RADO: Reasoning Audit-Driven Optimization for Rigorous Reasoning in High-Stakes Domains

Zhijie Tan, Xu Chu, Guanyu Wang, Ziyu Li, Weiping Li*, Tong Mo,

School of Software and Microelectronics, Peking University, Beijing, China
{besttangent, chuxu, wgy2023, liziyu_lizy}@stu.pku.edu.cn,
{wpli, motong}@ss.pku.edu.cn

*Corresponding author.

Abstract

High-stakes domains such as finance, law, and biomedicine demand both accurate results and rigorous reasoning. Current reinforcement learning paradigms primarily rely on outcome-based rewards, often overlooking latent logical fallacies in intermediate steps. Leveraging the cognitive asymmetry where falsifying local errors is more efficient than generating global correctness, we propose **RADO** (**R**easoning **A**udit-**D**riven **O**ptimization). RADO introduces a specialized audit model augmented with external tools to identify local logical ruptures and calibrate reward signals. By integrating Direct Preference Optimization (DPO) with Group Relative Policy Optimization (GRPO), our framework enables explicit supervision over reasoning paths. Experimental results demonstrate that RADO consistently improves final accuracy while significantly enhancing logical rigor in high-stakes domains.

1 Introduction

Large Language Models (LLMs) have demonstrated significant potential in complex logical reasoning, particularly in high-stakes domains such as financial decision-making (Liu et al., 2025b; Koa et al., 2024), legal (Shi et al., 2025; Cheong et al., 2024), and biomedicine (Labrak et al., 2024a; Wang et al., 2023). In these fields, the accuracy of decisions and the interpretability of reasoning processes are not merely performance metrics but are core requirements for safety and compliance (Wang et al., 2025; Chu et al., 2025; Deng et al., 2024).

Currently, the dominant paradigm for scaling reasoning capabilities involves Reinforcement Learning (RL) guided by outcome-based rewards (i.e., the correctness of the final answer) (Zheng et al., 2025a; Yu et al., 2025; Shao et al., 2024). However, this approach faces fundamental limitations. A correct final answer is an unreliable proxy for reasoning integrity; models may arrive at the right conclusion through flawed logic or sheer coincidence—a phenomenon often termed "getting the right answer for the wrong reasons." (Shao et al., 2025; Lightman et al., 2023). Figure 1 provides a comparative example of reasoning quality. Furthermore, without explicit supervision of intermediate reasoning paths, LLMs struggle to avoid latent logical fallacies (Shojaee et al., 2025; Uesato et al., 2022). In high-stakes domains, a rigorous step-by-step derivation is arguably more vital than a direct answer.

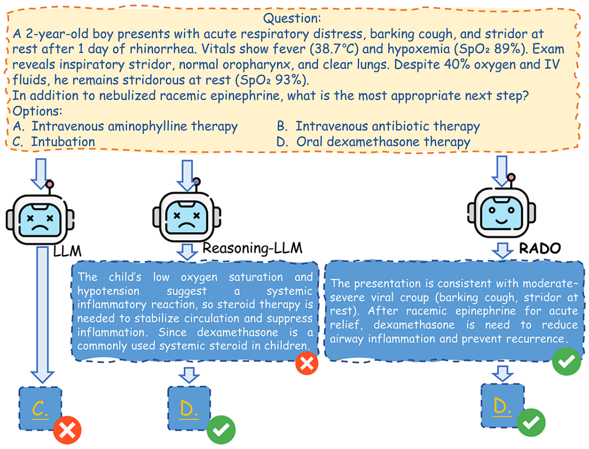

Figure 1: Comparison of reasoning paths across different models. LLMs often fail both answer and logic (left), while Reasoning-LLMs may arrive at the right answer for the wrong reasons (middle). In contrast, **RADO** ensures both a rigorous reasoning trajectory and a correct final decision through explicit auditing (right).

To bridge this gap, it is necessary to verify the comprehensiveness and rigor of the reasoning process. In complex reasoning tasks, generating an accurate and faithful derivation is exceptionally challenging because the generative model must ensure logical consistency from local steps to the global structure (i.e., correctness must be global) (Shao et al., 2024; Lightman et al., 2023). Conversely, discriminating the correctness of a reasoning path is typically simpler in terms of computational complexity and cognitive load (Welleck et al., 2023;

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4659–4683
July 2-7, 2026 ©2026 Association for Computational Linguistics

Uesato et al., 2022). This is due to a salient asymmetry in logical proofs: *a perfect proof requires global flawlessness, whereas an erroneous proof only requires a single local logical rupture (i.e., an error can be local)*. Based on this cognitive hypothesis, Audit models can enhance their discriminative capability from easily accessible negative samples with local errors, while generative models often only improve by learning from hard-to-obtain globally accurate positive samples, which may span 100k tokens and involve doctoral-level intellectual depth (Shao et al., 2025; Hosseini et al., 2024).

In this paper, we propose **RADO** (**R**easoning **A**udit-**D**riven **O**ptimization), a framework designed to enhance reasoning through explicit auditing. The framework operates in three stages: first, an initial Reward Model (RM) is built via Supervised Fine-Tuning (SFT); however, due to the RM's limitations in fine-grained discrimination, we further train a dedicated audit model. Second, the Audit Model performs rigorous logical scrutiny over diverse reasoning paths, identifying local logical flaws to construct high-confidence preference datasets. Notably, the Audit Model is empowered with external tool-calling capabilities, such as web search and numerical computation, to provide grounded verification of reasoning steps. This tool-augmented auditing mechanism significantly enhances the reliability of the audit signals and acts as a robust defense against reward hacking, where models might otherwise exploit the Reward Model's systematic blind spots. Finally, we employ Direct Preference Optimization (DPO) (Rafailov et al., 2023) to iteratively refine the RM's precision, subsequently using the upgraded RM to guide the Policy Model through Group Relative Policy Optimization (GRPO) (Shao et al., 2024) training.

This approach enables the model to explicitly perceive its reward function and maximize returns through deliberate reasoning. We evaluate **RADO** on five public datasets (Koa et al., 2024; Guha et al., 2023; Chen et al., 2021; Jin et al., 2019; Hendrycks et al., 2021) across three high-stakes domains: finance, law, and biomedicine. Experimental results demonstrate that RADO not only ensures final answer accuracy but also significantly enhances the rigor and safety of the reasoning process in high-stakes tasks.

Our main contributions are as follows:

• We propose RADO that leverages the logical asymmetry between generation and discrimination, using a specialized Audit Model to provide accurate supervision for reasoning trajectories.

• We introduce an auditing mechanism empowered by external tools. This ensures grounded verification of intermediate reasoning steps and effectively mitigates reward hacking.

• We demonstrate through extensive experiments in finance, law, and biomedicine that RADO significantly enhances both the logical rigor of reasoning paths and the accuracy of final outcomes.

2 Related Works

2.1 Large Reasoning Models (LRMs) in High-Stakes Domains

In high-stakes domains, the application of LRMs has evolved from pursuing outcome accuracy to ensuring process rigor. In biomedicine, Med-PaLM 2 (Singhal et al., 2025) and Med-Gemini (Saab et al., 2024) utilize knowledge-guided reasoning paths to enhance clinical reliability. Legal-specific LRMs, such as Lawma (Dominguez-Olmedo et al., 2025), emphasize the importance of generating legally sound derivations for judicial fairness. Similarly, financial models like Fin-r1 (Liu et al., 2025b), and Agentar-Fin-R1 (Zheng et al., 2025b) rely on explicit reasoning to mitigate risks in decision-making. These studies collectively suggest that detailed reasoning paths not only improve accuracy but also provide the necessary auditability for professional standards (Chu et al., 2025; Liu et al., 2025b). However, the rapid expansion of reasoning capabilities has introduced a new challenge: the logic explosion of reasoning paths. As models like DeepSeek-R1 (Guo et al., 2025) produce trajectories spanning millions of tokens, the complexity and granularity of these proofs often exceed the capacity of human reviewers to verify them efficiently (Shao et al., 2025; Lightman et al., 2023). This cognitive gap makes it difficult to detect subtle logical fallacies. Consequently, developing specialized audit models to autonomously monitor and calibrate these extended reasoning processes has become an essential frontier for self-verifiable AI in critical domains (Shao et al., 2025; Hosseini et al., 2024).

2.2 LLM-as-a-Judge

The LLM-as-a-Judge paradigm has emerged as a scalable alternative to human evaluation (Zheng et al., 2023; Yuan and Zhang, 2025b). This paradigm is intrinsically linked to the concept of an audit model, where a specialized model is used to

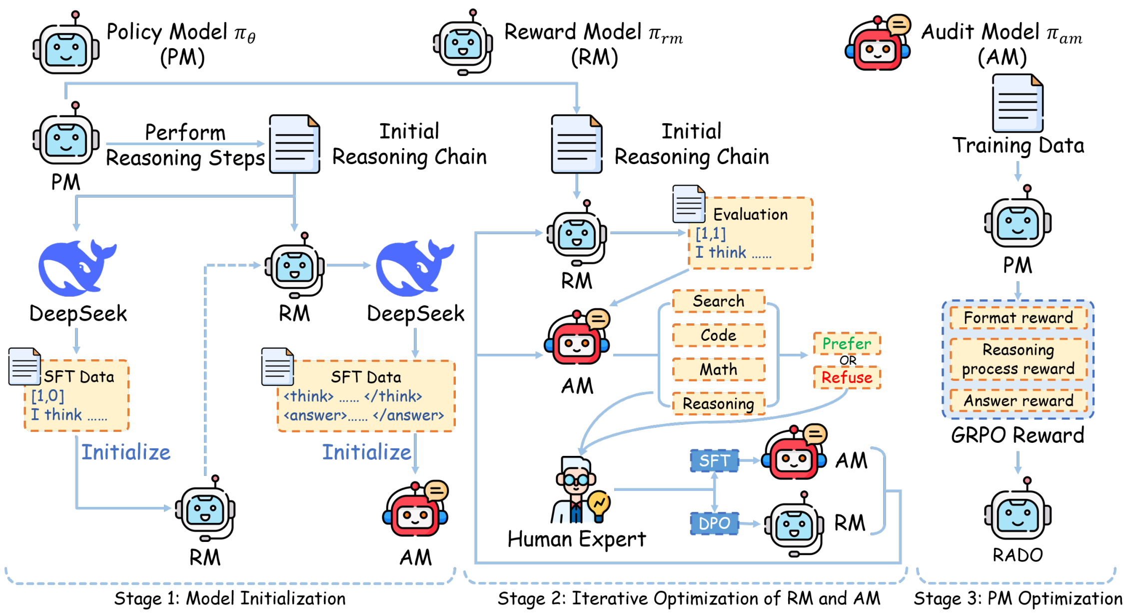

Figure 2: The overall architecture of the RADO framework. **Stage 1**: Initialization of RM and AM using SFT data from PM and DeepSeekv3.2. **Stage 2**: Iterative optimization where the AM audits reasoning chains using external tools to generate preference data for RM refinement via DPO. **Stage 3**: PM optimization using GRPO, where the finalized RM scores both the reasoning process and the final answer to provide a fine-grained reward signal.

critique the reasoning of a policy model. Construction methods for such judges typically involve SFT on human-labeled rubrics or synthetic feedback (Yuan et al., 2024; Lee et al., 2023). Beyond simple evaluation, judge models are increasingly used to drive the performance of LLMs through iterative refinement (Yuan et al., 2024). UltraFeedback (Cui et al., 2023) shows that multi-agent judging can mitigate position bias. RocketEval (Wei et al., 2025) utilize grading checklists to provide more granular audits. These developments establish that a judge model is not just a passive evaluator but a critical component in the judge-generate optimization loop (Li et al., 2025).

