification steps, but the overall increase remains modest.

Table 12: Detailed Performance and Faithfulness Analysis on OOD Datasets. The table presents outcome metrics (Correct Rate, Miss Rate, Hallucination Rate) and process-level metrics (Faithfulness and Unfaithfulness). Values in parentheses indicate the raw counts (numerator/denominator).

Dataset	Method	Outcome Metrics			Process Validity		
		Correctness	Miss	Hallucination	Faith. (Correct)	Faith. (Other)	Unfaithful
GSM8k	FaithRL	86.9% (1146/1319)	0.9% (12/1319)	12.2% (161/1319)	68.3% (3303/4838)	5.8% (281/4838)	25.9% (1254/4838)
	GRPO	72.3% (954/1319)	1.2% (16/1319)	26.5% (349/1319)	52.3% (2994/5728)	15.2% (873/5728)	32.5% (1861/5728)
	TruthRL	74.1% —	6.1% —	19.8% —	55.7% —	14.2% —	30.1% —
MATH500	FaithRL	53.6% (268/500)	1.6% (8/500)	44.8% (224/500)	32.6% (683/2098)	19.8% (415/2098)	47.7% (1000/2098)
	GRPO	46.6% (233/500)	6.2% (31/500)	47.2% (236/500)	24.0% (765/3191)	21.2% (676/3191)	54.8% (1750/3191)
	TruthRL	46.2% —	8.0% —	45.8% —	24.8% —	23.6% —	51.6% —

Table 13: Hyperparameter sensitivity analysis of α on Qwen2.5-7B-Instruct. α balances the trade-off between standard reinforcement learning rewards and our proposed faithfulness constraints. Note that $\alpha = 1.0$ is equivalent to the vanilla RL method with our geometric reward. The chosen setting for our main experiments is highlighted in gray.

Setting	2WikiMultihopQA-Full				HotpotQA-Full				MuSiQue-Full				Average	
	Ans.		Unans.		Ans.		Unans.		Ans.		Unans.		A	H
	A	H	A	H	A	H	A	H	A	H	A	H		
Prompting	50.4	18.8	92.6	7.4	66.2	17.7	79.2	20.8	31.2	19.7	87.2	12.8	67.8	16.2
$\alpha = 0$ (Ours)	78.0	8.6	92.7	7.3	79.8	8.3	92.4	7.6	54.2	9.1	97.2	2.8	82.4	7.3
$\alpha = 0.25$	76.8	8.9	90.5	9.5	81.6	8.1	90.3	9.7	51.0	9.5	96.7	3.3	81.1	8.2
$\alpha = 0.5$	74.5	7.7	94.3	5.7	78.2	7.8	89.6	10.4	50.0	9.3	95.8	4.2	80.4	7.5
$\alpha = 0.75$	74.2	8.0	93.3	6.7	80.5	7.5	88.5	11.5	44.0	9.6	96.4	3.6	79.5	7.8
$\alpha = 1.0$ (Vanilla)	73.2	6.8	93.2	6.8	82.0	7.5	85.6	14.4	50.7	12.2	94.5	5.5	79.8	8.8

Table 14: Detailed ablation results on component effectiveness.

Setting	2WikiMultihopQA-Full				HotpotQA-Full				MuSiQue-Full				Average	
	Ans.		Unans.		Ans.		Unans.		Ans.		Unans.		A	H
	A	H	A	H	A	H	A	H	A	H	A	H		
Prompting	50.4	18.8	92.6	7.4	66.2	17.7	79.2	20.8	31.2	19.7	87.2	12.8	67.8	16.2
GRPO	82.0	14.1	87.9	12.1	86.8	11.1	81.6	18.4	65.9	23.5	92.9	7.1	82.8	14.4
+ $\mathcal{R}_{geo}$	73.2	6.8	93.2	6.8	82.0	7.5	85.6	14.4	50.7	12.2	94.5	5.5	79.8	8.8
+FAAM	78.2	8.5	88.1	11.9	83.2	7.1	73.1	26.9	52.4	9.5	91.7	8.3	77.8	12.0
FaithRL	78.0	8.6	92.7	7.3	79.8	8.3	92.4	7.6	54.2	9.1	97.2	2.8	82.4	7.3

Analysis of efficiency. Three implementation choices keep the actual wall-clock overhead below 15%:

1. **Batched multi-step faithfulness verification:** The verifier first segments each response by its numbered reasoning steps and then places all segments from that response into a single prompt. The judge returns a JSON list of 0/1 labels with the same length and order as the segments. Thus, the request-level overhead grows by approximately one step-verification request per response, instead of N separate

requests.

