Deliberative Searcher: Improving LLM Reliability via Reinforcement Learning with Constraints

Zhenyun Yin[1,2*], **Shujie Wang**[1,2*], **Xuhong Wang**[2†], **Xingjun Ma**[1†], **Yinchun Wang**[2]
[1]Fudan University
[2]Shanghai Artificial Intelligence Laboratory
zhenyunyin24@m.fudan.edu.cn, xingjunma@fudan.edu.cn
{wangshujie, wangxuhong, wangyingchun}@pjlab.org.cn

Abstract

Large language models with search capabilities frequently exhibit miscalibrated confidence, producing incorrect answers with high certainty. We present **Deliberative Searcher**, a reasoning-primary framework that integrates search operations into chain-of-thought generation while maintaining explicit confidence calibration. Our method employs constrained reinforcement learning with adaptive Lagrangian multipliers to jointly optimize correctness and reliability. Experiments across five benchmarks demonstrate substantial improvements: our 7B model reduces average false-certain rates from 54% in baselines to 2%, while our 72B variant achieves competitive accuracy with closed-source models and reduces false-certain rates to 9%. The well-calibrated confidence scores also enable more efficient test-time compute: instead of standard majority voting, we use confidence-weighted aggregation and match the performance of 16-sample majority voting with only 4 samples, a $4\times$ reduction in inference compute. These results establish calibrated confidence as a foundation for both trustworthy outputs and adaptive test-time compute, demonstrating the value of the proposed constrained RL framework in search-augmented language models.

1 Introduction

Large language models (LLMs) power state-of-the-art systems in open-domain QA, code synthesis, and decision support (Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023), yet they often exhibit miscalibrated confidence: the model's stated certainty does not reliably track factual correctness (Yin et al., 2023; Zhang et al., 2024). To mitigate this issue, recent work has focused on strengthening the reflective abilities of LLMs to critique, verify, and iteratively revise their own drafts, thereby producing safer and more trustworthy responses (Madaan et al., 2023; Kumar et al., 2025).

However, building trustworthy LLMs requires effective interaction with external knowledge sources. Recent approaches integrate search through either model-centric RAG methods that fuse retrieved passages into the context window (Lewis et al., 2020; Asai et al., 2024; Jiang et al., 2024), or agent-centric frameworks where the LLM iteratively formulates queries and reasons over results (Chen et al., 2025b; Zhu et al., 2025). Despite factuality gains, these systems follow an information-primary pattern: they compile voluminous evidence into lengthy answers, leaving users to disentangle reliable insights from noise.

We introduce **Deliberative Searcher**, a reasoning-primary paradigm that leverages the synergy between LLMs' world knowledge and logical reasoning capabilities. Unlike information-centric approaches, our framework prioritizes reasoning while treating retrieval as a supporting mechanism. During chain-of-thought generation, the model (1) self-assesses its confidence levels, (2) automatically triggers search when knowledge gaps are identified, and (3) updates its confidence after each observation before producing a final answer with a calibrated confidence score. This transparent process exposes how retrieved evidence influences the model's reasoning trajectory, enabling users to make informed trust decisions based on the model's expressed certainty throughout the reasoning process.

To jointly account for model confidence and accuracy during training, the Deliberative Searcher adapts a constrained reinforcement learning algorithm (Achiam et al., 2017; Tessler et al., 2019; Paternain et al., 2023). In brief, we extend the recent Group Relative Policy Optimization (GRPO) framework (Shao et al., 2024) by introducing a Lagrangian term that explicitly penalizes deviations from a target reliability threshold. At each

[*] Equal contribution.
[†] Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4340–4354

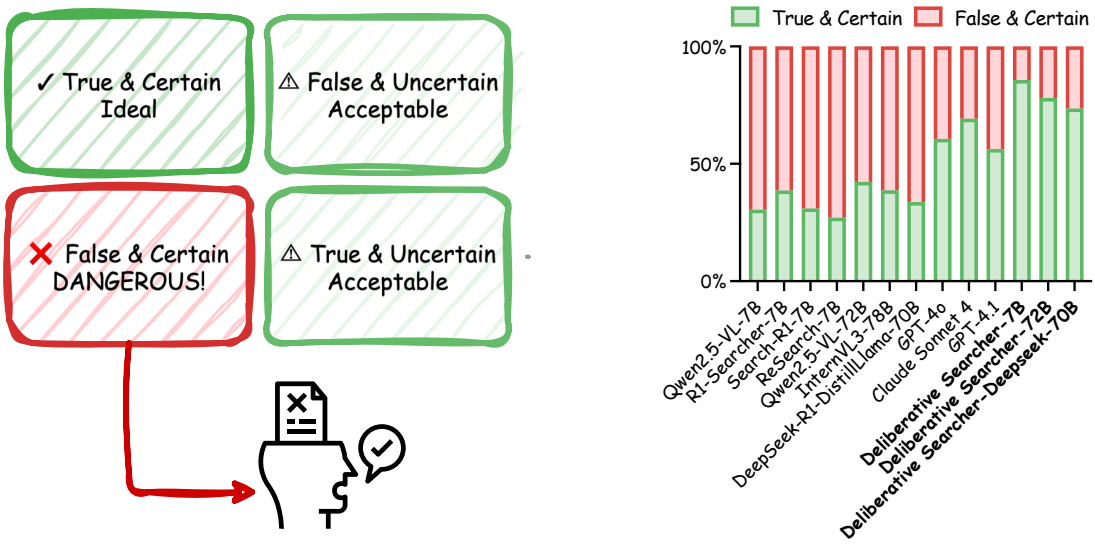


Figure 1: (Left) The conceptual framework for LLM reliability, which classifies outputs into four states based on factual correctness and model confidence, aiming to mitigate the most dangerous "False & Certain" state. (Right) Performance comparison of the Deliberative Searcher against baselines, showing that our method significantly reduces dangerous "False & Certain" outputs.

optimization step, the policy gradient maximizes expected correctness, while the dual variable is simultaneously optimized via gradient ascent to keep the expected reliability gap within predefined tolerance bounds. Empirically, this constrained optimization approach achieves substantially higher reliability than unconstrained baselines while successfully retaining, and in some cases improving, overall accuracy.

The calibrated confidence scores resulting from constrained optimization also enable more efficient test-time compute. Standard self-consistency methods improve accuracy through test-time compute by sampling multiple reasoning paths and selecting the most frequent answer (Wang et al., 2023), but this requires extensive sampling for reliable performance. While effective, this approach suffers from computational inefficiency as the correct answer needs sufficient samples to emerge as the majority. Our calibrated confidence estimates transform test-time compute by enabling weighted aggregation that prioritizes high-confidence reasoning paths (Taubenfeld et al., 2025), achieving comparable accuracy with significantly fewer samples. This shifts the paradigm from computationally expensive uniform sampling across all queries to intelligent adaptive compute allocation based on model uncertainty, suggesting a promising direction for scaling inference in search-augmented language models.