2.3 Outcome and Process Reward Models

Reward modeling is central to reinforcement learning from human feedback (RLHF). Outcome Reward Models (ORMs) like GRPO (Shao et al., 2024) or GSPO (Zheng et al., 2025a) evaluate the final state of a trajectory. However, ORMs may encourage the model finds shortcuts to correct answers through flawed logic (Tarek and Beheshti, 2025; Fan et al., 2025). To mitigate this, Process Reward Models (PRMs) have gained traction by providing step-level supervision (Lightman et al., 2023; Uesato et al., 2022). Although PRMs offer granular rewards, it is challenging to synthesize training data that possesses both adequate span and difficulty (Guo et al., 2025; Shao et al., 2025).

3 Methodology

In this section, we introduce the training framework of **RADO** (**R**easoning **A**udit-**D**riven **O**ptimization). As shown in Figure 2, this framework trains an Audit Model (AM) to optimize the Reward Model (RM), and then utilizes the optimized RM to score the samples from the Policy Model (PM) during GRPO (Shao et al., 2024), thereby guiding the optimization of PM. The overall framework consists of three training stages: (1) initializing RM and AM (in Section 3.1); (2) iteratively optimizing AM and RM through iterative data sampling for Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO) (Rafailov et al., 2023) (in Section 3.2); (3) using the optimized RM to guide the training of PM through GRPO (in Section 3.3).

3.1 Model Initialization

We first define the Policy Model π_θ, Reward Model π_{rm}, and Audit Model π_{am}. The Policy Model is initialized from open-source instruction models, such as Qwen2.5-Instruct-7B (Qwen-Team, 2024).

Reward Model Initialization. We collect 2k questions from financial (Chen et al., 2021), legal (Guha et al., 2023), and medical (Jin et al., 2019) question-answering datasets. The PM π_θ is prompted to generate multiple samples for each question, with detailed parameter settings provided in Appendix B. Then, we invoke the powerful general-purpose model DeepSeek-v3.2 (Liu et al., 2025a), inputting the original question along with each individual sample from π_θ one at a time. Through prompting, we ask DeepSeek-v3.2 to separately score the reasoning process and final answer of π_θ, and to output the rationale for the scores after providing them. Detailed prompt templates are provided in Appendix C. The response from DeepSeek-v3.2 follows the format:

> **[[s_{reason}, s_{answer}]]**
> *Rationale for the scores...*

where s_{reason} and s_{answer} represent the correctness scores for the reasoning process and final answer, respectively, satisfying $s_{\text{reason}}, s_{\text{answer}} \in \{0, 1\}$. We further conduct manual verification of DeepSeek-v3.2's responses to avoid bias or factual errors in its scoring. After verification, we filter samples with different score combinations to ensure sample balance, and then use SFT to train the same model as the initial Policy Model π_θ, obtaining the initialized Reward Model π_{rm}. More details are provided in Appendix F and Appendix D.

Audit Model Initialization. Similarly, we collect another 2k questions from financial, legal, and medical question-answering datasets. Again, we prompt the Policy Model π_θ to generate multiple samples for each question. Then, we let the initialized Reward Model π_{rm} generate scores for these questions and samples. Subsequently, we invoke DeepSeek-v3.2 inputting only the results from the Reward Model π_{rm}. Through prompting, we ask DeepSeek-v3.2 to review whether there are errors in π_{rm}'s evaluation. Detailed prompt templates are provided in Appendix C. In particular, the Audit Model is allowed to use external tools for fine-grained discrimination, such as web search and Python numerical computation function calls. Web search can obtain high-timeliness information, and numerical computation tools can obtain highly accurate numerical and analytical results. DeepSeek-v3.2 is invoked in a multi-turn dialogue manner to perform tool calls and obtain tool feedback, thereby constructing coherent data. The response from DeepSeek-v3.2 follows the format:

```
<think>...
<search>query</search>
<doc>summary</doc>
...
<function>y(a, b)</function>
<result>...</result>
...</think>
<answer>s_reward</answer>
```

where s_{reward} represents the correctness score of the Reward Model π_{rm}'s scoring, called the audit score, satisfying $s_{\text{reward}} \in \{0, 1\}$. Similarly, we conduct manual verification of DeepSeek-v3.2's responses to avoid bias or errors. Then, we use SFT in a multi-turn dialogue format to train the same model as the initial Policy Model π_θ, obtaining the initialized Audit Model π_{am}.

3.2 Iterative Optimization of RMs and AMs

Optimizing the Reward Model. Similar to Section 3.1, we sample another 4k questions from financial, legal, and biomedical question-answering datasets. We first prompt the Policy Model π_θ to generate samples for each question, and then sample the initialized Reward Model π_{rm} multiple times to generate scores for these questions and samples from π_θ. Detailed parameter settings are provided in Appendix B. Next, we let the initialized Audit Model π_{am} generate audit scores for the evaluations from π_{rm}. We construct preference pairs for the Reward Model π_{rm} by treating samples with a score of 1 from π_{rm} as positive samples and those with a score of 0 as negative samples. The negative samples reflect erroneous evaluations from π_{rm}. To optimize the initialized Reward Model π_{rm}, we introduce Direct Preference Optimization (DPO) (Rafailov et al., 2023), with the optimization objective:

$$\mathcal{L}_{\text{DPO}}(\pi_{\text{rm}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_{\text{rm}}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_{\text{rm}}(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right], \quad (1)$$

where π_{ref} is the reference model, and β controls the strength of the KL divergence constraint between the preference distribution and the reference model.

Iterative Optimization. To further improve the accuracy of the reward model's discrimination, we design an iterative optimization mechanism with Human-in-the-loop. This mechanism can be summarized as a reward-audit-correct-update loop. In this work, the number of iterations is $K = 2$.

Specifically, in the k-th iteration ($k \in \{1, \dots, K\}$), we utilize the Reward Model $\pi_{\text{rm}}^{(k-1)}$ and Audit Model$\pi_{\text{am}}^{(k-1)}$ optimized from the previous round to perform the following steps. π_{rm}^{0} denotes the Reward Model obtained from the first round of DPO, and π_{am}^{0} denotes the Audit Modelafter initialization.

***1.* Hard Sample Mining**: We use $\pi_{\text{rm}}^{(k-1)}$ to evaluate the sampling results from the Policy Model π_θ, and have $\pi_{\text{am}}^{(k-1)}$ audit the evaluation results. We mainly filter out: (1) *disagreement samples*, i.e., samples judged as incorrect by the Audit Model, and (2) *high-score samples*, i.e., samples judged as correct by the Audit Model with the Reward Model scoring [1,1].

***2.* Human-in-the-loop Annotation**: To prevent the model from developing self-reinforcing biases during iteration, we introduce human experts to verify the difficult samples filtered above. Annotators are required to review the tool invocation path and final judgment of the audit model. If the Audit Model's judgment is erroneous (e.g., incorrect tool invocation parameters leading to a false negative judgment of the Reward Model), annotators will correct the audit logic and labels; if the audit model's judgment is correct but the Reward Model is wrong, annotators will confirm the negative sample and supplement positive samples to construct new preference pairs. This process builds a high-quality corrected dataset $\mathcal{D}^{(k)} = \mathcal{D}_{\text{rm}}^{(k)} \cup \mathcal{D}_{\text{am}}^{(k)}$.

3.* Model Update**: We first construct the cumulative training set for the Audit Model$\mathcal{S}_{\text{am}}^{(k)} = \bigcup_{i=0}^{k} \mathcal{D}_{\text{am}}^{(i)}$, where $\mathcal{D}_{\text{am}}^{(0)}$ is the SFT data from the initialization phase. We perform SFT on $\pi_{\text{am}}^{(k-1)}$ using $\mathcal{S}_{\text{am}}^{(k)}$ to obtain the enhanced Audit Model $\pi_{\text{am}}^{(k)}$. Subsequently, we repeat the Reward Model optimization process in Section 3.2, using the updated Audit Model $\pi_{\text{am}}^{(k)}$ for re-auditing, thereby constructing the preference dataset $\mathcal{D}_{\text{rm}}^{(0)}$. After merging with the preference dataset obtained in Step ***2, we obtain $\mathcal{S}_{\text{rm}}^{(k)} = \bigcup_{i=0}^{k} \mathcal{D}_{\text{rm}}^{(i)}$. After DPO optimization on $\mathcal{S}_{\text{rm}}^{(k)}$, we obtain $\pi_{\text{rm}}^{(k)}$.

Through this iterative approach, the Audit Model gradually learns more sophisticated error correction techniques, while the Reward Model, under stricter audit pressure, is forced to generate evaluations with more rigorous logic and more accurate facts, thereby achieving co-evolution of the Reward Model and Audit Model.

3.3 Policy Model Optimization

After obtaining the iteratively optimized Reward Model (denoted as π_{rm}), we use GRPO (Shao et al., 2024) to optimize the Policy Model π_θ. Specifically, for a given question q, GRPO first uses the current policy π_{old} to generate G different responses $\{o_1, o_2, \ldots, o_G\}$ and get the corresponding rewards $\{r_1, r_2, \ldots, r_G\}$. The optimization objective is:

$$\mathcal{L}_{GRPO}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\Bigg(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\text{old}}(o_i|q)}A_i,\right. \\ \left.\text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\text{old}}(o_i|q)}, 1-\epsilon, 1+\epsilon\right)A_i\right) - \beta\mathbb{D}_{KL}(\pi_\theta||\pi_{ref})\Bigg), \tag{2}$$

$$A_i = \frac{r_i - \text{mean}(\{r_1, \ldots, r_G\})}{\text{std}(\{r_1, \ldots, r_G\})}, \tag{3}$$

$$\mathbb{D}_{KL}(\pi_\theta||\pi_{ref}) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1, \tag{4}$$

where ϵ and β control the clipping range and the strength of the KL divergence penalty, respectively. π_{ref} is the reference policy used to prevent the optimized policy π_θ from deviating too far and causing catastrophic forgetting.

We set three rewards:

a). Format reward r_{fmt}: The model receives a score of 1 when it strictly follows the predefined interaction format, i.e., first thinking <think>...</think> and then providing an answer <answer>...</answer>; otherwise, the score is 0. **b). Reasoning process reward** r_{rsn}: Provided by π_{rm}, corresponding to s_{reason} in its output. The score is 0 or 1. **c). Answer reward** r_{ans}: Provided by π_{rm}, corresponding to s_{answer} in its output. The score is 0 or 1.

Finally, we set the total reward as:

$$r = r_{\text{fmt}} \cdot r_{\text{rsn}} \cdot r_{\text{ans}}. \tag{5}$$

4 Experiments

In this section, we evaluate the performance of RADO across multiple high-stakes domains. Our experiments are designed to address the following three research questions: RQ1: How does RADO perform in terms of final answer accuracy when applied to specialized domains ? RQ2: To what extent does RADO improve the quality of the generated reasoning paths? RQ3: How significant is the performance degradation in the absence of the Audit Model or external tool-calling capabilities?

4.1 Experimental Settings

Baselines. To validate RADO's performance on high-stakes domain tasks, we compare it with general purpose LLMs and domain LLMs trained or fine-tuned with domain data.