2. **KV-cache reuse (prefix caching):** Judge prompts are structured as a static prefix followed by a short dynamic suffix. The server is launched with `-enable-prefix-caching`, allowing repeated instruction, evidence, and context prefixes to reuse cached KV states across rollouts and fallback step-verification calls. This avoids repeatedly paying the full prefill cost for largely identical prefixes, so the marginal cost of additional verification is dom-

Table 15: Breakdown of total GPU hours calculation.

Method	Update Time (h)	Judge Server Cost (h)	Total GPU Hours	Increase
GRPO	$4 \times 22.4 = 89.6$	$22.4 \times 22.37\% \times 2 \approx 10.0$	99.6	-
FaithRL	$4 \times 24.8 = 99.2$	$24.8 \times 28.17\% \times 2 \approx 14.0$	113.2	+13.7%

inated by the small suffix and short decoding output.

3. **Concurrent verification processing:** Outcome and step-verification jobs are submitted concurrently through a thread pool controlled by LLM_JUDGE_MAX_WORKERS. On the server side, vLLM is configured with MAX_NUM_SEQS=64, enabling request batching and concurrent service. Empirical monitoring reveals that the judge-server GPU utilization remains relatively low (approximately 20%–30%) for both GRPO and FaithRL, so the extra lightweight verification requests do not translate into a linear increase in latency.

In conclusion, due to batched step judgments, efficient prefix caching, and the server’s capacity to handle concurrent requests without saturation, the additional computational overhead of FaithRL remains marginal.

G Case Study

To provide concrete insights into the behavioral differences between FaithRL and GRPO, we present two representative cases from the HotpotQA-Full dataset. These examples illustrate how our method enhances both reasoning capability and faithfulness compared to outcome-driven baselines.

G.1 Analysis of Answerable Scenarios

In the answerable case (Table 16), the model is required to perform multi-hop reasoning: identifying a specific compilation album, finding a track on it, and verifying its sampling source.

- **FaithRL (Ours):** Our model successfully synthesizes information from multiple documents. It correctly identifies that “Fantastic Voyage: The Greatest Hits” contains tracks from “Gangsta’s Paradise”, and links this to the sampling information of Stevie Wonder. The reasoning chain is logically sound and leads to the correct answer.

- **GRPO (Baseline):** The baseline model fails to make the necessary logical connection. Although it retrieves relevant information about “Gangsta’s Paradise”, it fails to link the compilation album context correctly, leading to an unnecessary refusal (“I don’t know”). This highlights the issue of over-conservativeness or reasoning failure in standard RL approaches when faced with complex evidence integration.

G.2 Analysis of Unanswerable Scenarios

In the unanswerable case (Table 17), the model must identify a specific location of an uprising. While the general context (Haitian Revolution) is present, the specific location “Saint-Domingue” is missing from the provided evidence.

- **FaithRL (Ours):** The model correctly identifies the event and the general context (Haiti) but notes that “the documents do not provide a specific location within Haiti”. It therefore refuses to answer, adhering to the provided knowledge rather than filling the gap with parametric memory.
- **GRPO (Baseline):** The baseline model succumbs to hallucination or internal knowledge bias. It attempts to answer with “Haiti”, which is technically the resulting state but not the specific answer required by the ground truth context (which was pruned). This illustrates the tendency of standard RL to “guess” based on partial associations, violating the faithfulness constraint.

G.3 Reasoning Faithfulness and Step-wise Contribution

Beyond correctness, we observe a significant disparity in reasoning efficiency. As demonstrated in the examples, FaithRL typically resolves complex multi-hop queries within approximately 3 concise steps. In contrast, the GRPO-trained model often requires more than double the number of steps to arrive at a conclusion. Crucially, regarding reasoning faithfulness, every step generated by FaithRL

validly contributes to the final derivation, achieving a 100% valid step ratio in these cases. Conversely, the GRPO baseline exhibits a high proportion of redundant or irrelevant reasoning steps, diluting the logical density and indicating a reliance on search-based trial-and-error rather than systematic deduction.