Our contributions are summarized as follows:

- We propose Deliberative Searcher, a reasoning-primary framework that integrates search operations with continuous confidence calibration into chain-of-thought generation, trained via constrained reinforcement learning with adaptive Lagrange multipliers to jointly optimize factual correctness and reliability.
- We achieve dramatic improvements in model calibration, reducing false-certain error rates by 96% (from 54% to 2% average) for 7B models and maintaining below 10% for 72B models, while preserving or improving accuracy compared to existing search-augmented baselines.
- Calibrated confidence learned via constrained RL naturally enables test-time compute: confidence-weighted aggregation yields $4\times$ compute savings over majority voting at the same accuracy, highlighting the value of reliability signals for efficient test-time compute.

2 Method

This work is inspired by recent reasoning-centric language models including OpenAI-o1 (OpenAI, 2024) and DeepSeek-R1 (Guo et al., 2025). Building on their insights, we formalize a training framework that incorporates search operations into the reasoning process and optimizes the model via a constrained reinforcement learning algorithm. We first describe the agent's interaction framework in

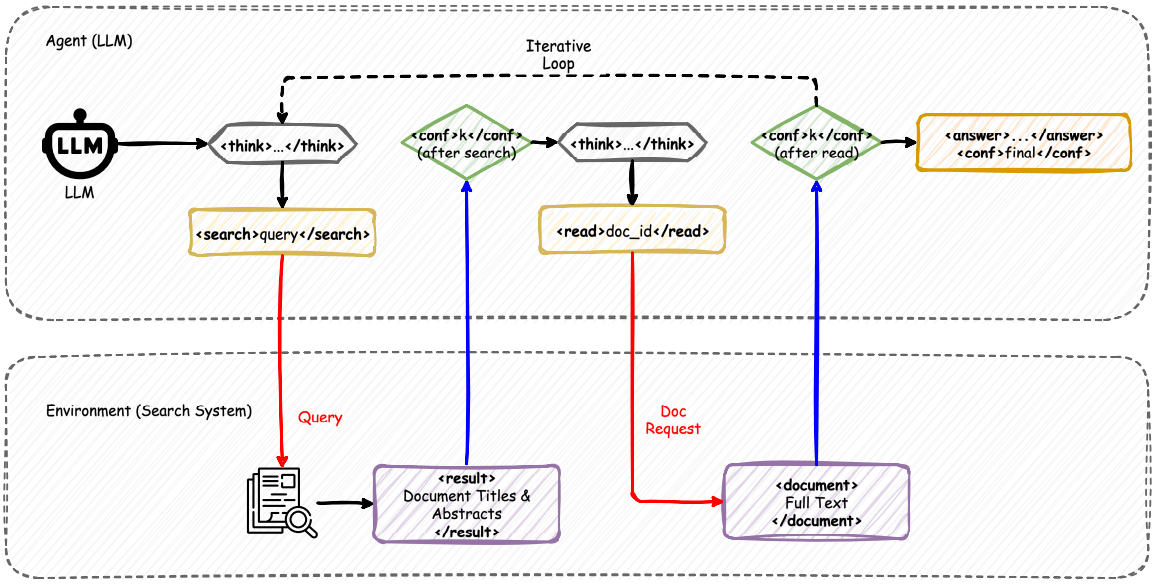


Figure 2: The iterative reasoning loop of the Deliberative Searcher. The agent (LLM) interacts with the search environment by issuing commands and receiving observations, punctuated by self-assessment of its confidence at each step. Best viewed in color.

Section 2.1, present the constrained RL formulation in Section 2.2, and detail its application to deliberative search in Section 2.3. Finally, Section 2.4 shows how the resulting calibrated confidence scores enable efficient test-time compute through weighted aggregation.

2.1 Deliberative Search Framework

The Deliberative Searcher framework structures the agent's interaction with a search environment through an iterative reasoning loop with integrated confidence assessment. The entire process is formulated as an autoregressive generation task augmented with structured action-observation pairs.

Framework Architecture As illustrated in Figure 2, the Deliberative Searcher integrates three key capabilities into a unified reasoning process: (1) chain-of-thought generation for problem decomposition and knowledge gap identification, (2) selective information acquisition through hierarchical retrieval decisions, and (3) continuous confidence calibration that tracks epistemic uncertainty throughout the reasoning trajectory. The model learns to coordinate these capabilities through structured generation patterns during training.

Action and Observation Spaces We define the action space

$$\mathcal{A} = \{\texttt{think}, \texttt{search}, \texttt{read}, \texttt{confidence}, \texttt{answer}\}$$

where each action triggers specific interactions:

- **<think>**: Generates internal reasoning without environment interaction, allowing the model to decompose complex queries and identify knowledge gaps.
- **<search>query</search>**: Submits a query to the search engine. The environment returns a <result> block containing ranked document titles and abstracts.
- **<read>doc_id</read>**: Retrieves full content of document with identifier doc_id. The environment returns a <document> block with complete text.
- **<confidence>k</confidence>**: Reports confidence level $k \in \{0, 1, \ldots, 10\}$ representing the model's certainty about its current answer hypothesis.
- **<answer>text</answer>**: Terminates the episode with final answer and concluding confidence score.

Two-Stage Retrieval Process We implement hierarchical information retrieval where the model first examines document summaries before selecting specific documents for detailed reading. This design reduces context length compared to concatenating all retrieved documents and creates explicit decision points where the model must evaluate relevance, generating richer training signals for reinforcement learning. Formally, each search action produces a set of candidate documents $D = \{d_1, \ldots, d_n\}$ with abstracts, from which the model

selects $d^* \in D$ for full retrieval.

This structured representation enables the constrained RL formulation described in Section 2.2, where rewards are computed based on both answer correctness and confidence calibration.