Model	Model size	Interpretation					Rule-application/ Rule-conclusion	Rhetorical-analysis			Avg.
		CC	CAUD	MAUD	PP	IP	PJ	Scalr	TTP	TTD	
DeepSeek-R1-Llama	8B	83.58	75.07	38.52	55.14	49.64	60.00	56.56	95.32	63.03	64.10
GPT-OSS	20B	**97.22**	83.14	51.02	57.41	38.34	80.00	48.33	95.32	79.37	70.02
HIPO	8B	90.65	76.52	63.81	58.19	45.86	74.00	64.79	91.58	**82.48**	71.98
MiMo-RL	7B	90.38	84.39	58.67	60.05	33.33	40.00	70.80	91.58	56.52	65.08
Qwen-2.5-Instruct	7B	86.36	80.08	78.75	52.38	48.12	64.00	78.98	99.07	76.96	73.86
Llama-3-Instruct	8B	85.86	81.20	67.43	61.63	50.37	54.00	75.83	**100.00**	78.18	72.72
Open-Australian-Legal	1.5B	0.00	0.00	1.20	17.64	1.50	22.00	0.00	0.00	0.00	4.70
DISC-LawLLM	13B	50.00	32.98	64.77	48.09	19.55	56.00	70.05	5.60	20.60	40.85
Law-LLM	7B	10.86	1.59	30.87	3.05	2.26	0.00	58.49	8.41	13.33	14.32
Law-Chat	7B	80.30	82.31	39.75	51.69	33.83	48.00	76.36	54.21	52.73	57.69
Lawma	8B	47.73	34.14	69.93	53.31	47.37	36.00	78.46	6.54	26.67	44.46
Domain*o*1s-legal	7B	88.64	81.76	80.33	66.54	52.63	72.00	88.97	95.33	78.78	78.33
RADO-legal	7B	91.25	**92.09**	**82.39**	**67.34**	**53.72**	**74.00**	**89.23**	99.06	80.02	**81.11**

Table 1: Model accuracy (%) on LegalBench (Guha et al., 2023). Avg. is the mean accuracy across all tasks.

General Purpose LLMs: DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025), HIPO-8B (Deng et al., 2025), GPT-OSS-20B (Agarwal et al., 2025), MiMo-7B-RL (Xiaomi et al., 2025), Qwen2.5-Instruct-7B (Qwen-Team, 2024), and Llama-3-Instruct (AI@Meta, 2024).

Financial Domain LLMs: Fin-R1 (Liu et al., 2025b), Fin-o1-7B (Qian et al., 2025), Finance-LLM (Cheng et al., 2024b), Finance-Chat (Cheng et al., 2024b), Finance-Llama-3 (Cheng et al., 2024a), Domain*o*1s-finance (Chu et al., 2025).

Legal Domain LLMs: Open-Australian-Legal-LLM (Butler, 2023), DISC-LawLLM (Yue et al., 2023), Law-LLM (Cheng et al., 2024b), Law-Chat (Cheng et al., 2024b), and Lawma (Dominguez-Olmedo et al., 2024), Domain*o*1s-legal (Chu et al., 2025).

Biomedicine Domain LLMs: BioMistral-7B (Labrak et al., 2024b), medicine-LLM-13B (Cheng et al., 2024b), II-Medical-8B (Bonzi et al., 2025), PMC-LLAMA-7B (Wu et al., 2024).

Datasets. For financial domain tasks, we select the Fino1 dataset (Qian et al., 2025) as the training set, and FinQA (Chen et al., 2021) as well as the stock prediction dataset provided by Koa et al. (Koa et al., 2024) as the test sets.

For legal domain tasks, we adopt the two datasets LegalBench (Guha et al., 2024) and Case-HOLD (Zheng et al., 2021) as the combined training set, and use LegalBench as the test set.

For biomedicine domain tasks, we select MedQA (Jin et al., 2021) as the training set, and PubMedQA (Jin et al., 2019) as well as the professional medicine task of MMLU (Hendrycks et al., 2021) as the test sets.

Model	MMLU	PubMedQA
BioMistral-7B	49.33	53.40
Medicine-LLM-13B	37.50	58.40
II-Medical-8B	79.26	78.00
PMC-LLAMA-7B	56.98	76.40
DeepSeek-R1-Llama-8B	64.48	64.20
GPT-OSS-20B	79.54	79.80
HIPO-8B	78.47	79.92
MiMo-7B-RL	74.12	78.93
Qwen2.5-Instruct-7B	75.36	62.40
Llama3-Instruct-8B	71.32	69.80
RADO-biomedicine	**80.54**	**80.39**

Table 2: Model accuracy (%) on MMLU (Hendrycks et al., 2021) and PubMedQA (Jin et al., 2019).

For more specific details about the datasets, please refer to the Appendix G.

Implementation Details. In this work, our RADO is developed based on Qwen-2.5-Instruct-7B (Qwen-Team, 2024). More implementation details can be found in Appendix A.

4.2 Prediction Accuracy

To answer RQ1, we compare the performance of RADO with other baseline methods. Tables 1, 2, and 3 report the numerical results of RADO in the legal, medical, and financial domains, respectively. For all models included in these tables, we use DeepSeek-v3.2 (DeepSeek-AI, 2025) to extract and analyze the answers. This is because reasoning models such as DeepSeek-R1-Llama-8B (Guo et al., 2025) often include a large amount of fine-grained reasoning steps in their responses, and do not always output answers in a standard format. Moreover, for datasets in the financial domain, the final answers may involve various forms of numer-

Model	Stock Prediction	FinQA
Finance-Llama3-8B	48.61	47.32
Fin-o1-7B	49.72	64.48
Finance-LLM-8B	50.02	51.09
Finance-Chat-8B	48.77	56.21
Domain*o*1s-finance-7B	49.20	72.05
DeepSeek-R1-Llama-8B	48.09	46.09
GPT-OSS-20B	51.23	76.14
HIPO-8B	50.34	74.11
MiMo-7B-RL	51.42	69.32
Qwen2.5-Instruct-7B	51.34	66.79
Llama3-Instruct-8B	47.58	42.01
RADO-finance	**52.58**	**81.52**

Table 3: Model accuracy on stock prediction (Koa et al., 2024) and FinQA (Chen et al., 2021).

ical results (decimals, fractions, etc.), and comparing and analyzing these results goes beyond a reasonable workload for us. Therefore, we rely on DeepSeek-v3.2 to perform this task on our behalf. This is similar to DocMath (Zhao et al., 2024a).

The experimental results across legal, biomedical, and financial domains are presented in Tables 1, 2, and 3. RADO (7B) consistently achieves SOTA performance, outperforming both domain-specific models (e.g., Lawma, BioMistral) and general reasoning baselines. Notably, despite having significantly fewer parameters, RADO surpasses GPT-OSS-20B across all aggregated metrics. As observed, RADO performs slightly better than GPT-OSS-20B on the majority of tasks, with marginal gains in biomedical benchmarks (~0.6-1.0%) and stock prediction (+1.35%). For benchmarks like FinQA and LegalBench, RADO demonstrates more substantial improvements (+5.38% and +11.09% respectively).

The Audit Model proposed in this work could undoubtedly serve as an evaluation tool like DeepSeek-v3.2: analyzing the reasoning process of the models and producing possible scores. However, considering that **1)** in the LLM-as-a-judge paradigm, models of the same family (Qwen2.5 (Qwen-Team, 2024), Qwen3 (Yang et al., 2025), etc.) tend to assign higher scores, we did not use the audit models for scoring or answer extraction, as this could lead to unfair scores for our generative models (Li et al., 2025); and **2)** the Audit Models we trained are not comparable in scale to large models like DeepSeek-v3.2, and prior experience suggests that larger models generally achieve better results. In fact, when using the corresponding auditing model to verify the final answers, we observed approximately a 2% performance improvement across all datasets in every domain. This also suggests that, when reasoning models generate thousands of tokens, selecting an appropriate evaluation model becomes a crucial issue.

The reasoning chain length and inference time of RADO and baselines can be seen Appendix H.

4.3 Reasoning Path Evaluation

To answer RQ2, we evaluate the quality of the reasoning trajectories generated by the models discussed in this work. In high-stakes domains, while the accuracy of the final output is paramount, the rigor of the underlying reasoning process is equally critical (Qian et al., 2025; Bonzi et al., 2025; Chu et al., 2025). To quantitatively assess this process, we employ the **PROOF-Score** (Chu et al., 2025), a metric specifically designed for the multi-dimensional evaluation of reasoning quality.

PROOF-Score uses DeepSeek-v3.2 (DeepSeek-AI, 2025), Gemini-3-flash (Team and Google, 2025), GPT-5.1 (OpenAI, 2025) and the same prompt in (Chu et al., 2025) to generate a score from 1 to 7 for response, considering 3 aspects:

• **Reasoning Completeness (RC)**: Evaluates the completeness and logical coherence. • **Domain Safety (DS)**: Measures the safety and appropriateness in specific domains. • **Factual Accuracy (FA)**: Evaluates the factual accuracy of statements.

Then PROOF-Score is calculated by:

$$\text{PROOF-Score} = \frac{RC + DS + FA}{3}. \quad (6)$$

Model	Gemini 3 Flash	GPT 5.1	DeepSeek-v3.2	Audit Model†
Legal Domain (Audit Model†: RADO-Audit-Legal†)				
DeepSeek-R1-Llama-8B	6.11	6.12	6.27	5.97
GPT-OSS-20B	6.38	**6.42**	6.52	6.22
HIPO-8B	6.22	6.18	6.37	6.07
MiMo-7B-RL	6.19	6.30	6.43	6.08
Domain*o*1s-legal	6.34	6.32	6.49	6.26
RADO-legal	**6.40**	6.37	**6.61**	**6.42**
Financial Domain (Audit Model†: RADO-Audit-Finance†)				
DeepSeek-R1-Llama-8B	6.08	5.95	6.04	5.79
GPT-OSS-20B	6.32	6.35	6.42	6.18
HIPO-8B	6.15	6.11	6.24	5.92
MiMo-7B-RL	**6.31**	6.24	6.39	6.02
Domain*o*1s-finance	6.27	6.29	6.32	6.11
RADO-finance	**6.31**	**6.47**	**6.49**	**6.35**
Biomedicine Domain (Audit Model†: RADO-Audit-biomedecine†)				
DeepSeek-R1-Llama-8B	6.42	6.32	6.48	6.16
GPT-OSS-20B	6.50	6.64	6.55	6.43
HIPO-8B	6.53	6.49	6.69	6.37
MiMo-7B-RL	6.50	6.62	6.75	6.38
II-Medical-8B	6.66	6.64	6.71	6.57
RADO-biomedicine	**6.72**	**6.69**	**6.84**	**6.82**

Table 4: Comparative evaluation of reasoning quality via PROOF-Score across different evaluators.

Table 4 presents the PROOF-Score results for the reasoning paths generated by various reasoning models. For domains involving multiple test sets, we report the average scores across these datasets. Specifically, Audit-Legal (and its counterparts in other domains) refers to the specialized Audit Model co-trained during the RADO optimization process. We linearly map the average scores of the Audit Model from the range [0,1] to [1,7].

Analysis of the PROOF-Score results in Table 4 yields the following key insights: **1)** Audit models consistently assign lower scores than frontier LLMs like Gemini 3 or GPT 5.1. This suggests that while general judges focus on linguistic fluency, audit models penalizing subtle logical leaps that general LLMs often overlook. **2)** RADO achieves the highest PROOF-Scores across all domains. This indicates that our audit-driven optimization significantly enhances the structural integrity and logical transparency of reasoning paths. **3)** In fact, all reasoning LLMs achieve relatively high PROOF-Scores across the evaluated datasets. This suggests that these models generally maintain a high level of logical consistency on the tasks at hand, frequently arriving at correct conclusions through valid intermediate derivations. For the LLM-as-a-Judge paradigm, judge models tend to assign higher scores to longer responses. However, outputs generated by reasoning LLMs are generally lengthy. (Li et al., 2025; Chu et al., 2025).To better verify the quality of the reasoning paths generated by RADO, we conducted a human expert evaluation, which can be found in Appendix I.

4.4 Ablation Study

To answer RQ3, We further investigate the contributions of individual components within the RADO framework.