H Prompt Template

We provide the comprehensive prompt templates utilized throughout our framework: Table 18 details the transformation of question decompositions into declarative evidence; Table 19 defines the instruction for the reasoning policy; Table 20 outlines the batched step-level faithfulness verification protocol used during training, while Table 21 outlines the single-segment protocol used for OOD faithfulness evaluation; Table 22 specifies the criteria for judging final answer correctness .

Question: What song by the American rapper Coolio is on his greatest hits album "Fantastic Voyage" and samples from a Stevie Wonder song?

FaithRL (Ours) Output:

1. First, identify the greatest hits album by Coolio that is mentioned. From Document 'Fantastic Voyage: The Greatest Hits', we learn "Fantastic Voyage: The Greatest Hits is a compilation album by rapper Coolio, released in 2001. It includes the track 'Aw Here it Goes' he contributed as a main theme to TV series 'Kenan & Kel'. Other songs originally appear on his first three albums, 'It Takes a Thief', 'Gangsta's Paradise' and 'My Soul'. " This sentence mentions that the album is "Fantastic Voyage" and that it includes tracks from Coolio's other albums. However, it does not directly state which song samples a Stevie Wonder song.

2. Next, identify a song by Coolio that samples a Stevie Wonder song. From Document 'Gangsta's Paradise', we learn "'Gangsta's Paradise" is a song by American rapper Coolio, featuring singer L.V. The song was released on Coolio's album of the same name, as well as the soundtrack for the 1995 film "Dangerous Minds". It samples the chorus and instrumentation of Stevie Wonder's 1976 song "Pastime Paradise". " This sentence states that "Gangsta's Paradise" by Coolio samples a Stevie Wonder song. Therefore, "Gangsta's Paradise" is the song by Coolio that samples a Stevie Wonder song.

3. Verify if "Gangsta's Paradise" is on the greatest hits album "Fantastic Voyage". From Document 'Fantastic Voyage: The Greatest Hits', we learn "Fantastic Voyage: The Greatest Hits is a compilation album by rapper Coolio, released in 2001. It includes the track 'Aw Here it Goes' he contributed as a main theme to TV series 'Kenan & Kel'. Other songs originally appear on his first three albums, 'It Takes a Thief', 'Gangsta's Paradise' and 'My Soul'. " This sentence mentions that "Gangsta's Paradise" is one of the songs originally appearing on his first three albums, implying it is on the greatest hits album "Fantastic Voyage".

Answer: Gangsta's Paradise (Reward: y_0)

GRPO (Baseline) Output:

1. First, we need to identify a song by Coolio that is on his greatest hits album "Fantastic Voyage" and samples from a Stevie Wonder song. From the document "Fantastic Voyage: The Greatest Hits", we learn that this album by Coolio includes the track "Aw Here it Goes" which he contributed as a main theme to the TV series "Kenan & Kel".

2. Next, we need to find out if this song or any other song by Coolio samples from a Stevie Wonder song. The document "Gangsta's Paradise" states that "Gangsta's Paradise" by Coolio samples the chorus and instrumentation of Stevie Wonder's 1976 song "Pastime Paradise".

3. Given that "Gangsta's Paradise" is a Coolio song that samples a Stevie Wonder song, and it is also mentioned that "Gangsta's Paradise" is featured on the album of the same name, which is not "Fantastic Voyage", we need to find another Coolio song on "Fantastic Voyage" that samples a Stevie Wonder song.

4. The document "Fantastic Voyage (Coolio song)" tells us that "Fantastic Voyage" by Coolio samples "Fantastic Voyage" by Lakeside, and it is on his debut album "It Takes a Thief".

5. The document "Curtain Falls" mentions that the background music of "Curtain Falls" is almost identical to that of Coolio's 1995 hit, "Gangsta's Paradise", as they both share usage of a sample of Stevie Wonder's "Pastime Paradise".

6. Given the information, it is not explicitly stated that a song on "Fantastic Voyage" samples a Stevie Wonder song, but "Gangsta's Paradise" does.

Answer: I don't know (Reward: 0)

Table 16: Comparison on an Answerable Case. FaithRL correctly bridges the evidence, whereas GRPO fails to link the compilation album to the original track.

Question: Where did Henri Christophe and other slaves hold an uprising from 1791 to 1804 that led to the founding of a state which was both free from slavery and ruled by non-whites and former captives?