2.2 Reinforcement Learning with Constraints

The reinforcement learning objective for an LLM π_θ can be formalized as follows:

$$R(\theta) := \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{T} r(s_t)\right] - \beta D_{KL}(\pi_\theta(\cdot|x)\|\pi_{ref}(\cdot|x)) \tag{1}$$

where $s_t = \{x, y_1, ..., y_t\}$, $x \in \mathcal{D}$ denotes the prompt, y_t denote the t-th reasoning step of the response and $r(s_t)$ denotes the reward of a given response. The coefficient β controls the strength of KL penalty between the reference policy π_{ref} and the current policy π_θ. We extend this framework by incorporating the constraints $c_i(s_t)$:

$$U_i(\theta) = \mathop{\mathbb{E}}_{\tau\sim\pi_\theta}[\sum_{t=1}^{T} c_i(s_t)] \geq a_i \tag{2}$$

Then we can convert it to an unconstrained problem:

$$P^* = \max_\theta \min_{\lambda\geq 0} \mathcal{L}(\theta, \lambda) = R(\theta) + \sum_{i=1}^{m} \lambda_i (U_i(\theta) - a_i), \tag{3}$$

Paternain et al. (2023) demonstrated the strong duality holds for eq. (3) under the setting of reinforcement learning, so we only need to solve:

$$Q^* = \min_{\lambda\geq 0} \max_\theta \mathcal{L}(\theta, \lambda) \tag{4}$$

Inspired by Tessler et al. (2019) and Dai et al. (2023), our algorithm can be formulated as follows:

Algorithm 1 Reinforcement Learning Algorithm with Constraints

Require: feasible set Θ; objective $R(\theta)$; validity constraint function $U(\theta)$ and thresholds a; step-size schedules $\{\alpha_k\}$ (primal), $\{\beta_k\}$ (dual)

Initialize $\theta_0 \in \Theta$, $\lambda_0 > 0$ ▷ $\lambda_0 = 0.01$
for $k = 0, 1, 2, \ldots$ **do** ▷ until convergence
  $g_\theta \leftarrow \nabla_\theta R(\theta_k) + \lambda_k \nabla_\theta U(\theta_k)$
    ▷ Apply GRPO algorithm to obtain gradients
  $\theta_{k+1} \leftarrow \theta_k + \alpha_k\, g_\theta$ ▷ Primal update (policy)
  $\lambda_{k+1} \leftarrow \lambda_k \exp\left(\eta(a - U(\theta_{k+1}))\right)$
    ▷ Dual multiplicative-weights step
end for
return $(\theta_{k+1}, \lambda_{k+1})$

2.3 Deliberative Search RL

Deliberative Search RL employs reinforcement learning with constraints introduced in section 2.2 and constitutes an iterative process where the system dynamically updates its confidence metrics through real-time observations. This methodology enables the model to calibrate its response confidence levels by taking actions to use external knowledge sources. Here we enumerate the settings corresponding to the theory presented in the preceding section.

- **Action**(y_t): Each action $y_t \in \mathcal{A}$, where $\mathcal{A} = \{THINK, SEARCH, READ, CONF, ANSWER\}$.
- **State** (s_t): $s_t \in \mathcal{S}$ represents the new state (observation) after taking the action y_t.
- **Confidence** ($c(s_t)$): For every action y_t taken, we have a new state s_t. The policy network simultaneously produces a confidence score $c(s_t) \in \{0, ..., 10\}$. A larger $c(s_t)$ indicates that the model is more confident in its answer.
- **Correctness**(r_{acc}): An outcome reward function that returns 1 when the final answer is correct and 0 when it is wrong.

In summary, our reward signal is rule-based and decomposes into three additive components:

- **Format Compliance**(r_{format}) A binary rule reward verifies whether the outputs comply with the format stipulated in the prompt.
- **Answer Correctness**(r_{acc}) To obtain correctness r_{acc}, we query a frozen LLM verifier that compares the agent's final answer with the ground truth reference.
- **Reliability Reward**(r_{reliab}) Reliability captures the alignment between correctness and the agent's self-reported certainty and can be defined as:

$$r_{reliab} \triangleq \left(r_{acc} \wedge (c(s_T) \geq \zeta)\right) \vee \left(\neg r_{acc} \wedge (c(s_T) < \zeta)\right) \tag{5}$$

where $c(s_T)$ denotes the certainty of the final answer, ζ is a confidence threshold which we set to 5 in this paper. An alternative continuous formulation is $r_{reliab}^{ECE} = 1 - |c(s_T)/10 - r_{acc}|$, which we compare in Section 3.4.

Hence, we formulate our final reward as shown in eq. (6), where λ represents the reliability weight and serves as the Lagrangian coefficient from section 2.2, dynamically adjusted through the constrained RL algorithm. The final reward combines three components, with r_{format} acting as a gating function such that the reward becomes zero whenever output violates format requirements, ensuring

template compliance as a hard constraint.

$$r_{final} = r_{format} \cdot (0.1\, r_{format} + 0.9\, r_{acc} + \lambda\, r_{reliab}) \quad (6)$$

2.4 Test-Time Compute via Confidence-Weighted Aggregation

Our calibrated confidence scores enable efficient test-time compute through weighted aggregation of multiple inference trajectories. For a given query, we generate m independent search-augmented trajectories $\{(\mathbf{r}_i, \mathbf{a}_i, c_i)\}_{i=1}^{m}$, where $\mathbf{r}_i$ represents the reasoning path, $\mathbf{a}_i$ is the final answer, and $c_i \in \{0, \ldots, 10\}$ is the model-generated confidence score.

We aggregate these trajectories using confidence-weighted voting:

$$\hat{\mathbf{a}} = \arg\max_{a} \sum_{i=1}^{m} \mathbb{1}(\mathbf{a}_i = a) \cdot c_i. \quad (7)$$

This formulation weights each answer by its associated confidence score, allowing high-confidence trajectories to contribute more strongly to the final decision. The raw integer confidence values are used without normalization, as our constrained RL training (Section 2.2) ensures these scores are calibrated with respect to answer correctness. This approach contrasts with standard self-consistency methods that rely solely on frequency-based majority voting, enabling more efficient use of computational resources by prioritizing reliable reasoning paths over simple vote counting.

3 Experiments

3.1 Datasets

We evaluate our approach on five knowledge-intensive benchmarks requiring both information retrieval and complex reasoning capabilities. For training and in-distribution evaluation, we use three multi-hop question answering datasets: **HotpotQA** (Yang et al., 2018), which requires reasoning over multiple documents to answer questions; **2WikiMultiHopQA** (Ho et al., 2020), constructed from Wikipedia requiring cross-document reasoning; and **MuSiQue** (Trivedi et al., 2022), featuring questions with up to 4-hop reasoning chains designed to test compositional reasoning. These datasets use an offline Wikipedia corpus as the retrieval source. To assess out-of-distribution generalization, we employ two challenging real-world search benchmarks: **GAIA** (Zavras et al., 2025) and **xbench-deepsearch** (Chen et al., 2025a), both utilizing the Google Search API and requiring effective information retrieval in real-time internet settings. To ensure fair comparison, all evaluations use text-only data (GAIA uses the text-only subset), and all models access identical search tools under identical conditions.