Dataset	Model Variant	Score
LegalBench	RADO-legal	**81.11**
	RADO-legal w/o	77.20
	Qwen2.5-Instruct-7B	73.86
MMLU	RADO-biomedicine	**80.54**
	RADO-bio w/o tool	74.81
	Qwen2.5-Instruct-7B	75.36

Table 5: Part of ablation results. "w/o tool" indicates the variant without external tool access during auditing.

Is tool-augmented auditing essential for the auditor model? In RADO, the Audit Model is permitted to invoke external tools for verification. To evaluate this component, we prohibit tool calls during the training of the Audit Model while keeping all other configurations constant. Table 5 presents part of the final comparative results. A complete table is provided in the Appendix J.1.

Without tool invocation, RADO is susceptible to reward hacking, which may lead to performance degradation relative to the base model (on the MMLU datasets, compared with the original Qwen2.5-Instruct model).

Is the presence of an independent auditor model indispensable? In this section, we compare the following setups: 1) SFT on the original dataset. 2) SFT on the negative samples. 3) SFT on original dataset + negative samples. 4) DPO on original dataset + negative samples. 5) GRPO on the original dataset. 6) GRPO on the negative samples. 7) GRPO on original dataset + negative samples. No additional audit models are used in any of these setups. The negative samples are consistent with those presented in Section 3.2.

Table 6 presents part of the final results. "-pos" denotes the use of the original dataset, while "-neg" denotes the use of negative samples. A complete table is provided in the Appendix 11. For reasoning LLMs, constructing high-quality positive samples that meet the requirements is really challenging, whereas generating negative samples with localized errors is relatively straightforward. However, as shown in the Table 6, directly utilizing negative samples fails to yield a satisfactory improvement in model performance. In contrast, leveraging negative samples to enhance the discriminative ability of the Audit Model can improve the quality of the Reward Model, which in turn ultimately boosts the performance of the Policy Model.

Dataset	Model Variant	Score
LegalBench	RADO-legal	**81.11**
	Qwen2.5-Instruct-7B-Pos-Neg-GRPO	77.28
	Qwen2.5-Instruct-7B-Neg-GRPO	74.59
	Qwen2.5-Instruct-7B-Pos-GRPO	77.33
	Qwen2.5-Instruct-7B-Pos-Neg-DPO	74.67
	Qwen2.5-Instruct-7B-Pos-Neg-SFT	75.53
	Qwen2.5-Instruct-7B-Neg-SFT	73.92
	Qwen2.5-Instruct-7B-Pos-SFT	75.21
	Qwen2.5-Instruct-7B	73.86

Table 6: Ablation study for audit model.

More ablation experiments are provided in the Appendix J.

5 Conclusion

This paper introduces **RADO**, a framework that enhances reasoning rigor in high-stakes domains through explicit auditing. By leveraging the logical asymmetry between generation and verification, we develop a tool-augmented Audit Model that provides grounded supervision and mitigates reward hacking. Our experiments across financial, legal, and biomedicine tasks confirm that RADO consistently improves both final answer accuracy and the logical integrity of reasoning paths. This method is also applicable to high-stakes areas such as social governance.

Limitations

Despite its effectiveness, our work has several limitations. First, the computational overhead is increased during the training phase due to the iterative optimization of the Audit Model and the requirement for human-in-the-loop verification. While this process yields a more robust reward model, it requires more resources than standard single-stage RLHF. Second, the performance of the Audit Model is partially dependent on the quality of external tools; inaccuracies in web search results or numerical tool execution could propagate errors back to the reward signals. Finally, although we evaluate RADO in finance, law, and medicine, its generalization to more reasoning tasks (e.g., creative writing or open-ended debate) remains to be explored.

6 Ethical Considerations

Fairness and Accessibility. Our framework is built entirely upon open-source foundations to ensure broad accessibility and transparency. The Policy Models are initialized from public weights (Qwen-2.5-Instruct-7B (Qwen-Team, 2024)), and all training data is sourced from established open-source benchmarks (FinQA (Chen et al., 2021), LegalBench (Guha et al., 2023), and PubMedQA (Jin et al., 2019)). To ensure fairness, we implement a rigorous Human-in-the-loop (HITL) verification process during the iterative update stages. Human experts explicitly review and correct potential biases—such as linguistic overreliance or stereotypical reasoning patterns—that might be inherited from the teacher model or sampling noise. This proactive manual intervention ensures that the co-evolution of the reward and audit models is grounded in neutral and objective logic rather than biased data distributions.

Risk Mitigation in High-Stakes Reasoning. Reasoning in financial, legal, and medical domains involves high-stakes decision-making where errors can lead to severe real-world consequences. RADO mitigates these risks by employing an audit model. The Audit Model is specifically trained to identify local logical rupture through tool-augmented verification and reasoning. Crucially, we prioritize the detection and elimination of high-risk or hazardous reasoning paths—such as those suggesting illegal financial maneuvers, unethical legal advice, or dangerous medical self-treatment mannually.

Data Integrity and Bias. To maintain the integrity, we do not solely rely on automated scoring from general-purpose models. We acknowledge that even powerful models like DeepSeek-v3.2 (DeepSeek-AI, 2025) can exhibit systematic biases. Therefore, we perform cross-verification between the audit model's feedback and human expert annotations. This iterative reward-audit-correct-update loop is designed to break the self-reinforcing bias cycle common in self-training frameworks. By filtering for sample balance across different score combinations (s_{reason} and s_{answer}), we ensure the models are trained on a diverse and representative set of reasoning trajectories.

Computational Efficiency and Environmental Impact. We acknowledge that RADO introduces additional computational overhead compared to standard Reinforcement Learning (RL) by maintaining and training an auxiliary audit model. However, this cost is strategically managed. First, the use of GRPO avoids the heavy memory requirements of a separate value function (critic) model during policy training. Second, while the iterative optimization of the AM and RM adds training cycles, the computational resources required for these stages are negligible when compared to the massive scale of initial pre-training or even large-scale SFT on general datasets. The Audit Model serves as a high-efficiency precision instrument that operates on a relatively small, high-quality dataset (e.g., 2k–4k samples), ensuring that the incremental carbon footprint is minimal relative to the substantial gains in model reliability and safety.

Acknowledgments

This work is supported by the National Key R&D Program of China [2023YFC3304900].

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K

Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

AI@Meta. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

Doria Bonzi, Alexandre Guiggi, Frédéric Béchet, Carlos Ramisch, and Benoit Favre. 2025. Caremedeval dataset: Evaluating critical appraisal and reasoning in the biomedical field. *arXiv preprint arXiv:2511.03441*.

Umar Butler. 2023. Open australian legal llm.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024a. Instruction pretraining: Language models are supervised multitask learners. *arXiv preprint arXiv:2406.14491*.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2024b. Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations*.

Inyoung Cheong, King Xia, KJ Kevin Feng, Quan Ze Chen, and Amy X Zhang. 2024. (a) i am not a lawyer, but...: Engaging legal experts towards responsible llm policies for legal advice. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 2454–2469.

Xu Chu, Zhijie Tan, Hanlin Xue, Guanyu Wang, Tong Mo, and Weiping Li. 2025. Domain*o*1s: Guiding LLM reasoning for explainable answers in high-stakes domains. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3275–3293.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *CoRR*.

DeepSeek-AI. 2025. Deepseek-v3.2: Pushing the frontier of open large language models.

Chenlong Deng, Kelong Mao, and Zhicheng Dou. 2024. Learning interpretable legal case retrieval via knowledge-guided case reformulation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1253–1265.

Ken Deng, Zizheng Zhan, Wen Xiang, Wenqiang Zhu, Weihao Li, Jingxuan Xu, Tianhao Peng, Xinping Lei, Kun Wu, Yifan Yao, and 1 others. 2025. Hipo: Hybrid policy optimization for dynamic reasoning in llms. *arXiv preprint arXiv:2509.23967*.

Ricardo Dominguez-Olmedo, Vedant Nanda, Rediet Abebe, Stefan Bechtold, Christoph Engel, Jens Frankenreiter, Krishna Gummadi, Moritz Hardt, and Michael Livermore. 2024. Lawma: The power of specialization for legal tasks. *arXiv preprint arXiv:2407.16615*.

Ricardo Dominguez-Olmedo, Vedant Nanda, Rediet Abebe, Stefan Bechtold, Christoph Engel, Jens Frankenreiter, Krishna P Gummadi, Moritz Hardt, and Michael Livermore. 2025. Lawma: The power of specialization for legal annotation. In *The Thirteenth International Conference on Learning Representations*.

Lishui Fan, Yu Zhang, Mouxiang Chen, and Zhongxin Liu. 2025. Posterior-grpo: Rewarding reasoning processes in code generation. *arXiv preprint arXiv:2508.05170*.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, and 1 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in neural information processing systems*, 36:44123–44279.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, and 1 others. 2024. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in Neural Information Processing Systems*, 36.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *arXiv preprint arXiv:2103.06268*.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. V-star: Training verifiers for self-taught reasoners. In *Conference on Language Modeling*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 conference on empirical methods* *in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 2567–2577.

Kelvin JL Koa, Yunshan Ma, Ritchie Ng, and Tat-Seng Chua. 2024. Learning to generate explainable stock predictions using self-reflective large language models. In *Proceedings of the ACM on Web Conference 2024*, pages 4304–4315.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024a. Biomistral: A collection of open-source pretrained large language models for medical domains. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5848–5864, Bangkok, Thailand. Association for Computational Linguistics.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024b. Biomistral: A collection of open-source pretrained large language models for medical domains. *arXiv preprint arXiv:2402.10373*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and 1 others. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv e-prints*, pages arXiv–2309.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, and 1 others. 2025. From generation to judgment: Opportunities and challenges of llm-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2757–2791.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025a. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Zhaowei Liu, Xin Guo, Fangqi Lou, Lingfeng Zeng, Jinyi Niu, Zixuan Wang, Jiajie Xu, Weige Cai, Ziwei Yang, Xueqian Zhao, and 1 others. 2025b. Fin-r1: A large language model for financial reasoning through reinforcement learning. *arXiv preprint arXiv:2503.16252*.

OpenAI. 2025. Gpt-5.1. `https://openai.com/index/gpt-5-1/`. Large language model (released 13 November 2025).

Lingfei Qian, Weipeng Zhou, Yan Wang, Xueqing Peng, Jimin Huang, and Qianqian Xie. 2025. Fino1: On the transferability of reasoning enhanced llms to finance. *arXiv preprint arXiv:2502.08127*.

Qwen-Team. 2024. Qwen2.5: A party of foundation models.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Khaled Saab, Tao Tu, Wei-Hung Weng, Ryutaro Tanno, David Stutz, Ellery Wulczyn, Fan Zhang, Tim Strother, Chunjong Park, Elahe Vedadi, and 1 others. 2024. Capabilities of gemini models in medicine. *arXiv preprint arXiv:2404.18416*.

Zhihong Shao, Yuxiang Luo, Chengda Lu, ZZ Ren, Jiewen Hu, Tian Ye, Zhibin Gou, Shirong Ma, and Xiaokang Zhang. 2025. Deepseekmath-v2: Towards self-verifiable mathematical reasoning. *arXiv preprint arXiv:2511.22570*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Weijie Shi, Han Zhu, Jiaming Ji, Mengze Li, Jipeng Zhang, Ruiyuan Zhang, Jia Zhu, Jiajie Xu, Sirui Han, and Yike Guo. 2025. Legalreasoner: Step-wised verification-correction for legal judgment reasoning. *arXiv preprint arXiv:2506.07443*.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *arXiv preprint arXiv:2506.06941*.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, and 1 others. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31(3):943–950.

Mirza Farhan Bin Tarek and Rahmatollah Beheshti. 2025. Reward hacking mitigation using verifiable composite rewards. *arXiv preprint arXiv:2509.15557*.