FaithRL (Ours) Output:

1. First, identify the state founded by the uprising. From Document 'State of Haiti', we learn "The State of Haiti (French: État d'Haïti, Haitian: Leta an Ayiti) was the name of the state in northern Haiti. It was created on October 17, 1806 following the overthrow of the Empire of Haiti following the assassination of Emperor Jacques I. The northern State of Haiti was ruled by Henri Christophe originally as 'Provisional Chief of the Haitian Government' from October 17, 1806 until February 17, 1807 when he became 'President of the State of Haiti'. The 1807 constitution for the State of Haiti made the post of president a position for life with the president having the power to appoint his successor. On March 28, 1811 President Henri was proclaimed King Henry I, thereby dissolving the State of Haiti and creating the Kingdom of Haiti." This document mentions the State of Haiti but does not directly state it was free from slavery and ruled by non-whites and former captives. However, it does refer to the Haitian Revolution, which is mentioned in other documents. Document 'Jean-Baptiste Sans Souci' states "Jean-Baptiste Sans-Souci was a leader of rebel slaves during the Haitian Revolution. He was assassinated by rival black rebel leader, Henri Christophe, in 1803, shortly before Haiti won its independence." This suggests the Haitian Revolution led to the founding of Haiti, which was free from slavery and ruled by non-whites and former captives. Document 'Henri Christophe' confirms Henri Christophe was a "key leader in the Haitian Revolution, which succeeded in gaining independence from France in 1804." Document 'State of Haiti' also mentions the Haitian Revolution in relation to the founding of the State of Haiti. Therefore, the Haitian Revolution is the uprising referred to in the question.

2. The question asks where Henri Christophe and other slaves held an uprising. From Document 'Jean-Baptiste Sans Souci', we learn "Jean-Baptiste Sans-Souci was a leader of rebel slaves during the Haitian Revolution. He was assassinated by rival black rebel leader, Henri Christophe, in 1803, shortly before Haiti won its independence." This suggests the uprising took place in Haiti. However, the document does not specify a particular location within Haiti. Given the information, it is clear the uprising occurred in Haiti, but the exact location is not specified.

3. The question asks for the location of the uprising, but the documents do not provide a specific location within Haiti. Therefore, I do not have enough information to pinpoint the exact location of the uprising.

Answer: I don't know (Reward: y_0)

GRPO (Baseline) Output:

1. First, we need to identify the region where Henri Christophe and other slaves held an uprising from 1791 to 1804. From the document "Henri Christophe", we learn that Henri Christophe was a key leader during the Haitian slave rebellion (1791–1804).

2. The Haitian Revolution is the uprising we are looking for. Next, we need to find out if the state that was founded was free from slavery and ruled by non-whites and former captives.

3. The document "Jean-Baptiste Sans Souci" mentions that Jean-Baptiste Sans-Souci was a leader of rebel slaves during the Haitian Revolution, and that Haiti won its independence from France in 1804.

4. The document "State of Haiti" provides more information about the State of Haiti, which was created in northern Haiti after the overthrow of the Empire of Haiti following the assassination of Emperor Jacques I. It was ruled by Henri Christophe.

5. The document "Kingdom of Haiti" confirms that Henri Christophe established the Kingdom of Haiti in 1811, which was in the north of Haiti.

6. Given that Henri Christophe was a former slave and a key leader in the Haitian Revolution, and that the documents mention Haiti gaining independence and becoming a state ruled by non-whites (former slaves), we can conclude that the state in question is Haiti.

Answer: Haiti (Reward: $-x_0$)

Table 17: Comparison on an Unanswerable Case. FaithRL recognizes the missing specific evidence and refuses, while GRPO hallucinates/guesses a related entity.

Table 18: Prompt for converting question decomposition into declarative evidence statements.

Evidence Construction Prompt

System Input:

You are an expert at turning subquestion-answer pairs into one clear, natural English statement.

Instructions: 1. Produce ONE grammatically correct declarative sentence using the given subquestion and its answer. 2. Preserve the semantic relationship between entities. 3. If the subquestion contains references like #1, #2, etc., resolve them using the list of prior answers (1-indexed). 4. Return ONLY the resulting sentence, nothing else.

Examples: Q: who wrote crazy little thing called love original artist | A: Freddie Mercury → Freddie Mercury wrote Crazy Little Thing Called Love.