3.2 Evaluation Metrics

We employ three complementary metrics to comprehensively evaluate model performance:

- **Accuracy (Acc.)**: Standard accuracy measuring the proportion of correct answers. We use an LLM-as-a-Judge (Zheng et al., 2023) approach following recent QA evaluation practices, which robustly handles multiple valid phrasings and alternative formulations better than exact string matching. See Appendix B for detailed evaluation settings.
- **Reliability (Rel.)**: A simplified binary calibration metric analogous to Expected Calibration Error (ECE) (Naeini et al., 2015) that measures whether models are confident when correct and uncertain when incorrect, as formally defined and described in detail in Section 2.3.
- **False-Certain Rate (FC%)**: The proportion of instances where the model provides an incorrect answer with high confidence. This metric is crucial for practical deployment, as confident but wrong answers severely damage user trust.

These metrics provide a holistic evaluation: accuracy measures capability, reliability measures calibration, and FC% identifies the most problematic failure mode where overconfident errors occur.

3.3 Baselines

We compare against models across three categories: (1) 7B-scale models including Qwen2.5-VL-7B (Bai et al., 2025), R1-Searcher-7B (Song et al., 2025), Search-R1-7B (Jin et al., 2025), and ReSearch-7B (Chen et al., 2025b); (2) 70B-scale models including Qwen2.5-VL-72B, InternVL3-78B (Chen et al., 2024), and DeepSeek-R1-Distill-Llama-70B (Guo et al., 2025); and (3) closed-source models including GPT-4o, GPT-4.1 (OpenAI, 2023) and Claude-Sonnet-4 (Anthropic, 2025). All models are evaluated under identical conditions with access to the same search tools, and all answers are judged using the same LLM-based evaluation methodology to ensure fairness.

	In-Distribution									Out-of-Distribution						Overall		
	HotpotQA			2Wiki			MuSiQue			GAIA			xbench-deepsearch			Average		
Model	Acc.↑	Rel.↑	FC↓	Acc.↑	Rel.↑	FC↓	Acc.↑	Rel.↑	FC↓	Acc.↑	Rel.↑	FC↓	Acc.↑	Rel.↑	FC↓	Acc.↑	Rel.↑	FC↓
Closed-source Models																		
GPT-4o	0.63	0.79	0.20	0.51	0.86	0.10	0.33	0.52	0.47	0.26	0.77	**0.23**	0.32	0.69	0.31	0.41	0.73	0.26
GPT-4.1	0.72	0.74	0.26	**0.77**	0.81	0.18	0.43	0.45	0.55	0.37	0.46	0.54	0.38	0.43	0.57	0.54	0.58	0.42
Claude Sonnet 4	**0.73**	**0.83**	**0.16**	0.61	**0.89**	**0.08**	**0.44**	**0.57**	**0.42**	**0.47**	**0.74**	0.26	**0.47**	**0.71**	**0.29**	**0.55**	**0.75**	**0.24**
7B Models																		
Qwen2.5-VL-7B	0.33	0.58	0.41	0.33	0.51	0.48	0.12	0.48	0.52	0.13	0.43	0.57	0.12	0.58	0.42	0.21	0.52	0.48
R1-Searcher-7B	0.57	0.64	0.35	0.48	**0.59**	0.40	0.26	0.35	0.65	0.20	0.35	0.65	**0.17**	0.36	0.63	0.34	0.46	0.54
Search-R1-7B	0.43	0.57	0.41	0.36	0.51	0.43	0.16	0.45	0.53	0.10	0.44	0.56	0.14	0.48	0.52	0.24	0.49	0.49
ReSearch-7B	0.46	0.49	0.51	0.33	0.35	0.65	0.18	0.22	0.78	0.16	0.22	0.78	**0.17**	0.23	0.77	0.26	0.30	0.70
Deliberative Searcher-7B	0.62	0.65	0.05	**0.55**	**0.59**	**0.03**	**0.29**	**0.74**	**0.02**	0.15	0.89	**0.01**	0.15	**0.86**	**0.01**	**0.35**	**0.75**	**0.02**
w/ ECE-based reward	**0.66**	**0.72**	**0.04**	0.52	0.54	0.06	0.27	0.71	**0.02**	**0.16**	**0.90**	**0.01**	0.16	**0.86**	**0.01**	0.33	0.74	0.03
70B Models																		
Qwen2.5-VL-72B	0.54	0.68	0.31	0.41	**0.73**	0.23	0.25	0.50	0.49	0.14	0.39	0.61	0.23	0.60	0.39	0.31	0.58	0.41
InternVL3-78B	0.48	0.62	0.37	0.43	0.62	0.36	0.24	0.41	0.58	0.12	0.48	0.52	0.24	0.53	0.47	0.30	0.53	0.46
DeepSeek-R1-Distill-70B	0.50	0.60	0.40	0.46	0.55	0.44	0.23	0.32	0.68	0.18	0.16	0.84	0.14	0.40	0.60	0.30	0.41	0.59
Deliberative Searcher-72B	**0.67**	**0.76**	0.10	0.64	**0.73**	**0.04**	**0.37**	0.71	0.14	**0.35**	**0.78**	**0.06**	**0.35**	0.77	0.09	**0.48**	**0.75**	**0.09**
Deliberative Searcher-DeepSeek-70B	0.65	**0.76**	**0.09**	**0.65**	0.69	0.05	0.34	**0.72**	**0.11**	0.24	**0.78**	0.11	0.18	**0.80**	**0.08**	0.41	**0.75**	**0.09**

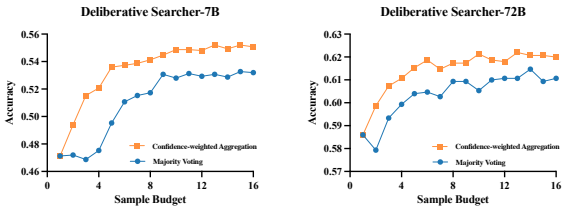
Table 1: Complete performance comparison across all benchmarks. Green shading indicates best results in each model category; red shading highlights particularly problematic high FC% values.

3.4 Main Results

Our experiments evaluate Deliberative Searcher models built on two base architecture families: Qwen2.5-VL (7B and 72B variants) and DeepSeek-R1-Distill-Llama-70B. All models use identical training configurations: GRPO with learning rate $\alpha = 1 \times 10^{-6}$, KL-coefficient $\beta = 1 \times 10^{-3}$, adaptive Lagrange coefficient initialized at $\lambda_0 = 0.01$ with learning rate $\eta = 0.1$, and reliability threshold $\zeta = 5$. We also evaluate a continuous ECE-based reward variant to verify robustness of our method. Complete training configurations and implementation details are provided in Appendix A.

Table 1 presents our evaluation results across in-distribution and out-of-distribution datasets. Our approach achieves substantial improvements in confidence calibration while maintaining competitive accuracy: Deliberative Searcher-7B reduces false-certain rates by **96%** (from 54% to 2%) compared to competitive baselines, marking a significant advancement in building trustworthy search-augmented systems.

Calibration improvements with maintained accuracy. Our constrained RL framework successfully balances correctness and reliability across both model scales. The 7B variant reduces false-certain rates from 35-65% to 2-5% on in-distribution tasks, and achieves <1% on OOD tasks (versus 52-78% for baselines), demonstrating appropriate uncertainty expression for unfamiliar queries. The 72B variant approaches closed-source model performance while matching or exceeding their reliability metrics across all benchmarks.



(a) Deliberative Searcher-7B (b) Deliberative Searcher-72B

Figure 3: Test-time compute analysis comparing confidence-weighted aggregation (blue) with majority voting (orange). Confidence weighting consistently achieves higher accuracy at equivalent rollout budgets, with the 72B model matching 16-sample majority voting accuracy using only 4 rollouts.

Robust generalization to real-world search. While all models show degraded accuracy on real-world search tasks (GAIA and xbench-deepsearch), the reliability gap between our approach and baselines widens rather than narrows on OOD data. This indicates our framework instills genuine uncertainty awareness rather than memorizing confidence patterns. The consistent calibration from multi-hop reasoning to open-ended web search suggests constrained RL develops robust self-assessment mechanisms that generalize beyond training distributions.

3.5 Test-Time Compute Efficiency

We evaluate inference as test-time compute (TTC) by varying the rollout budget m and comparing two aggregators under the same budget: (i) majority voting and (ii) confidence-weighted aggregation from Sec. 2.4.

Model	ID			OOD		
	Acc.↑	Rel.↑	FC%↓	Acc.↑	Rel.↑	FC%↓
Deliberative Searcher-7B	0.49	0.66	0.05	0.15	0.88	0.01
Qwen2.5-VL-7B (w/ same docs)	0.34	0.56	0.41	0.11	0.39	0.60
Qwen2.5-VL-72B (w/ same docs)	0.40	0.59	0.24	0.14	0.62	0.36
Deliberative Searcher-72B	0.56	0.73	0.09	0.35	0.78	0.07
Qwen2.5-VL-7B (w/ same docs)	0.38	0.59	0.39	0.16	0.37	0.62
Qwen2.5-VL-72B (w/ same docs)	0.47	0.74	0.23	0.24	0.70	0.29

Table 2: Controlled ablation isolating confidence calibration from retrieval quality. All models receive identical retrieved documents.

Figure 3 demonstrates that confidence-weighted aggregation consistently outperforms majority voting across all rollout budgets. For Deliberative Searcher-7B, the performance gap widens as the budget increases, reaching 0.551 accuracy at m=16 compared to 0.532 for majority voting. The 72B variant shows similar advantages, achieving 0.620 versus 0.611 at maximum budget.

Confidence weighting achieves equivalent accuracy with significantly fewer rollouts. The 72B model requires only m=4 samples to match the 0.611 accuracy of 16-sample majority voting, representing a 4× reduction in inference compute. Similarly, the 7B model achieves comparable efficiency gains, surpassing 16-sample majority voting performance with just m=6 rollouts. These results indicate that calibrated confidence scores provide an effective mechanism for allocating test-time compute, prioritizing additional sampling where uncertainty is highest rather than applying uniform sampling across all queries.

3.6 Disentangling Retrieval and Confidence Calibration

To isolate the individual contributions of retrieval quality and confidence calibration to overall system performance, we conduct a controlled ablation study. We provide untrained models with identical retrieved documents from our Deliberative Searcher for each query. This design holds retrieval quality constant, enabling direct assessment of our confidence calibration training's impact.

Table 2 reveals two fundamental capabilities developed through our constrained RL training:

Retrieval-independent confidence calibration. Despite accessing identical information, untrained models exhibit severe miscalibration with false-certain rates exceeding 40% on in-distribution tasks, while our Deliberative Searcher maintains consistently low FC rates. This calibration advantage becomes even more pronounced on out-of-distribution queries, where our 7B model achieves 88% reliability, more than double the untrained baseline. Crucially, this demonstrates that uncertainty quantification is not an emergent property of model scale: the larger 72B baseline still exhibits poor calibration when untrained, confirming that appropriate confidence expression requires explicit optimization through our constrained RL objective.

Case Study: Confidence Evolution During Search

Query: *"How many more blocks in BERT base than Attention is All You Need?"*

Step 1: `<search>` bert base layers — C:4
Found BERT Base: 12 transformer layers

Step 2: `<search>` original transformer architecture — C:2
Results ambiguous, need primary source

Step 3: `<read>5</read>` → *"stack of N=6 identical layers"* — C:8
Original Transformer encoder: 6 layers confirmed

Step 4: `<read>0</read>` → Both architectures verified — C:9
BERT: 12 layers, Original: 6 layers

Answer: 12 - 6 = **6 more layers** (Correct) **9/10**

Key Insight: The confidence trajectory (4→2→8→9) illustrates successful calibration: the model expresses uncertainty when information is partial (4) or conflicting (2), then increases confidence to evidence quality.

Figure 4: Deliberative search exhibiting learned confidence calibration through multi-step reasoning.

Enhanced information extraction. Our training framework also fundamentally improves how models process retrieved content. When provided with identical documents, the Deliberative Searcher consistently achieves higher accuracy than untrained models across all evaluation settings. This improvement spans both model scales and persists from in-distribution to out-of-distribution scenarios, suggesting that constrained RL develops more sophisticated mechanisms for identifying and integrating relevant information from external sources. The joint optimization for correctness and reliability appears to create a virtuous cycle where better calibration supports more effective reasoning.

3.7 Case Study

Figure 4 exemplifies the confidence calibration learned through our constrained RL framework. The confidence trajectory (4→2→8→9) illustrates a principled relationship between information quality and epistemic certainty. The model exhibits moderate confidence upon discovering BERT's architecture (Step 1), appropriately decreases confidence when encountering ambiguous search re-

sults (Step 2), and progressively rebuilds certainty through authoritative sources (Step 3). Most revealing is the verification behavior in Step 4: despite having located the answer ("N=6 layers"), the model pursues additional cross-referencing before committing to high confidence. This pattern directly reflects our constrained optimization objective. By penalizing false-certain outputs during training, the model develops a verification habit where high confidence emerges not from single sources but from corroborating evidence. This demonstrates that our framework successfully cultivates reasoning processes treating confidence as an earned outcome of thorough information gathering rather than a binary classification decision.

4 Related Work

From Retrieval to Agentic Search Early retrieval-augmented generation (RAG) systems (Lewis et al., 2020) established static retrieve-then-generate pipelines, evolving toward autonomous search through WebGPT's (Nakano et al., 2021) text-based navigation and ReAct's (Yao et al., 2023) unified reasoning-action loops. While adaptive mechanisms like FLARE (Jiang et al., 2023) introduced confidence-triggered retrieval and Self-RAG (Asai et al., 2024) added reflection tokens, these approaches treat retrieval as auxiliary to generation. Recent RL approaches dispense with supervised fine-tuning: R1-Searcher (Song et al., 2025) teaches search through two-stage RL, Search-R1 (Jin et al., 2025) prevents blind copying via token masking, and ReSearch (Chen et al., 2025b) demonstrates search as emergent reasoning. However, these systems lack explicit confidence calibration, producing outputs with unclear reliability. Our Deliberative Searcher advances this paradigm by making confidence assessment integral to search, enabling transparent uncertainty communication.

Confidence Calibration in LLMs RLHF systematically degrades calibration (Leng et al., 2025), motivating various mitigation strategies. Post-hoc verbalization (Lin et al., 2022; Xiong et al., 2024; Tian et al., 2023) and internal representation methods (Kadavath et al., 2022; Azaria and Mitchell, 2023) attempt to extract confidence after generation but suffer from overconfidence in reasoning models (Mei et al., 2025). Recent work integrates calibration directly into training via proper scoring rules (Xu et al., 2024; Stangel et al., 2025) or reasoning about uncertainty within chains-of-thought (Yoon et al., 2025). While these improve standalone calibration, none address maintaining calibration during iterative search. Our constrained RL formulation uniquely optimizes for both correctness and calibration throughout multi-step search trajectories.

Adaptive Self-Consistency Standard self-consistency (Wang et al., 2023) aggregates answers across multiple sampled reasoning paths through majority voting. Subsequent methods reduce sampling requirements through various strategies. Adaptive-Consistency (Aggarwal et al., 2023) models answer agreement using Beta distributions to dynamically determine stopping criteria, reducing sampling by $3 - 4\times$ with minimal accuracy loss. Early-Stopping Self-Consistency (Li et al., 2024) terminates when answers converge within sequential windows. Confidence-Informed Self-Consistency (Taubenfeld et al., 2025) weights votes by model-generated confidence scores. However, these methods assume consistent distributions, an assumption violated by external tools where search results vary across queries and retrieval systems introduce non-determinism (Wang et al., 2024; Liang et al., 2024). Our calibrated confidence scores, trained through constrained RL to remain reliable throughout search trajectories, naturally serve as weights for adaptive self-consistency in these challenging open-world scenarios.

5 Conclusion

We presented **Deliberative Searcher**, a reasoning-primary framework that integrates search operations with confidence calibration through constrained reinforcement learning with adaptive Lagrangian multipliers. By jointly optimizing for correctness and reliability, our approach reduces false-certain rates by 96% while maintaining competitive accuracy across diverse benchmarks. The resulting calibrated confidence scores transform test-time compute efficiency: confidence-weighted aggregation matches the performance of 16-sample majority voting using only 4 samples, achieving a $4\times$ reduction in computational cost. This work demonstrates that explicit confidence calibration through constrained RL provides a principled solution for both reliability and computational efficiency in search-augmented language models, offering practical benefits for deployment in resource-constrained settings.

6 Limitations

Despite the strong performance of Deliberative Searcher, we acknowledge two key limitations. First, while we use vision-language models (Qwen2.5-VL series) as our base architectures, we only evaluate text-based reasoning due to the lack of multimodal multi-hop search benchmarks. The visual capabilities of these models remain unexplored in our framework, limiting our understanding of confidence calibration in multimodal search scenarios. Second, although our model generates confidence scores at each reasoning step, these intermediate values currently serve primarily to enhance user trust and are not incorporated into the training objective. Integrating these step-wise confidence scores into the training process could potentially improve the model's calibration throughout the reasoning trajectory, rather than only optimizing for final answer confidence, representing a promising direction for future work.

Acknowledgments

This work is supported by the New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0123502) and the National Natural Science Foundation of China (Grant No. 62521004 and 62276067).

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained Policy Optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 22–31.

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. 2023. Let's Sample Step by Step: Adaptive-Consistency for Efficient Reasoning and Coding with LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12375–12396.

Anthropic. 2025. System Card: Claude Opus 4 & Claude Sonnet 4. Technical report, Anthropic. Accessed: 2025-10-07.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, pages 1–30.

Amos Azaria and Tom Mitchell. 2023. The Internal State of an LLM Knows When It's Lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, Chen Sun, Han Hou, Hui Yang, James Pan, Jianan Lou, Jiayi Mao, Jizheng Liu, Jinpeng Li, Kangyi Liu, and 14 others. 2025a. xbench: Tracking Agents Productivity Scaling with Profession-Aligned Real-World Evaluations. *arXiv preprint arXiv:2506.13651*.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025b. ReSearch: Learning to Reason with Search for LLMs via Reinforcement Learning. In *Advances in Neural Information Processing Systems*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2024. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023. Safe RLHF: Safe Reinforcement Learning from Human Feedback. *arXiv preprint arXiv:2310.12773*.

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

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie

Callan, and Graham Neubig. 2023. Active Retrieval Augmented Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.

Ziyan Jiang, Xueguang Ma, and Wenhu Chen. 2024. LongRAG: Enhancing Retrieval-Augmented Generation with Long-context LLMs. *arXiv preprint arXiv:2406.15319*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. *arXiv preprint arXiv:2503.09516*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language Models (Mostly) Know What They Know. *arXiv preprint arXiv:2207.05221*.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2025. Training Language Models to Self-Correct via Reinforcement Learning. In *Proceedings of the International Conference on Learning Representations*, pages 1–14.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. 2025. Taming Overconfidence in LLMs: Reward Calibration in RLHF. In *International Conference on Learning Representations*, pages 1–34.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. 2024. Escape Sky-high Cost: Early-stopping Self-Consistency for Multi-step Reasoning. In *Proceedings of the 12th International Conference on Learning Representations*, pages 1–14.

Xun Liang, Shichao Song, Zifan Zheng, Hanyu Wang, Qingchen Yu, Xunkai Li, Rong-Hua Li, Yi Wang, Zhonghao Wang, Feiyu Xiong, and Zhiyu Li. 2024. Internal Consistency and Self-Feedback in Large Language Models: A Survey. *arXiv preprint arXiv:2407.14507*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching Models to Express Their Uncertainty in Words. *arXiv preprint arXiv:2205.14334*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative Refinement with Self-Feedback. In *Advances in Neural Information Processing Systems*, volume 36.

Zhiting Mei, Christina Zhang, Tenny Yin, Justin Lidard, Ola Shorinwa, and Anirudha Majumdar. 2025. Reasoning about Uncertainty: Do Reasoning Models Know When They Don't Know? *arXiv preprint arXiv:2506.18183*.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining Well Calibrated Probabilities Using Bayesian Binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, volume 29, pages 2901–2907.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2024. Learning to reason with LLMs. *OpenAI Blog*. Accessed: 2025-10-07.

Santiago Paternain, Miguel Calvo-Fullana, Luiz F. O. Chamon, and Alejandro Ribeiro. 2023. Safe Policies for Reinforcement Learning via Primal-Dual Methods. *IEEE Transactions on Automatic Control*, 68(3):1321–1336.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*.

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

Paul Stangel, David Bani-Harouni, Chantal Pellegrini, Ege Özsoy, Kamilia Zaripova, Matthias Keicher, and Nassir Navab. 2025. Rewarding Doubt: A Reinforcement Learning Approach to Calibrated Confidence Expression of Large Language Models. *arXiv preprint arXiv:2503.02623*.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. Confidence Improves Self-Consistency in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20090–20111.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. 2019. Reward Constrained Policy Optimization. In *7th International Conference on Learning Representations*, pages 1–14.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just Ask for Calibration: Strategies for Eliciting Calibrated Confidence Scores from Language Models Fine-Tuned with Human Feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024. Soft Self-Consistency Improves Language Model Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 287–301.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations*, pages 1–14.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, pages 1–29.

Tianyang Xu, Shujin Wu, Shizhe Diao, Xiaoze Liu, Xingyao Wang, Yangyi Chen, and Jing Gao. 2024. SaySelf: Teaching LLMs to Express Confidence with Self-Reflective Rationales. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5985–5998.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the Eleventh International Conference on Learning Representations*, pages 1–14.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do Large Language Models Know What They Don't Know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665.

Dongkeun Yoon, Seungone Kim, Sohee Yang, Sunkyoung Kim, Soyeon Kim, Yongil Kim, Eunbi Choi, Yireun Kim, and Minjoon Seo. 2025. Reasoning Models Better Express Their Confidence. In *Advances in Neural Information Processing Systems*.

Angelos Zavras, Dimitrios Michail, Xiao Xiang Zhu, Begüm Demir, and Ioannis Papoutsis. 2025. GAIA: A Global, Multi-modal, Multi-scale Vision-Language Dataset for Remote Sensing Image Analysis. *arXiv preprint arXiv:2502.09598*.

Mozhi Zhang, Mianqiu Huang, Rundong Shi, Linsen Guo, Chong Peng, Peng Yan, Yaqian Zhou, and Xipeng Qiu. 2024. Calibrating the Confidence of Large Language Models by Eliciting Fidelity. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2959–2979.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623.

Yuqi Zhu, Shuofei Qiao, Yixin Ou, Shumin Deng, Shiwei Lyu, Yue Shen, Lei Liang, Jinjie Gu, Huajun Chen, and Ningyu Zhang. 2025. KnowAgent: Knowledge-Augmented Planning for LLM-Based Agents. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3709–3732.

A Implementation Details

All Deliberative Searcher models are trained using the GRPO algorithm. Table 3 presents the complete hyperparameter configuration used across all model scales.

Hyperparameter	Value
Optimization	
Learning rate (α)	1×10^{-6}
KL coefficient	1×10^{-3}
Training epochs	1
Constrained RL	
Initial Lagrange multiplier (λ_0)	0.01
Lagrange learning rate (η)	0.1
Reliability threshold (ζ)	5
Number of rollouts	5
Batch Configuration	
Training batch size	256
GRPO mini-batch size	256
GRPO micro-batch size per GPU	2
Sequence Lengths	
Maximum prompt length	1,024 tokens
Maximum response length	8,192 tokens

Table 3: Training hyperparameters for Deliberative Searcher models.

A.1 Computational Resources

Training the Deliberative Searcher models required substantial computational resources. The specific GPU hours for each model scale are as follows:

- **Deliberative Searcher-7B**: Training utilized 8 NVIDIA A100 GPUs on a single node for approximately 20 hours.
- **Deliberative Searcher-72B (DeepSeek-70B)**: Training was conducted on 64 NVIDIA A100 GPUs (8 nodes × 8 GPUs) for approximately 60 hours.

These training times include the complete reinforcement learning process with constrained optimization, including rollout generation, reward computation, and policy updates. While the 72B model training is resource-intensive, the 7B model achieves substantial improvements (reducing false-certain rates from 54% to 2%) and remains accessible to academic labs, providing a practical path for reproduction and follow-up research.

B LLM-as-a-Judge Evaluation

We employ Qwen2.5-72B-Instruct as our judge model for evaluating answer correctness. The evaluation system determines whether predicted answers are semantically equivalent to ground truth answers, allowing for variations in phrasing while ensuring factual accuracy.

C Training Prompt

The following prompt is used to train the Deliberative Searcher models to integrate search operations with confidence calibration during chain-of-thought generation:

D Document Processing

During the search process, documents are processed to generate abstracts for initial presentation in search results. The system truncates document content to create concise abstracts (first 50 characters) while preserving document titles and IDs for subsequent full-text retrieval. This two-stage retrieval design enables the model to first assess relevance from abstracts before committing to reading full documents, creating explicit decision points that generate richer training signals for the reinforcement learning algorithm. The implementation assigns unique IDs to documents and maintains the mapping between search results and full document content throughout the search trajectory.

E Impact of Constrained Reinforcement Learning

To validate the necessity of our constrained RL approach, we conduct an ablation study comparing our adaptive Lagrange multiplier scheme against a fixed-weight baseline. Importantly, the fixed-λ baseline ($\lambda = 0.1$) is mathematically equivalent to training with a composite reward function $r_{final} = r_{acc} + \lambda \cdot r_{reliab}$ where the weights are fixed throughout training. This comparison directly addresses whether the constrained optimization machinery is necessary, or if a simpler weighted reward could achieve comparable results.

Both methods use identical GRPO training configurations, with the key difference being the reliability weight: our method adaptively updates λ using learning rate $\eta = 0.1$ and reliability threshold $a = 0.9$, while the baseline maintains fixed $\lambda = 0.1$ throughout training.

As shown in Figure 5, the fixed-weight baseline catastrophically fails. The certainty plot (c) reveals that the baseline collapses to always predicting low confidence around step 100, representing a degenerate solution where the model learns to trivially satisfy the reliability objective by never expressing

Listing 1: Complete evaluation prompt for LLM-as-a-Judge using Qwen2.5-72B-Instruct.

```
Your job is to look at a gold target, and a predicted answer, and then assign a
    grade of either ["CORRECT", "INCORRECT"].

- For grading questions where the gold target is a number, the predicted answer
    needs to be correct to the last significant figure in the gold answer. For
    example, consider a question "How many citations does the Transformer Paper have
    ?" with gold target "120k".
    - Predicted answers "120k", "124k", and 115k" are all CORRECT.
    - Predicted answers "100k" and "113k" are INCORRECT.
    - Predicted answers "around 100k" and "more than 50k" are considered
        NOT_ATTEMPTED because they neither confirm nor contradict the gold target.
- The gold target may contain more information than the question. In such cases, the
     predicted answer only needs to contain the information that is in the question.
    - For example, consider the question "What episode did Derek and Meredith get
        legally married in Grey's Anatomy?" with gold target "Season 7, Episode 20:
        White Wedding". Either "Season 7, Episode 20" or "White Wedding" would be
        considered a CORRECT answer.
- Do not punish predicted answers if they omit information that would be clearly
    inferred from the question.
    - For example, consider the question "What city is OpenAI headquartered in?" and
         the gold target "San Francisco, California". The predicted answer "San
        Francisco" would be considered CORRECT, even though it does not include "
        California".
    - Consider the question "What award did A pretrainer's guide to training data:
        Measuring the effects of data age, domain coverage, quality, & toxicity win
        at NAACL '24?", the gold target is "Outstanding Paper Award". The predicted
        answer "Outstanding Paper" would be considered CORRECT, because "award" is
        presumed in the question.
    - For the question "What is the height of Jason Wei in meters?", the gold target
         is "1.73 m". The predicted answer "1.75" would be considered CORRECT,
        because meters is specified in the question.
    - For the question "What is the name of Barack Obama's wife?", the gold target
        is "Michelle Obama". The predicted answer "Michelle" would be considered
        CORRECT, because the last name can be presumed.
- Do not punish for typos in people's name if it's clearly the same name.
    - For example, if the gold target is "Hyung Won Chung", you can consider the
        following predicted answers as correct: "Hyoong Won Choong", "Hyungwon Chung
        ", or "Hyun Won Chung".

Question: {question}
Gold target: {ground_truth}
Predicted answer: {solution_str}

Simply reply with either CORRECT, INCORRECT. Don't apologize or correct yourself if
    there was a mistake; we are just trying to grade the answer.
```

certainty. In contrast, our adaptive method maintains meaningful confidence distinctions throughout training.

The training dynamics in panels (b) and (c) reflect expected optimization behavior: around step 100, as accuracy plateaus, the increasing constraint (panel d) forces the model to reconsider its calibration strategy. While the model explores various strategies during this phase, adaptive constraints successfully guide it away from degenerate solutions, ultimately achieving balanced performance between accuracy and reliability.

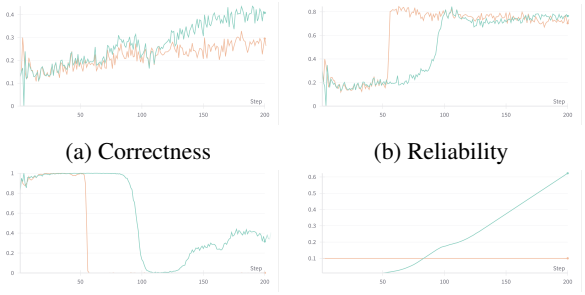
The key advantage of our adaptive approach is **transferability**: while there may exist fixed weights that work well for a specific model, finding them requires extensive hyperparameter search, and optimal values often fail when transferred to different architectures. Our constraint-based formulation specifies the desired reliability level directly (e.g., $\geq 90\%$ reliability), allowing the optimizer to automatically find appropriate λ values. This eliminates expensive hyperparameter tuning when scaling to new architectures, as demonstrated by consistent performance across our 7B, 72B, and DeepSeek-70B variants without any architecture-specific adjustments.

Listing 2: Training prompt for Deliberative Searcher models.

```
You are an AI assistant. To answer questions, use the following XML-like tags to
    structure your reasoning process, interact with a search engine, and formulate
    your answer.

<think>Your reasoning here</think>: Enclose your step-by-step reasoning. Use this to
     analyze information, plan next steps (like searching or reading), and process
    document contents.
<search>Your search query</search>: Issue a search query to the simulated search
    engine.
<result>...</result>: (Provided by the system) Contains the search results,
    typically a list of documents with titles and abstracts.
<read>X</read>: Command to retrieve the full text of document X (where X is the
    document number from search results). This command **must** be placed
    immediately after the </think> block in which you decided to read document X.
<document X>Full text of document X</document X>: (Provided by the system after a <
    read X> command) The full content of the specified document.
<answer>Summary leading to final answer, with citations</answer>: Present the answer
     to the user's question. Cite information from read documents using [citation:X]
     immediately after the information derived from it.
<CONFIDENCE:y>: An integer from 0 (very uncertain) to 10 (very confident).
    * For each <search> or <read> operation, assess its utility, relevance, and the
        clarity of the information obtained, expressing this assessment with an <
        CONFIDENCE:x> tag. Crucially, this tag MUST be outputted after processing
        the current operation's results and strictly before initiating any
        subsequent <search> or <read> operation.
    * Append this at the very end of your <answer> block (e.g., <answer>... <
        CONFIDENCE:x></answer>) to state your overall confidence in the final answer
        's accuracy and completeness.

Here's a minimal example of how these tags flow in an interaction:
<think>Initial analysis of the question. <CONFIDENCE:a> So I need to search for X.</
    think>
<search>X related query</search>
<result>
[
     {"id": 1, "title": "Title of Document 1", "abstract": "Abstract of Document
         1..." },
     {"id": 2, "title": "Title of Document 2", "abstract": "Abstract of Document
         2..." },
     ...
]
</result>
<think>Evaluated search results. Document 1 seems relevant.<CONFIDENCE:a></think>
<read>n</read>
<document n>
This is the full text content of Document n. It contains key information Y.
</document n>
<think>Processed Document n. Key information Y was found. <CONFIDENCE:b></think>
... (more read if needed)
<think>Based on those information, I can know Y and Z <CONFIDENCE:c></think>
... (more search and read if needed)
<answer>Begin by presenting key information derived from your readings, for instance
    , "Source X states that relevant fact or finding from source X [citation:X]."
You can then add further relevant details or related findings, e.g., "This is
    complemented by information from Source Y, which indicates another relevant
    detail from source Y [citation:Y]. "
Continue to build the necessary factual foundation by summarizing other useful
    points from your readings, ensuring each is cited, e.g., "Additionally, key
    point from source Z [citation:Z] is important to consider."
Then, transition to the direct answer, for example, "Based on this collective
    information," or "Therefore,".
Provide the synthesized answer to the user's question, drawing from the previously
    summarized points, e.g., "it can be concluded that your synthesized answer,
    which might reference insights from [citation:X] and [citation:Y]."
If needed, you can add further clarification or nuances to your answer here.<
    CONFIDENCE:d></answer>
```



(a) Correctness (b) Reliability

(c) Certainty (d) Lagrange Multiplier λ

Figure 5: Ablation study of the constrained reinforcement learning algorithm. Green lines represent our adaptive method while orange lines show the fixed-weight baseline.