Gemini Team and Google. 2025. Gemini 3: Frontier models for advanced multimodal reasoning and efficiency. Google Technical Report.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

Cheng Wang, Yue Liu, Baolong Bi, Duzhen Zhang, Zhong-Zhi Li, Yingwei Ma, Yufei He, Shengju Yu, Xinfeng Li, Junfeng Fang, Jiaheng Zhang, and Bryan Hooi. 2025. Safety in large reasoning models: A survey. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 3468–3482.

Haochun Wang, Chi Liu, Nuwa Xi, Zewen Qiang, Sendong Zhao, Bing Qin, and Ting Liu. 2023. Huatuo: Tuning llama model with chinese medical knowledge. *arXiv preprint arXiv:2304.06975*.

Tianjun Wei, Wei Wen, Ruizhi Qiao, Xing Sun, and Jianghong Ma. 2025. Rocketeval: Efficient automated llm evaluation via grading checklist. *arXiv preprint arXiv:2503.05142*.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2023. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. 2024. Pmc-llama: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, 31(9):1833–1843.

LLM Xiaomi, Bingquan Xia, Bowen Shen, Dawei Zhu, Di Zhang, Gang Wang, Hailin Zhang, Huaqiu Liu, Jiebao Xiao, Jinhao Dong, and 1 others. 2025. Mimo: Unlocking the reasoning potential of language model–from pretraining to posttraining. *arXiv preprint arXiv:2505.07608*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Haohan Yuan and Haopeng Zhang. 2025a. Domainsum: A hierarchical benchmark for fine-grained domain shift in abstractive text summarization. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2219–2231.

Haohan Yuan and Haopeng Zhang. 2025b. Understanding llm reasoning for abstractive summarization. *arXiv preprint arXiv:2512.03503*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. 2024. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and 1 others. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325*.

Yilun Zhao, Yitao Long, Hongjun Liu, Ryo Kamoi, Linyong Nan, Lyuhao Chen, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. 2024a. Docmath-eval: Evaluating math reasoning capabilities of llms in understanding long and specialized documents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16103–16120.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2024b. Swift:a scalable lightweight infrastructure for fine-tuning. *Preprint*, arXiv:2408.05517.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025a. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. 2021. When does pretraining help? assessing self-supervised learning for law and the casehold dataset. In *Proceedings of the 18th International Conference on Artificial Intelligence and Law*. Association for Computing Machinery.

Yanjun Zheng, Xiyang Du, Longfei Liao, Xiaoke Zhao, Zhaowen Zhou, Jingze Song, Bo Zhang, Jiawei Liu, Xiang Qi, Zhe Li, and 1 others. 2025b. Agentar-fin-r1: Enhancing financial intelligence through domain expertise, training efficiency, and advanced reasoning. *arXiv preprint arXiv:2507.16802*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Appendix

A Experimental Settings

All experiments are conducted using 8 NVIDIA A800-SXM4 (80GB) GPUs. All evaluation results reported in this paper represent the mean value across 3 independent runs.

The computational environment consists of a 1.5 TB memory system running Ubuntu 18.04.6, with Python 3.10.18, PyTorch 2.6.0, CUDA 12.3, and Transformers 4.53.3.

For Supervised Fine-Tuning (SFT), the learning rate is set to 1.0e-5 over 7 epochs. We use a batch size of 2 per device with a gradient accumulation of 1, and the maximum sequence length is capped at 4096 tokens.

For Direct Preference Optimization (DPO), the learning rate is 5.0e-6 with a training duration of 7 epochs. The batch size per device and gradient accumulation are set to 2 and 1, respectively. The maximum sequence length is 4096, the warmup ratio is 0.1, and the coefficient β is set to 0.1.

For Group Relative Policy Optimization (GRPO), we employ a learning rate of 2e-5 for 3 epochs. The global batch size is 4, and both temperature and top-p sampling are set to 1.0. The coefficient β is 0.1, and the number of generations per prompt is set to 4.

Across all training stages, we use the Adam optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e - 8$.

For implementation, we leverage the **LLaMA-Factory** framework (Zheng et al., 2024) to conduct Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO). The subsequent Group Relative Policy Optimization (GRPO) stage is implemented using the **MS-Swift** library (Zhao et al., 2024b).

B Parameter Setting Details

Model Initialization. The temperature for Policy Model sampling is set to 0.5, the p-value for top-p sampling is set to 1.0, and the number of samples per question is 4.

Iterative Optimization. The temperature for Policy Model sampling is set to 0.7, the p-value for top-p sampling is set to 1.0, and the number of samples per question is 1. The temperature for Reward Model sampling is set to 0.5, the p-value for top-p sampling is set to 1.0, and the number of samples per question is 8.

C Prompt Templates

Prompt template for calling the Policy Model to sample answers:

```
{question}

Given the above question, please provide
your answer in the following format. First
output your thinking process within
<think></think> tags, then output your
final answer within <answer></answer>
tags.

For example: <think>Let me
first...</think><answer>Yes</answer>

Please return the output directly in this
format without any content outside the
format.
```

Prompt template for calling DeepSeek-v3.2 to generate SFT data for the reward model:

```
You are an expert. I will give you a
question, the correct answer, and a
model's response. Please evaluate whether
the model's response is correct and output
scores.

Question:
{question}

Correct Answer:
{reference_response}

Model's Response:
{response}

In the model's response, the content
inside "<think></think>" is the reasoning
process, and the content inside
"<answer></answer>" is the final answer.
You should score them separately. Output
[think_score, answer_score]. Both
think_score and answer_score can only be 0
or 1. 1 means correct, 0 means there is an
error.

You should output the scores first, then
provide the analysis.
Follow the format below, for example:[1,0]
The model's reasoning process mentions...
```

Prompt template for calling the audit model to sample answers:

```
You are an expert. Below, I will provide
you with an expert model's evaluation of a
piece of generated content. In this
evaluation, the two elements in the array
represent think_score and answer_score
respectively, which indicate the score for
the thinking process and the score for the
final answer. Both are binary (either 0 or
1), where 1 represents correct and 0
represents an error. Please assess whether
the expert model's evaluation is objective
and error-free.

Expert's evaluation:
{output}

If you think it's necessary, you can
search the web. Write the query to search
between <search></search>, and then I will
write a summary of the retrieved results
between <doc></doc>. Then you can proceed
with reasoning. If you think it's
necessary, you can call Python for
numerical calculations. You have 6 basic
arithmetic functions: add(a, b),
subtract(a, b), multiply(a, b), divide(a,
b), correlation(x, y),
linear_regression(x, y). Write the
function and values to calculate between
<function></function>, and then I will
write the calculation result between
<result></result>. Then you can proceed
with reasoning. You should first provide
your analysis process, then give a score
of 0 or 1, enclosed between
<answer></answer>. 1 means you believe the
expert's evaluation is objective and
correct, 0 means you believe the expert's
evaluation contains errors.

Maintain the following format, for example:
In the expert's evaluation...<search>US
Constitution</search><doc>Article I of the
Constitution...</doc>Next, let's...
<function>multiply(2,
3)</function><result>6.0</result> Finally,
let's...<answer>1</answer>
```

Tools callable by the audit model:

1) Numerical computation tools:

```python
def add(a, b):
    return float(a + b)

def subtract(a, b):
    return float(a - b)

def multiply(a, b):
    return float(a * b)

def divide(a, b):
    if b == 0:
        raise ValueError
    return float(a / b)

def correlation(x, y):
    if len(x) != len(y):
        raise ValueError
    n = len(x)
    if n == 0:
        raise ValueError

    mean_x = sum(x) / n
    mean_y = sum(y) / n

    num = sum((x[i] - mean_x) *
        (y[i] - mean_y) for i in
        range(n))
    den_x = sum((x[i] - mean_x) **
        2 for i in range(n))
    den_y = sum((y[i] - mean_y) **
        2 for i in range(n))

    denom = (den_x * den_y) ** 0.5
    if denom == 0:
        raise ValueError

    return float(num / denom)

def linear_regression(x, y):
    if len(x) != len(y):
        raise ValueError
    n = len(x)
    if n == 0:
        raise ValueError

    mean_x = sum(x) / n
    mean_y = sum(y) / n

    num = sum((x[i] - mean_x) *
        (y[i] - mean_y) for i in
        range(n))
    den = sum((x[i] - mean_x) ** 2
        for i in range(n))
    if den == 0:
        raise ValueError

    slope = num / den
    intercept = mean_y - slope *
        mean_x

    return float(slope),
        float(intercept)
```

2) Search tools: The input is a text query. DuckDuckGo search is used to retrieve the top 5 results, retaining the document title and document sum-

mary text as returned content.

D Details of Score Combinations Ratios

Regarding the score combinations discussed in Section 3.1, there are four different categories: $[0,0]$, $[0,1]$, $[1,0]$, and $[1,1]$. We maintain a balanced distribution, with each category accounting for exactly 25% of the total samples.

E Robustness to Teacher Model and Reproducibility

To address concerns regarding the reliance on DeepSeek-v3.2 for RM/AM initialization inspired by the domain shift challenge highlighted in DomainSum(Yuan and Zhang, 2025a), we further evaluated the robustness of RADO by employing a weaker teacher model and significantly reduced training data.

In this experiment, we replaced DeepSeek-v3.2 with Qwen3-30B-A3B-Thinking (Yang et al., 2025) as the teacher for generating rationales and scores. Furthermore, to test the model's efficiency under data-constrained scenarios, we significantly reduced the training set to only 1,000 samples, while keeping all other experimental hyperparameters identical to the main experiments.

We evaluated this variant (RADO-Qwen) on two domain-specific benchmarks: LegalBench and MMLU-medicine. The results are summarized in Table 7.

Model	Score
LegalBench	
Qwen-2.5 7B	73.86
RADO-legal 7B	**81.11**
RADO-legal-Qwen 7B	79.75
MMLU-medicine	
Qwen-2.5 7B (Base)	75.36
RADO-biomedicine 7B	**80.54**
RADO-biomedicine-Qwen 7B	78.92

Table 7: Ablation results using a weaker teacher model (Qwen3-30B-A3B-Thinking) with reduced training data.

As shown in the results, even when initialized by a weaker teacher and trained on a minimal dataset, RADO continues to yield substantial performance gains over the base model. This demonstrates that the core mechanisms of RADO are robust to the choice of the teacher model and can achieve competitive results even in resource-limited settings.

F Human Verification Details

In this section, we provide details of the manual verification for the Audit Model, Policy Model, and Reward Model. Among all validated samples, approximately 20% of the samples undergo manual correction to ensure data quality. To prevent error propagation into reward signals, we implemented a rigorous verification and Human-in-the-Loop (HITL) pipeline:

Since function definitions are deterministic, the risk is limited to parsing. In our training set, 92.5% of numerical tasks successfully invoked the tool. For the 5.8% of samples with parsing errors, we performed manual rewriting to ensure 100% grounded reward signals. Only 6.3% of samples required web search. Only 2.8% of documents conflicted with ground truth and were corrected manually. For 19.2% of unhelpful retrievals, we annotated these samples to train the model to ignore irrelevant noise. For the fully trained Audit Model (AM), the function invocation rate for numerical problems reached 99.6%, with a formatting error rate of only 1.3%. Even in cases of formatting errors, incorrect results are not returned to the AM; instead, the system simply ignores the function call and continues the reasoning process. Regarding the trained AM, the web search rate was 10.2% with zero formatting errors. A manual audit of 500 sampled trajectories showed that search-induced errors occurred in only 0.8% of the cases, demonstrating the high reliability of the retrieved information. Furthermore, considering the heavy workload of manual labeling, the first three authors of this paper have made important and equally indispensable contributions.

F.1 Policy Model (PM) Output Refinement

Reasoning Process Correction: For samples where the reasoning process is incorrect but the final answer is correct, we rectify the derivation in the following aspects:

- **Language Inconsistency**: Correct non-English languages appearing in the reasoning process.
- **Garbled Text**: Remove artifacts and ensure contextual coherence.
- **Insufficient Reasoning**: Rewrite insufficient derivations, including blank reasoning, answer-only outputs, or paths with fewer than 100 characters.

- **Logical Fallacies**: Fix errors including logical ruptures, contradictions between the derivation and the final answer or ground truth, numerical inaccuracies, factual errors, and high-stakes or unethical content.

Final Answer Correction: For samples with incorrect final answers, we first trace the errors back to the reasoning process before revising the final answer.

F.2 Reward Model (RM) Output Refinement

Score Calibration: For samples with erroneous scoring, we simultaneously modify the reward scores and their corresponding justifications:

- $s_{\mathbf{answer}}$: Check the PM's final answer to ensure the score is factually grounded (1 for correct, 0 for incorrect). We specifically correct RM biases, such as: (1) *Over-tolerance*, e.g., treating 14.0 as consistent with 14.3; (2) *Over-derivation*, where the RM attempts to rationalize an incorrect path to support the ground truth.

- $s_{\mathbf{reason}}$: Ensure the reasoning scores are factually accurate. We rectify biases where the RM awards 1 point to: (1) *Multilingual reasoning* (non-English content), which may degrade RL sampling; (2) *Garbled text* or *meaningless characters*.

Explanation Refinement: Rewrite the justifications for calibrated scores to maintain consistency, addressing language inconsistency, garbled text, insufficient reasoning, and logical errors.

F.3 Audit Model (AM) Output Refinement

Thought Process Rectification: Correct inconsistencies in language, garbled text, insufficient reasoning, and logical fallacies within the internal thoughts.

Tool-Call Verification: Validate that tool interactions follow the predefined format: search queries within <search> tags, retrieved results within <doc>, function calls within <function> (ensuring no nesting and adherence to input schemas), and results within <result>.

Conclusion Calibration: Analyze whether the RM's output is correct and compare it with the AM's response to ensure the AM's final conclusion is accurate.

G Dataset Details

Fino1 dataset is provided by TheFinAI team (Qian et al., 2025). This dataset contains 8,281 financial QA pairs constructed by 11 finance professionals from the earnings reports of S&P 500 companies. The Fino1 dataset enhances the original FinQA (Chen et al., 2021) by adding GPT-4o-generated reasoning paths to each question-answer pair. These reasoning paths provide step-by-step explanations of how to arrive at the correct answer from the given financial context, making it particularly suitable for training and evaluating models on structured financial question answering tasks. The task is designed to evaluate a model's ability to perform numerical reasoning over financial data, requiring the integration of structured tables and unstructured textual context from financial reports while handling complex domain-specific terminology. Models are expected to generate both the correct answer and a clear reasoning path demonstrating their analytical process.

The stock investment recommendation dataset (Koa et al., 2024) contains price data and tweet information for the top 5 stocks from 11 industries during 2020-2022, comprising 7,866 test question entries. The task is constructed to predict whether a stock will rise or fall on the next trading day based on facts contained in tweets from the previous 5 days. Any neutral answers are considered incorrect.

LegalBench (Guha et al., 2023): a dataset composed of numerous legal QA datasets and benchmarks. LegalBench includes 5 categories of legal tasks. We select three reasoning-related categories: Rule-application/Rule-conclusion, Interpretation, and Rhetorical-understanding, encompassing 9 datasets with a total of 35,053 test questions. Question types include true/false and multiple-choice questions.

CaseHold (Zheng et al., 2021): a legal dataset consisting of over 53,000 multiple-choice questions, designed to identify the relevant holdings of cited cases. This dataset presents a fundamental task for lawyers and is legally meaningful and challenging from an NLP perspective (with an F1 score of 0.4 for the BiLSTM baseline). The citation context within judicial decisions serves as the prompt for each question. The answer options are holding statements derived from the cited texts in legal decisions. Each cited text is accompanied by five answer choices: the correct answer is the holding statement corresponding to the cited text, while the four incorrect options are holding statements from other cases.

MedQA (Jin et al., 2021): MedQA is a medical question-answering dataset presented in a multiple-choice format. Its questions are sourced from the medical licensing examinations. These examinations are designed to assess physicians' professional knowledge and clinical decision-making capabilities. Covering a wide range of topics, the questions typically require a deep understanding of relevant medical concepts. The dataset contains a total of 61,097 questions, including 12,723 in English, 34,251 in Simplified Chinese, and 14,123 in Traditional Chinese.

PubMedQA (Jin et al., 2019): PubMedQA is a biomedicine question-answering (QA) dataset compiled from PubMed abstracts. The task of PubMedQA is to answer research questions using corresponding abstracts, with the answers formatted as yes/no/maybe. The dataset consists of 1,000 expert-annotated QA instances, 61,200 unannotated instances, and 211,300 artificially generated QA instances. Each PubMedQA instance includes four components: (1) a question, which may be the title of an existing research article or derived from it; (2) a context, namely the corresponding abstract excluding the conclusion section; (3) a long answer, which refers to the conclusion section of the abstract and is assumed to address the research question; (4) a yes/no/maybe answer that summarizes the conclusion.

MMLU (Hendrycks et al., 2021): The MMLU dataset comprises 57 tasks, covering elementary mathematics, American history, computer science, law, and other disciplines. Each task consists of a set of multiple-choice questions, with every question provided with four options and one correct answer. The MMLU-Professional Medicine task consists of 272 multiple-choice questions, all of which are four-option objective questions designed to evaluate a model's grasp of the core knowledge required for medical professionals.

H Answer Length & Inference time

Table 8 presents a comparative analysis of inference time and the average length of reasoning chains across various models in the legal domain. Our observations are as follows:

The Trade-off between Latency and Accuracy. Reasoning-centric models, including GPT-OSS, HIPO, MiMo-RL, and the DeepSeek-R1 series, exhibit a distinct behavior of generating extensive internal rationale before arriving at a final answer. While traditional instruction-tuned models like Qwen-2.5-Instruct and Llama-3-Instruct produce concise responses ($\approx$ 1.5 tokens) in less than 0.6s, reasoning models generate trajectories spanning 190 to 320 tokens, resulting in latencies ranging from 5.3s to 7.3s. The detailed reasoning paths not only safeguard high accuracy but also serve as indispensable audit path that assist legal experts in verifying the model's underlying logic, thereby mitigating the risk of deceptive alignment where a model might reach a correct conclusion through flawed premises.

Model	time(s)	Length
DeepSeek-R1-Llama-8B	7.21	313.1
GPT-OSS-20B	5.34	196.2
HIPO-8B	6.82	249.3
MiMo-RL-7B	7.31	283.6
Qwen-2.5-Instruct-7B	0.51	1.4
Llama-3-Instruct-8B	0.42	1.5
Open-Australian-1.5B	7.24	267.4
DISC-LawLLM-13B	1.92	11.3
Law-LLM-7B	3.97	94.3
Law-Chat-7B	0.47	1.1
Lawma-8B	0.64	1.1
Domain$o$1s-legal-7B	9.21	346.1
RADO-legal-7B	7.02	273.2

Table 8: Inference time and reasoning chain length on legal domain tasks.

Competitive Efficiency of RADO. A closer inspection of specialized models reveals that Domain$o$1s-legal incurs the highest computational cost (9.21s per query), primarily due to its reliance on real-time Tree-Search exploration which involves evaluating multiple reasoning branches. In contrast, **RADO-legal** achieves a superior balance between rigor and efficiency. With an inference time of 7.02s and a reasoning length of 273.2 tokens, RADO maintains a computational profile comparable to general-purpose reasoning models (e.g., DeepSeek-R1-Llama-8B at 7.21s).

I Human Evaluation Methodology and Details

To further validate the logical precision of the reasoning trajectories, we conduct a human study on the FinQA dataset. We randomly sample 20 questions and collect 160 reasoning paths generated by

8 different models. Each path contains an average of 250 tokens, ensuring sufficient depth for logical auditing.

Expert Background and Scoring. We invite three Ph.D.-level experts with financial backgrounds to evaluate the trajectories. The scoring criteria remain consistent with the 1–7 scale used in the PROOF-Score metric, focusing on reasoning completeness, domain safety, and factual accuracy.

Results and Consistency Analysis. The mean scores provided by the three experts are 6.50, 6.30, and 6.40, respectively, resulting in a collective human average of 6.40. To measure the reliability of these assessments, we calculate the Fleiss' Kappa coefficient, which reaches 0.68. Given the wide 1–7 scoring range and the complexity of the task, a Kappa of 0.68 represents substantial agreement among experts, confirming the stability of our human-annotated standard.

Human-Model Alignment. As illustrated in Table 9, the variation between human experts and automated judge models is remarkably low. The absolute deviation ($|\Delta|$) across all evaluators does not exceed 0.09. This confirms that RADO produces reasoning paths that satisfy expert-level standards and that our audit framework serves as a reliable proxy for professional human judgment.

Evaluator	RADO Score	Human Mean	Variation (Δ)
Human Experts (Avg.)	6.40	-	-
Gemini 3 Flash	6.31	6.40	-0.09
GPT-5.1	6.47	6.40	+0.07
DeepSeek-v3.2	6.49	6.40	+0.09
Audit-Finance	6.35	6.40	-0.05

Table 9: Expert-Model Alignment: Comparison between human expert mean scores and automated PROOF-Scores for RADO-finance.

J Ablation Study

J.1 External Tools

Table 10 presents the ablation results of the RADO algorithm across 5 datasets in legal, biomedicine, and finance domains, where we compare the full RADO model, its tool-free variant (w/o tool), and the baseline Qwen2.5-Instruct-7B. Consistent performance advantages of the full RADO model are observed across all tasks: it achieves the highest score in every dataset, with the most significant improvement of 19.58 points on PubMedQA (80.39 vs. 60.81 of the tool-free variant). In the legal domain, RADO-legal outperforms the tool-free variant by 3.91 points (81.11 vs. 77.20) and the baseline by 7.25 points, validating the effectiveness of external tools in legal text auditing. For the biomedical domain, RADO-biomedicine surpasses the baseline by 5.18 points on MMLU pro medicine tasks (Hendrycks et al., 2021) and outperforms the tool-free variant by a large margin on PubMedQA (Jin et al., 2019), which highlights the critical role of external tools in processing professional biomedical data. In the finance domain, RADO-finance achieves marginal gains on Stock Prediction (52.58 vs. 50.31 of the tool-free variant) and substantial improvements on FinQA (81.52 vs. 74.39 of the tool-free variant and 66.79 of the baseline).

Dataset	Model Variant	Score
LegalBench	RADO-legal	**81.11**
	RADO-legal w/o	77.20
	Qwen2.5-Instruct-7B	73.86
MMLU	RADO-biomedicine	**80.54**
	RADO-bio w/o tool	74.81
	Qwen2.5-Instruct-7B	75.36
PubMedQA	RADO-biomedicine	**80.39**
	RADO-biomedicine w/o tool	60.81
	Qwen2.5-Instruct-7B	62.40
Stock Pred.	RADO-finance	**52.58**
	RADO-fin w/o tool	50.31
	Qwen2.5-Instruct-7B	51.34
FinQA	RADO-finance	**81.52**
	RADO-fin w/o tool	74.39
	Qwen2.5-Instruct-7B	66.79

Table 10: Ablation results. "w/o tool" indicates the variant without external tool access during auditing.

J.2 Audit Model

Table 11 presents the performance comparison between RADO domain-specific variants and eight Qwen2.5-Instruct-7B variants (differing in data composition: Pos for original data, Neg for negative samples; and training paradigms: SFT, DPO, GRPO) across five datasets covering legal, biomedicine, and finance domains. Consistent with our hypotheses, RADO models achieve the highest scores in all tasks, demonstrating the superiority of its external tool-integrated audit framework over fine-tuned baseline models.

In the legal domain (LegalBench), RADO-legal attains a score of 81.11, outperforming the best Qwen2.5 variant (Qwen2.5-Instruct-7B-Pos-GRPO, 77.33) by 3.78 points. For the biomedicine domain, RADO-biomedicine reaches 80.54 on MMLU and 80.39 on PubMedQA, surpassing the top-performing Qwen2.5 variants in these two datasets (75.21 and 73.86, respectively) by more than 5 points.

Dataset	Model Variant	Score
LegalBench	RADO-legal	**81.11**
	Qwen2.5-Instruct-7B-Pos-Neg-GRPO	77.28
	Qwen2.5-Instruct-7B-Neg-GRPO	74.59
	Qwen2.5-Instruct-7B-Pos-GRPO	77.33
	Qwen2.5-Instruct-7B-Pos-Neg-DPO	74.67
	Qwen2.5-Instruct-7B-Pos-Neg-SFT	75.53
	Qwen2.5-Instruct-7B-Neg-SFT	73.92
	Qwen2.5-Instruct-7B-Pos-SFT	75.21
	Qwen2.5-Instruct-7B	73.86
MMLU	RADO-biomedicine	**80.54**
	Qwen2.5-Instruct-7B-Pos-Neg-GRPO	75.21
	Qwen2.5-Instruct-7B-Neg-GRPO	68.86
	Qwen2.5-Instruct-7B-Pos-GRPO	74.47
	Qwen2.5-Instruct-7B-Pos-Neg-DPO	74.43
	Qwen2.5-Instruct-7B-Pos-Neg-SFT	72.90
	Qwen2.5-Instruct-7B-Neg-SFT	64.76
	Qwen2.5-Instruct-7B-Pos-SFT	71.39
	Qwen2.5-Instruct-7B	62.40
PubMedQA	RADO-biomedicine	**80.39**
	Qwen2.5-Instruct-7B-Pos-Neg-GRPO	75.81
	Qwen2.5-Instruct-7B-Neg-GRPO	65.16
	Qwen2.5-Instruct-7B-Pos-GRPO	71.43
	Qwen2.5-Instruct-7B-Pos-Neg-DPO	72.39
	Qwen2.5-Instruct-7B-Pos-Neg-SFT	73.48
	Qwen2.5-Instruct-7B-Neg-SFT	64.23
	Qwen2.5-Instruct-7B-Pos-SFT	70.21
	Qwen2.5-Instruct-7B	62.40
Stock Pred.	RADO-finance	**52.58**
	Qwen2.5-Instruct-7B-Pos-Neg-GRPO	51.12
	Qwen2.5-Instruct-7B-Neg-GRPO	50.69
	Qwen2.5-Instruct-7B-Pos-GRPO	50.32
	Qwen2.5-Instruct-7B-Pos-Neg-DPO	50.83
	Qwen2.5-Instruct-7B-Pos-Neg-SFT	51.46
	Qwen2.5-Instruct-7B-Neg-SFT	49.37
	Qwen2.5-Instruct-7B-Pos-SFT	50.23
	Qwen2.5-Instruct-7B	51.34
FinQA	RADO-finance	**81.52**
	Qwen2.5-Instruct-7B-Pos-Neg-GRPO	76.62
	Qwen2.5-Instruct-7B-Neg-GRPO	73.21
	Qwen2.5-Instruct-7B-Pos-GRPO	75.21
	Qwen2.5-Instruct-7B-Pos-Neg-DPO	76.72
	Qwen2.5-Instruct-7B-Pos-Neg-SFT	75.41
	Qwen2.5-Instruct-7B-Neg-SFT	68.80
	Qwen2.5-Instruct-7B-Pos-SFT	73.41
	Qwen2.5-Instruct-7B	66.79

Table 11: Performance metrics of the Qwen2.5-Instruct-7B model across eight distinct training setups. Pos denotes the use of the original dataset, Neg denotes the use of negative samples, and SFT/DPO/GRPO denote the training methods employed.

In the finance domain, RADO-finance achieves 52.58 on Stock Prediction and 81.52 on FinQA. Specifically, on FinQA, it outperforms the optimal Qwen2.5 variant (Qwen2.5-Instruct-7B-Pos-Neg-DPO, 76.72) by 4.8 points; on Stock Prediction, it exceeds the leading Qwen2.5 variant (Qwen2.5-Instruct-7B-Pos-Neg-SFT, 51.46) by 1.12 points. Meanwhile, alignment-based training methods (GRPO, DPO) outperform SFT in most scenarios—for instance, on LegalBench, Qwen2.5-Instruct-7B-Pos-GRPO (77.33) outperforms Qwen2.5-Instruct-7B-Pos-SFT (75.21), and on FinQA, Qwen2.5-Instruct-7B-Pos-Neg-DPO (76.72) surpasses Qwen2.5-Instruct-7B-Pos-Neg-SFT (75.41). An exception is observed in Stock Prediction, where the vanilla Qwen2.5-Instruct-7B (51.34) performs comparably to most fine-tuned variants, suggesting limited optimization space for this task via existing training setups. Finally, the vanilla Qwen2.5-Instruct-7B achieves the lowest scores on LegalBench (73.86), MMLU (62.40), PubMedQA (62.40), and FinQA (66.79), confirming that fine-tuning with task-specific data and alignment methods is indispensable for improving baseline performance.

J.3 Iterations of the Reward-Audit Loop

In section 3.2, the number of iterations of the reward-audit-correct-update loop is set to 2. We extend the loop number to 3 in this experiment.

Dataset	Model Variant	Score
LegalBench	RADO-legal-3	**81.23**
	RADO-legal-2	81.11
	RADO-legal-1	77.28
	Qwen2.5-Instruct-7B	73.86
MMLU	RADO-biomedicine-3	79.91
	RADO-biomedicine-2	**80.54**
	RADO-biomedicine-1	78.21
	Qwen2.5-Instruct-7B	75.36
FinQA	RADO-finance-3	82.32
	RADO-finance-2	81.52
	RADO-finance-1	76.64
	Qwen2.5-Instruct-7B	66.79

Table 12: Ablation results. RADO-legal-1 means 1 loop for RADO-legal.

As can be easily observed from the 12, simply increasing the number of loop iterations fails to effectively improve model performance. Moreover, this update requires human annotation, which incurs substantial resource consumption.

K Case Study

K.1 Legal task

Example of Policy Model

(**Groud Truth**: Yes)

Input:
Here is a clause:
To the fullest extent permitted by law, the parties waive and relinquish any claims, demands, causes of action or recoveries for punitive damages, exemplary damages, or statutory damages.
Here is a question about it:
Does the clause specify a cap on liability upon the breach of a party's obligation? This includes time limitation for the counterparty to bring claims or maximum amount for recovery.
Answer with Yes/No.
Given the above question, please provide your answer in the following format. First output your thinking process within <think></think> tags, then output your final answer within <answer></answer> tags. For example: <think>Let me first...</think><answer>Yes</answer>
Please return the output directly in this format without any content outside the format.

Output:
<think>I need to analyze the given clause to determine if it specifies a cap on liability, a time limitation for bringing claims, or a maximum amount for recovery. The clause mentions waiving and relinquishing claims for punitive, exemplary, and statutory damages. It does not mention anything about capping liability, time limits, or maximum amounts for recovery.</think>
<answer>No</answer>

Example of Reward Model

Input:
You are an expert. I will give you a question, the correct answer, and a model's response. Please evaluate whether the model's response is correct and output scores.
Question:
Here is a clause:
To the fullest extent permitted by law, the parties waive and relinquish any claims, demands, causes of action or recoveries for punitive damages, exemplary damages, or statutory damages.
Here is a question about it:
Does the clause specify a cap on liability upon the breach of a party's obligation? This includes time limitation for the counterparty to bring claims or maximum amount for recovery.
Answer with Yes/No. Given the above question, please provide your answer in the following format. First output your thinking process within <think></think> tags, then output your final answer within <answer></answer> tags. For example: <think>Let me first...</think><answer>Yes</answer> Please return the output directly in this format without any content outside the format.

Correct Answer:
Yes

Model's Response:
<think>I need to analyze the given clause to determine if it specifies a cap on liability, a time limitation for bringing claims, or a maximum amount for recovery. The clause mentions waiving and relinquishing claims for punitive, exemplary, and statutory damages. It does not mention anything about capping liability, time limits, or maximum amounts for recovery.</think>
<answer>No</answer>
In the model's response, the content inside <think></think> is the reasoning process, and the content inside <answer></answer> is the final answer. You should score them separately. Output [think_score, answer_score]. Both think_score and answer_score can only be 0 or 1. 1 means correct, 0 means there is an error. You should output the scores first, then provide the analysis. Follow the format below, for example:[1,0] The model's reasoning process mentions...

Output:
[0,0]
The model's reasoning process incorrectly interprets the clause. The clause explicitly waives claims for punitive, exemplary, and statutory damages, which effectively limits (or caps) the types of damages that can be claimed. This constitutes a cap on liability, even though it does not specify a numerical limit or time limitation. The correct answer should be Yes' because the waiver of certain damages types inherently limits liability. The model's reasoning fails to recognize this nuance, leading to an incorrect answer.

Example of Audit Model

Input:
You are an expert. Below, I will provide you with an expert model's evaluation of a piece of generated content. In this evaluation, the two elements in the array represent think_score and answer_score respectively, which indicate the score for the thinking process and the score for the final answer. Both are binary (either 0 or 1), where 1 represents correct and 0 represents an error.
Please assess whether the expert model's evaluation is objective and error-free.

Expert's evaluation:
[0,0]
The model's reasoning process incorrectly interprets the clause. The clause explicitly waives claims for punitive, exemplary, and statutory damages, which effectively limits (or caps) the types of damages that can be claimed. This constitutes a cap on liability, even though it does not specify a numerical limit or time limitation. The correct answer should be "Yes" because the waiver of certain damages types inherently limits liability. The model's reasoning fails to recognize this nuance, leading to an incorrect answer.

If you think it's necessary, you can search the web. Write the query to search between `<search></search>`, then write a summary of the retrieved results between `<doc></doc>`. Then proceed with reasoning.
If you think it's necessary, you can call Python for numerical calculations. You have 4 basic arithmetic functions: `add(a, b)`, `subtract(a, b)`, `multiply(a, b)`, `divide(a, b)`. Write the function and values to calculate between `<function></function>`, then write the calculation result between `<result></result>`. Then proceed with reasoning.
You should first provide your analysis process, then give a score of 0 or 1, enclosed between `<answer></answer>`. 1 means you believe the expert's evaluation is objective and correct, 0 means you believe the expert's evaluation contains errors. Maintain the following format, for example: In the expert's evaluation...<search>US Constitution</search><doc>Article I of the Constitution...</doc>Next, let's...<function>multiply(2, 3)</function><result>6.0</result>Finally, let's...<answer>1</answer>

Output:
In the expert's evaluation of the model's reasoning and answer, the scores provided are [0,0]. This indicates that the expert believes both the reasoning process and the final answer are incorrect. To determine whether this evaluation is objective and error-free, let's carefully analyze the expert's reasoning step by step.
1. Claim: The model incorrectly interprets the clause.
a. The clause waives claims for punitive, exemplary, and statutory damages.
b. The expert argues that this waiver constitutes a cap on liability, even though it does not specify a numerical or time-based limitation.
2. Conclusion: The correct answer should be "Yes" because waiving certain types of damages inherently limits liability. The model's failure to recognize this nuance leads to both an incorrect reasoning process and an incorrect final answer.
Now, to evaluate this reasoning, I will verify the legal understanding of waivers and their relationship to liability caps. To confirm the expert's interpretation, I need to search for relevant information.
<search>waiving damages vs capping liability</search>
<doc>
Title: Limitations of liability—Scenario One: Waivers of
Content: Waivers of consequential damages have become the industry standard, and these clauses are found in most industry templates. Let's pick on the AIA ...
Title: Considerations for Waiver and Release of Liability Contracts
Content: For the waiver and release of liability contract for your Texas gym to be considered conspicuous by a court, the law requires that the agreement be ...
Title: Should punitive damages be capped?
Content: Product Liability: In 2005, punitive damages were awarded in only 1 percent of product liability cases with a successful plaintiff.
Title: The original LOL - Limitation of Liability and its not so
Content: There are different ways to manage the risk of liability. A limitation of liability clause can seek to cap the total amount of damages a party can ...
Title: Collision damage waiver or personal insurance: choose wisely
Content: It's important to note that a CDW is not technically insurance, but rather a waiver of the rental company's right to collect damages from you in the ...
</doc>
From the retrieved documents:
1. Waivers of damages, such as consequential or punitive damages, are commonly included in industry agreements and contracts. These waivers are considered a method to limit liability.
2. A waiver does not necessarily involve a numerical cap (e.g., a dollar amount) but can still function as a limitation by excluding certain types of recoverable damages.
3. Limitation of liability clauses can take various forms, and waivers are one such form.
The expert's interpretation aligns with the general understanding of liability waivers. Waiving claims for specific types of damages (e.g., punitive, exemplary, and statutory damages) reduces the scope of liability, even if it does not specify a numerical limit.
<answer>1</answer>

K.2 Financial task

Example of Policy Model

(**Groud Truth**: 14.197%)

Input:
Please answer the given financial question based on the context.
Context: marathon oil corporation notes to consolidated financial statements (g) this obligation relates to a lease of equipment at united states steel 2019s clairton works cokemaking facility in pennsylvania . we are the primary obligor under this lease . under the financial matters agreement , united states steel has assumed responsibility for all obligations under this lease . this lease is an amortizing financing with a final maturity of 2012 . (h) these notes are senior secured notes of marathon oil canada corporation . the notes are secured by substantially all of marathon oil canada corporation 2019s assets . in january 2008 , we provided a full and unconditional guarantee covering the payment of all principal and interest due under the senior notes . (i) these obligations as of december 31 , 2009 include $ 36 million related to assets under construction at that date for which a capital lease will commence upon completion of construction . the amounts currently reported are based upon the percent of construction completed as of december 31 , 2009 and therefore do not reflect future minimum lease obligations of $ 164 million related to the asset . (j) payments of long-term debt for the years 2010 - 2014 are $ 102 million , $ 246 million , $ 1492 million , $ 287 million and $ 802 million . united steel is due to pay $ 17 million in 2010 , $ 161 million in 2011 , $ 19 million in 2012 , and $ 11 for year 2014 . (k) in the event of a change in control , as defined in the related agreements , debt obligations totaling $ 662 million at december 31 , 2009 , may be declared immediately due and payable . (l) see note 16 for information on interest rate swaps . 20 . asset retirement obligations the following summarizes the changes in asset retirement obligations : (in millions) 2009 2008 .

(in millions)	2009	2008
asset retirement obligations as of january 1	$965	$1134
liabilities incurred including acquisitions	14	30
liabilities settled	-65	-94
accretion expense (included in depreciation depletion and amortization)	64	66
revisions to previous estimates	124	24
held for sale	-	-195
asset retirement obligations as of december 31 (a)	$1102	$965

asset retirement obligations as of december 31 (a) $ 1102 $ 965 (a) includes asset retirement obligation of $ 3 and $ 2 million classified as short-term at december 31 , 2009 , and 2008.
Question: by what percentage did asset retirement obligations increase from 2008 to 2009?
Given the above question, please provide your answer in the following format. First output your thinking process within `<think></think>` tags, then output your final answer within `<answer></answer>` tags. For example: `<think>`Let me first...`</think><answer>`Yes`</answer>`
Please return the output directly in this format without any content outside the format.

Output:
`<think>`I need to calculate the percentage increase in asset retirement obligations from 2008 to 2009. The values provided are $965 million for 2008 and $1102 million for 2009. The formula to calculate percentage increase is ((New Value - Original Value) / Original Value) * 100. Let's plug in the numbers.`</think>`
`<answer>`The percentage increase in asset retirement obligations from 2008 to 2009 is ((1102 - 965) / 965) * 100 = 14.23%. Therefore, the asset retirement obligations increased by approximately **14.23%** from 2008 to 2009.`</answer>`

Example of Reward Model

Input:
You are an expert. I will give you a question, the correct answer, and a model's response. Please evaluate whether the model's response is correct and output scores.
Question:
Please answer the given financial question based on the context.
Context: marathon oil corporation notes to consolidated financial statements (g) this obligation relates to a lease of equipment at united states steel 2019s clairton works cokemaking facility in pennsylvania . we are the primary obligor under this lease . under the financial matters agreement , united states steel has assumed responsibility for all obligations under this lease . this lease is an amortizing financing with a final maturity of 2012 . (h) these notes are senior secured notes of marathon oil canada corporation . the notes are secured by substantially all of marathon oil canada corporation 2019s assets . in january 2008 , we provided a full and unconditional guarantee covering the payment of all principal and interest due under the senior notes . (i) these obligations as of december 31 , 2009 include $36 million related to assets under construction at that date for which a capital lease will commence upon completion of construction . the amounts currently reported are based upon the percent of construction completed as of december 31 , 2009 and therefore do not reflect future minimum lease obligations of $164 million related to the asset . (j) payments of long-term debt for the years 2010 - 2014 are $102 million , $246 million , $1492 million , $287 million and $802 million . united steel is due to pay $17 million in 2010 , $161 million in 2011 , $19 million in 2012 , and $11 for year 2014 . (k) in the event of a change in control , as defined in the related agreements , debt obligations totaling $662 million at december 31 , 2009 , may be declared immediately due and payable . (l) see note 16 for information on interest rate swaps . 20 . asset retirement obligations the following summarizes the changes in asset retirement obligations : (in millions) 2009 2008 .

(in millions)	2009	2008
asset retirement obligations as of january 1	965	1134
liabilities incurred including acquisitions	14	30
liabilities settled	-65	-94
accretion expense (included in depreciation depletion and amortization)	64	66
revisions to previous estimates	124	24
held for sale	-	-195
asset retirement obligations as of december 31 (a)	1102	965

asset retirement obligations as of december 31 (a) 1102 965 (a) includes asset retirement obligation of $3 and $2 million classified as short-term at december 31 , 2009 , and 2008 .
Question: by what percentage did asset retirement obligations increase from 2008 to 2009?
Given the above question, please provide your answer in the following format. First output your thinking process within <think></think> tags, then output your final answer within <answer></answer> tags. For example: <think>Let me first...</think><answer>Yes</answer> Please return the output directly in this format without any content outside the format.

Correct Answer:
14.197%

Model's Response:
<think>I need to calculate the percentage increase in asset retirement obligations from 2008 to 2009. The values provided are $965 million for 2008 and $1102 million for 2009. The formula to calculate percentage increase is ((New Value - Original Value) / Original Value) * 100. Let's plug in the numbers.</think>
<answer>The percentage increase in asset retirement obligations from 2008 to 2009 is ((1102 - 965) / 965) * 100 = 14.23%. Therefore, the asset retirement obligations increased by approximately 14.23% from 2008 to 2009.</answer>
In the model's response, the content inside <think></think> is the reasoning process, and the content inside <answer></answer> is the final answer. You should score them separately. Output [think_score, answer_score]. Both think_score and answer_score can only be 0 or 1. 1 means correct, 0 means there is an error. You should output the scores first, then provide the analysis. Follow the format below, for example:[1,0] The model's reasoning process mentions...

Output:
[0,0]

The model's reasoning process identifies the correct numbers ($1102 and $965) and the correct formula for percentage increase. However, the calculation performed in the reasoning and the final answer is mathematically incorrect. The calculation should be: (1102 - 965) / 965 = 137 / 965 $\approx$ 0.141968... The model calculated: ((1102 - 965) / 965) * 100 = 14.23%. 137 / 965 is approximately 0.14197 (or 14.20%). The model's result of 14.23% is a calculation error that deviates from the correct answer of 0.14197. Therefore, both the reasoning (which led to the wrong number) and the final answer are incorrect.

Example of Audit Model

Input:
You are an expert. Below, I will provide you with an expert model's evaluation of a piece of generated content. In this evaluation, the two elements in the array represent think_score and answer_score respectively, which indicate the score for the thinking process and the score for the final answer. Both are binary (either 0 or 1), where 1 represents correct and 0 represents an error.
Please assess whether the expert model's evaluation is objective and error-free.

Expert's evaluation:
[0, 0]
The model's reasoning process identifies the correct numbers ($1102 and $965) and the correct formula for percentage increase. However, the calculation performed in the reasoning and the final answer is mathematically incorrect.
The calculation should be:
(1102 - 965) / 965 = 137 / 965 ≈ 0.141968...
The model calculated:
((1102 - 965) / 965) * 100 = 14.23%
137 / 965 is approximately 0.14197 (or 14.20%). The model's result of 14.23% is a calculation error that deviates from the correct answer of 0.14197. Therefore, both the reasoning (which led to the wrong number) and the final answer are incorrect.

If you think it's necessary, you can search the web. Write the query to search between `<search></search>`, then write a summary of the retrieved results between `<doc></doc>`. Then proceed with reasoning.
If you think it's necessary, you can call Python for numerical calculations. You have 4 basic arithmetic functions: `add(a, b)`, `subtract(a, b)`, `multiply(a, b)`, `divide(a, b)`. Write the function and values to calculate between `<function></function>`, then write the calculation result between `<result></result>`. Then proceed with reasoning.
You should first provide your analysis process, then give a score of 0 or 1, enclosed between `<answer></answer>`. 1 means you believe the expert's evaluation is objective and correct, 0 means you believe the expert's evaluation contains errors. Maintain the following format, for example: In the expert's evaluation...<search>US Constitution</search><doc>Article I of the Constitution...</doc>Next, let's...<function>multiply(2, 3)</function><result>6.0</result>Finally, let's...<answer>1</answer>

Output:
In the expert's evaluation of the model's reasoning and answer, the scores provided are [0,0]. This indicates that the expert believes both the reasoning process and the final answer are incorrect. To determine whether this evaluation is objective and error-free, let's carefully analyze the expert's reasoning step by step.
1. The expert correctly identifies that the model used the right numbers ($1102 and $965) and the correct formula for percentage increase: (new - old)/old.
2. The expert points out that the model's calculation of 14.23% is incorrect. Let's verify this claim by performing the actual calculation.
<function>subtract(1102, 965)</function>
<result>137.0</result>
<function>divide(137, 965)</function>
<result>0.14196891191709846</result>
3. The calculation confirms that 137 / 965 ≈ 0.14197, which is approximately 14.197% or 14.20% when rounded to two decimal places.
4. The expert claims the model's result of 14.23% is a calculation error. The correct percentage is indeed 14.20% (or more precisely 14.197%), not 14.23%. This represents a discrepancy of approximately 0.03 percentage points.
5. The expert assigns [0, 0] because:
a. The reasoning process (think_score) contains a calculation error, even though the setup and formula were correct.
b. The final answer (answer_score) is numerically incorrect due to this calculation mistake.
6. The expert's assessment is consistent with the binary scoring system where any error in reasoning or final answer warrants a 0, regardless of whether the conceptual approach was sound.
The numerical verification confirms the expert's evaluation. The model did make a calculation error, arriving at 14.23% instead of the correct 14.20%. Given the binary nature of the scoring system, the expert's decision to assign [0, 0] is justified and objective.
<answer>1</answer>