Q: In what year did #1 die? | A: 1991 | prior: ['Freddie Mercury'] → Freddie Mercury died in 1991.

User Input:

Convert the following pair into a single statement.

Subquestion: {question}

Answer: {answer}

Prior answers (for # references): {prior_answers}

Table 19: Prompt used for generating reasoning chains (Reasoning Policy).

Reasoning Prompt
System Prompt: You are a helpful assistant. You are given a Question and References. The references may or may not help answer the question. Your task is to answer the question based on factual information in the references. If, after examining and reasoning over the References, you are uncertain or don't know the answer, output I don't know as specified below. CRITICAL - You MUST follow this EXACT format: <think> 1. [First complete reasoning phase] 2. [Second complete reasoning phase] 3. [Third complete reasoning phase] ... </think> <answer>Your final answer</answer> Rules (STRICTLY ENFORCED): 1. Put reasoning in <think></think> tags 2. Use numbered steps (1., 2., 3., ...) in your <think> section for clear structured reasoning 3. Each step must represent a complete logical inference. Do NOT split a single thought or evidence extraction into multiple tiny steps. Combine retrieving a document, quoting it, and deducing a fact into ONE single numbered step. 4. NEVER start with anything other than <think> or <answer> 5. The <answer> tag MUST contain your final answer or "I don't know" Remember: Any response without proper <answer></answer> tags is INCORRECT. User Prompt: References: {documents_context} Question: {question}

Table 20: Prompt for the LLM-based Faithfulness Verifier $\mathcal{V}$ (batched step-level training).

Batched Faithfulness Verification Prompt
System Input: You are a strict judge. Output JSON only. User Input: You are a strict reasoning consistency judge. Task: For EACH segment below, output 1 if it is FULLY SUPPORTED by the evidences; otherwise output 0. Rules: 1) Output ONLY valid JSON: a list of 0/1 with the same length and order as the segments. 2) Be strict: if a segment is vague, adds unsupported facts, or is not grounded in evidences, output 0. 3) Do not output any explanation or extra text. Evidences: {evd_text} Segments: 1. {segment_1} 2. {segment_2} ... K. {segment_K} Output JSON list:

Table 21: Prompt for the LLM-based Faithfulness Verifier $\mathcal{V}$ (OOD tasks).

Faithfulness Verifier Prompt
Input: You are a strict reasoning consistency judge. Decide if the reasoning segment is FULLY SUPPORTED by the provided evidences. Rules: 1) Output only one digit: 1 if the segment contains meaningful reasoning AND is strictly supported by the evidences; 0 otherwise. 2) Give 0 if the segment adds NO new information, is just a plan/re-statement, or lacks specific details (e.g., 'We should review the list...'). 3) Give 1 ONLY if the segment's key assertion semantically matches or is directly inferred from an evidence. 4) Base the decision strictly on the evidences; ignore world knowledge. 5) Do not provide explanations. User Input: Evidences: {evidence} Reasoning Segment: {reasoning_segment} Output (only 0 or 1):

Table 22: Prompt for the Outcome Correctness Judge.

Outcome Judge Prompt
Input: Assume you are a human expert in grading predictions given by a model. You are given a question and a model prediction. Judge if the prediction matches the ground truth answer by following these steps: 1: Take it as granted that the Ground Truth is always correct. 2: If the Prediction exactly matches the Ground Truth, "score" is 1. 3: If the Ground Truth is a number, "score" is 1 if and only if the Prediction gives a number that almost exactly matches the ground truth. 4: If the Prediction is self-contradictory, "score" must be -1. 5: If the prediction is not answering the question, "score" must be -1. 6: If the prediction is a concise and correct summary of the ground truth, "score" is 1. 7: If ground truth contains a set of items, prediction must contain exactly same items for the score to be 1. 8: Otherwise, "score" is -1. Output ONLY one digit: -1 or 1. No text, no explanation. You should make the judgment based on provided examples. Examples: Question: When did the director of film Lord Richard In The Pantry die? Ground Truth: 7 January 1984 Prediction: January 7, 1984 Output: 1 Question: Who is older, Charles Badham or Médéric De Vasselot De Régné? Ground Truth: Charles Badham Prediction: Médéric De Vasselot De Régné Output: -1 User Input: Question: {question} Ground Truth: {ground_truth_answer} Prediction: {predicted_answer} Output: