CAS: Conformalized Agentic Search via Adaptive Retrieval and Policy Weighting

Zixi Zhu[1*], Jiayuan Su[3*], Jian Zhang[1], Yu Lin[1†], Hongwei Wang[12†]
[1]ZJU-UIUC Institute, Zhejiang University.
[2]State Key Laboratory of CAD&CG, Zhejiang University.
[3]Tencent Inc.
{zixizhu, 12221038, hongweiwang}@zju.edu.cn
yulin@intl.zju.edu.cn
matt.jiayuan.su@gmail.com

Abstract

Search Agents face a severe reliability crisis during reinforcement learning (RL) fine-tuning. Heuristic Top-K retrieval often causes critical evidence loss or noise inclusion, while overconfidence induced by progressive RL leads to hallucinated answers and redundant searches. To build highly reliable agents, we introduce Conformal Prediction (CP) and propose Conformalized Agentic Search (CAS). This framework establishes reliability guarantees on both the retrieval and training sides: on the retrieval side, an Adaptive Prediction Set (APS), a specific CP realization, translates statistical coverage into dynamic document truncation to construct prediction sets that are adaptive in size; on the training side, Adaptive Conformal Inference (ACI), a dynamic CP algorithm, dynamically constructs prediction sets with controllable coverage to quantify answer confidence, which is then used to penalize low-confidence trajectories within the Group Relative Policy Optimization (GRPO) objective, ensuring the model learns only from reliable ones. Experiments across single-hop and multi-hop QA datasets demonstrate that our framework significantly improves reasoning accuracy while drastically reducing redundant tool invocations, establishing a highly reliable and efficient agent paradigm. Our code is available at https://github.com/S1llyBird/CAS.

1 Introduction

Large Language Models (LLMs) have significantly advanced complex problem-solving by integrating external knowledge (Schick et al., 2023; Liang et al., 2026). While traditional Retrieval-Augmented Generation (RAG) employs a static "retrieve-then-generate" paradigm (Lewis et al., 2021), the recent emergence of Agentic Search offers a more dynamic and autonomous approach (Jin et al., 2025; He et al., 2026; Zhao et al., 2025). By interleaving internal reasoning steps with external information-gathering actions within a continuous generation trajectory, the agent autonomously plans when to retrieve and how to integrate newly acquired knowledge (Yao et al., 2023).

However, fine-tuning these agents via reinforcement learning (RL) carries inherent risks of unreliability. First, on the retrieval side, heuristic Top-K truncation is inherently unreliable. Given varying query difficulties, a fixed K inevitably leads to the omission of critical facts or the inclusion of distracting noise (Liu et al., 2023). Second, on the training side, LLMs often exhibit overconfidence as RL progresses (Leng et al., 2025), leading the model to generate hallucinated responses. Without effective confidence constraints, Search Agents are easily trapped in a cycle of highly inefficient, redundant tool invocations (Yin et al., 2026).

To address these risks, we adopt Conformal Prediction (CP) (Angelopoulos and Bates, 2022), a principled statistical framework that quantifies model uncertainty with rigorous theoretical guarantees. Unlike heuristic methods, CP provides strong finite-sample coverage guarantees: given a user-specified error rate α, it constructs a prediction set that satisfies a target coverage level of at least $1-\alpha$. These properties make CP a theoretically grounded choice for establishing reliability in autonomous systems. Building on this, we propose Conformalized Agentic Search (CAS), which simultaneously applies CP to both the retrieval and training sides to provide rigorous statistical guarantees (Kumar et al., 2023; Su et al., 2025).

On the retrieval side, we implement CP via the Adaptive Prediction Set (APS) method (Romano et al., 2020). While ensuring a strict marginal coverage guarantee, APS dynamically adjusts the size of the set of retrieved items based on the model's "uncertainty" regarding the current input (Chakraborty et al., 2026). On the training side, to mitigate the

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

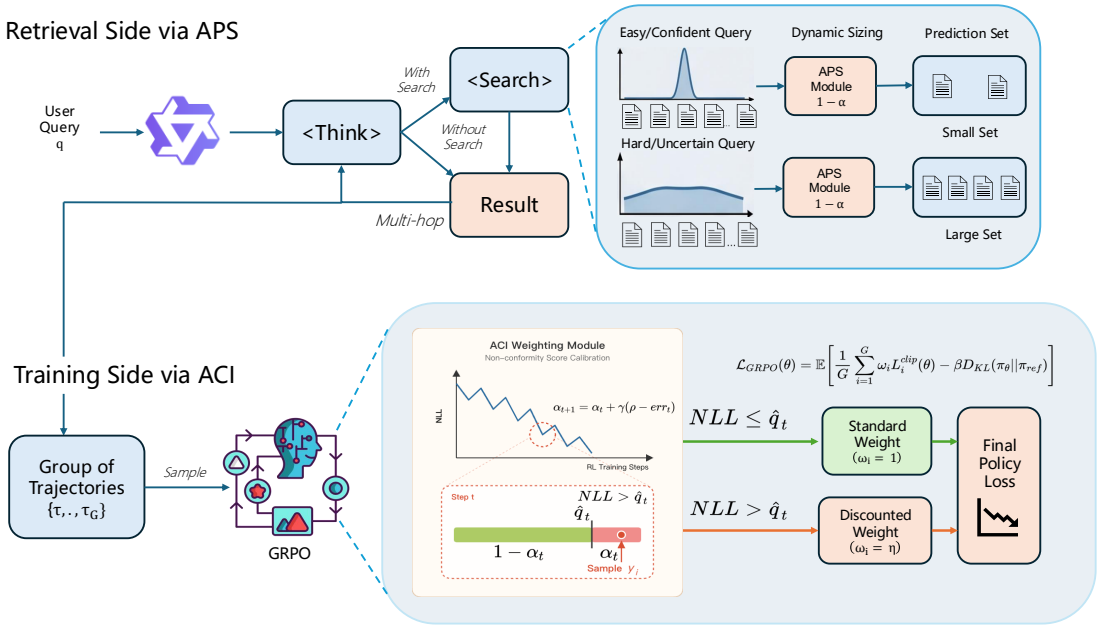


Figure 1: **The CAS framework. (A)** APS dynamically sizes the retrieved document set based on local query difficulty. **(B)** ACI modulates the GRPO policy loss by penalizing low-confidence trajectories (NLL $> q_{1-\alpha_t}$) to enforce reliable policy optimization.

severe calibration degradation and overconfidence inherent in standard RL, we utilize Adaptive Conformal Inference (ACI) (Gibbs and Candès, 2021), a dynamic CP algorithm. ACI dynamically constructs prediction sets with controllable coverage to quantify answer reliability during training. We optimize the Group Relative Policy Optimization (GRPO) (Shao et al., 2024) process by penalizing low-confidence trajectories (both blind overconfidence and erratic underconfidence), ensuring the model learns only from highly reliable reasoning paths.

In summary, our core contributions are threefold:

- A Reliable Theoretical Framework: We propose CAS, the first framework to introduce CP into the RL fine-tuning of search agents, providing rigorous statistical guarantees for reasoning and retrieval reliability.
- CP Constraints on Both Sides: We implement APS on the retrieval side to construct prediction sets that are adaptive in size, ensuring the inclusion of correct evidence. Concurrently, we apply ACI on the training side to optimize the GRPO process, mitigating low-confidence outputs.
- Superior Accuracy and Efficiency: Extensive experiments across single-hop and multi-hop QA datasets demonstrate that our framework significantly improves reasoning accuracy and training stability while drastically reducing redundant tool invocations, yielding a highly reliable and efficient agent paradigm.

2 Conformal Prediction

CP (Angelopoulos and Bates, 2022) is a principled statistical framework that quantifies uncertainty with rigorous coverage guarantees, regardless of the underlying model or data distribution. Central to CP is a non-conformity score function $s(x, y)$, which measures the "unusualness" of a candidate output y given an input x.

Let (X, Y) be a sample, where X represents the input and Y represents the output. Suppose we have a calibration set of n samples, denoted as $(X_i, Y_i)_{i=1}^n$, and a test sample (X_{test}, Y_{test}) drawn independently and identically (i.i.d.) from the same underlying distribution. Given a user-specified target error rate $\alpha \in (0, 1)$, CP computes a quantile threshold $\hat{q}$ corresponding to the $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$ empirical quantile of the calibration scores. It then constructs a prediction set $\mathcal{C}_{1-\alpha}(X_{test})$ defined as:

$$\mathcal{C}_{1-\alpha}(X_{test}) = \{y \in \mathcal{Y} : s(X_{test}, y) \leq \hat{q}\} \quad (1)$$

Under the i.i.d. assumption, this procedure formally guarantees marginal coverage: $P(Y_{test} \in \mathcal{C}_{1-\alpha}(X_{test})) \geq 1 - \alpha$. See Appendix B.1 for the formal proof.

2.1 Adaptive Prediction Set

As a specific realization of the CP framework for classification and generative tasks, APS (Romano et al., 2020) constructs prediction sets by defining a specialized non-conformity score. Specifically, given sorted predictive probabilities $\pi_{(1)}(x) \geq \cdots \geq \pi_{(n)}(x)$, APS defines the non-conformity score $s(x, y)$ as the cumulative mass up to the true label y: $s = \sum_{j=1}^{L(y)} \pi_{(j)}(x)$, where $L(y)$ denotes the rank of y. During the CP inference phase, APS identifies the minimum index k to form the prediction set such that the cumulative probability exceeds the calibrated threshold $\hat{q}$:

$$\sum_{i=1}^{k} \pi_{(i)}(x) \geq \hat{q} \tag{2}$$

This mechanism adaptively yields compact sets for confident inputs and expanded sets for ambiguous ones, strictly maintaining the $1 - \alpha$ CP coverage guarantee while asymptotically approximating conditional coverage. See Appendix B.2 for the formal proof and discussions on its conditional coverage.

2.2 Adaptive Conformal Inference

While the standard CP framework fundamentally relies on the exchangeability (i.i.d.) assumption, ACI (Gibbs and Candès, 2021) is a dynamic extension designed to handle data streams where the underlying data distribution may change over time. Instead of maintaining a static target error rate, ACI introduces a time-varying error parameter α_t.

At each time step t, we observe a test point (X_t, Y_t), where X_t is the input and Y_t is the true response. The algorithm evaluates whether Y_t was contained within the previous prediction set via the empirical miscoverage indicator:

$$err_t := \begin{cases} 1, & \text{if } Y_t \notin \mathcal{C}_t(\alpha_t) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $\mathcal{C}_t(\alpha_t) := \{y \in \mathcal{Y} : s(X_t, y) \leq \hat{Q}_t(1 - \alpha_t)\}$ is the dynamic prediction set, and $\hat{Q}_t(\cdot)$ is the empirical quantile function.

Given a long-term target error rate ρ and a step size $\gamma > 0$, ACI updates the error parameter via a simple online rule:

$$\alpha_{t+1} = \alpha_t + \gamma(\rho - err_t) \tag{4}$$

This recursive mechanism acts as a feedback loop: miscoverage ($err_t = 1$) decreases α_t, thereby expanding the subsequent prediction set to be more conservative. Conversely, success ($err_t = 0$) increases α_t, tightening the set. By continuously adapting α_t, ACI preserves valid uncertainty quantification even when the data distribution changes over time. The formal proof of this dynamic guarantee is provided in Appendix B.3.

3 Methodology

We present the CAS framework (Figure 1) to reliably bridge LLMs' internal reasoning with external retrieval. We introduce two synergistic modules for statistical reliability: APS to bound dynamic retrieval uncertainty, and ACI to penalize low-confidence trajectories during policy optimization.

3.1 Overview of Agentic Search

Our policy model $\pi_\theta(y|x)$ employs a strict generative grammar to interleave internal reasoning with external actions. Given an input x, the model initiates reasoning within `<think>...</think>` tags.Upon reaching a knowledge boundary, it emits a query q enclosed in `<search>...</search>` tags,, halting generation to invoke an external search engine $\mathcal{S}$. Crucially, rather than using a fixed top-k, the raw retrieved evidence $D = \mathcal{S}(q)$ is dynamically truncated into a reliable subset D_{APS} via an APS. This subset is then wrapped in `<information>...</information>` tags and appended to the context. This generation-retrieval cycle repeats until the model outputs its final prediction a_{pred} inside `<answer>...</answer>` tags. The complete prompt template is provided in Table 5.

3.2 Retrieval Side via APS

Traditional tool-use frameworks typically append a fixed number of top-k results to the context, which often injects redundant noise or truncates critical information. To rigorously bound the uncertainty of external evidence, we frame our dynamic retrieval mechanism within the general CP paradigm (Eq. (1)). However, while standard CP methods guarantee marginal coverage across the data distribution,

they fail to guarantee conditional coverage—often failing to adapt to the specific difficulty of a given input.

To heuristically approximate conditional coverage, we specify α_{APS} and construct a reliable document subset by implementing the APS detailed in Section 2.1, thereby obtaining the calibrated threshold $\hat{q}_{APS}$. Given a query q, the search engine $\mathcal{S}$ returns an initial candidate set $D = \{d_1, d_2, \ldots, d_n\}$ accompanied by their raw retrieval scores. We normalize these scores using a softmax operation to obtain a relevance probability $p(d_i|q)$ for each document. Following the rigorous APS inference procedure, our system identifies the truncation index k by accumulating these probabilities until the mass first exceeds the calibrated threshold $\hat{q}_{\text{APS}}$. By dynamically adjusting to the conditional probability distribution of q, this process yields a statistically guaranteed subset $D_{\text{APS}} = \{d_1, \ldots, d_k\}$. Crucially, this provides a statistical guarantee that ensures a $1 - \alpha_{APS}$ coverage rate while adapting the context length to the difficulty of q.

3.3 Reward Design

In our RL framework, we train the policy π_θ using a rule-based reward $r(x, y) = r_{acc} + r_{fmt}$. We define $r_{acc} = EM(a_{pred}, a_{gold}) \in \{0, 1\}$ as the exact match indicator. To enforce structural integrity, $r_{fmt}(y)$ incorporates two Boolean indicators, $\mathbb{I}_{\text{valid}}$ and $\mathbb{I}_{\text{ans}}$, representing strict grammatical correctness and the successful generation of the `<answer>` boundary, respectively. With a scaling factor $\gamma = 0.2$, the format reward is formulated as:

$$r_{\text{fmt}}(y) = \gamma \cdot \Big[- r_{\text{acc}}(1 - \mathbb{I}_{\text{valid}}) + (1 - r_{\text{acc}}) \left(\mathbb{I}_{\text{valid}} + \frac{1}{2}\mathbb{I}_{\text{ans}}(1 - \mathbb{I}_{\text{valid}}) \right) \Big] \quad (5)$$

This formulation rewards the correct format: when the answer is correct ($r_{\text{acc}} = 1$), it strictly applies a $-\gamma$ penalty for format violations to prevent reward hacking. Conversely, when the answer is incorrect ($r_{\text{acc}} = 0$), it provides a dense intermediate signal (γ for full validity, or $\frac{1}{2}\gamma$ for partial structural effort) to guide the model toward using the correct format.

3.4 Training Side via ACI

Standard CP relies on the strict exchangeability (i.i.d.) assumption. However, during RL, the policy π_θ continuously evolves, and the prevalent use of binary rewards often leads to model overconfidence. To mitigate the redundant invocations and hallucinated outputs caused by this overconfidence, we employ ACI.

In our framework, at each RL iteration t, for the i-th sampled trajectory y_i given input x_t, we define its non-conformity score $s(x_t, y_i)$ as the Negative Log-Likelihood (NLL) (Quach et al., 2024) of the generated tokens strictly within the `<answer>...</answer>` tags. Let $\mathcal{M}_i$ denote the set of token indices corresponding to these final answer tokens. The score is formulated as:

$$s(x_t, y_i) = \frac{\sum_{j \in \mathcal{M}_i} (-\log \pi_\theta(y_j | y_{<j}, x_t))}{|\mathcal{M}_i|} \quad (6)$$

Given a pre-specified target error rate ρ, instead of using the standard update rule, we employ a smoothed ACI update mechanism over previously observed data to mitigate local variations in the error rate (Gibbs and Candès, 2021). Specifically, we update the error rate α_t by evaluating the recent empirical miscoverage frequency using an exponentially weighted moving average of past errors:

$$\alpha_{t+1} = \alpha_t + \gamma \left(\rho - \sum_{s=1}^{t} v_s err_s \right) \quad (7)$$

where $\{v_s\}_{1 \le s \le t}$ is a sequence of increasing weights such that $\sum_{s=1}^{t} v_s = 1$. In practice, we define the temporal weights with a smoothing factor of 0.95 as

$$v_s := \frac{0.95^{t-s}}{\sum_{s'=1}^{t} 0.95^{t-s'}}.$$

This approach effectively produces smoother trajectories for α_t with less local variation. The updated α_t is then used to dynamically adjust the quantile threshold $\hat{q}_t = \hat{Q}_t(1 - \alpha_t)$. Specifically, the empirical quantile function $\hat{Q}_t(\cdot)$ is evaluated over a rolling calibration window of past scores from iteration $r = \max(1, t - 2000)$ to $t - 1$. This threshold $\hat{q}_t$ is subsequently employed to partition low-confidence trajectories.

We employ GRPO (Shao et al., 2024) for the RL training. For a given input x_t at iteration t, the policy samples a group of G trajectories $\{y_1, y_2, \ldots, y_G\}$. GRPO optimizes the policy by computing the relative advantage A_i for each trajectory y_i, which is obtained by normalizing its reward R_i within the group: $A_i = \frac{R_i - \text{mean}(\mathbf{R})}{\text{std}(\mathbf{R})}$.

Concurrently, to penalize low-confidence sequences generated during training, we introduce a

discount factor $\eta \in (0, 1)$ for samples falling into the low-confidence set. Accordingly, we minimize the ACI-guided GRPO loss function as follows:

$$\mathcal{L}_{GRPO}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\omega_i L_i^{clip}(\theta) - \beta D_{KL}(\pi_\theta||\pi_{ref})\right] \tag{8}$$

where βD_{KL} is the KL divergence penalty against the reference model π_{ref}, and $L_i^{clip}(\theta)$ is the standard GRPO clipped objective function driven by the advantage A_i. The dynamic confidence-based weight ω_i is defined as:

$$\omega_i = \begin{cases} 1, & \text{if } s(x_t, y_i) \leq \hat{q}_t \\ \eta, & \text{otherwise} \end{cases} \tag{9}$$

Crucially, this weighting mechanism synergizes with the RL advantage A_i to provide a dual-constraint on model reliability. For unconfident lucky guesses ($s(x_t, y_i) > \hat{q}_t$ with $A_i > 0$), the positive reinforcement is discounted by η, preventing the model from learning to guess. Conversely, if the model is confidently incorrect ($s(x_t, y_i) \leq \hat{q}_t$ but yielding $A_i < 0$), the full weight ($\omega_i = 1$) ensures the model receives the maximum penalty.

To effectively penalize low-confidence samples during early training while preventing over-penalization of relatively high-confidence trajectories classified as low-confidence after the model converges, we set the discount factor to $\eta = 0.5$. The complete procedure of CAS is formally presented in Algorithm 1.

4 Experiments

4.1 Experimental Setup

Datasets. To comprehensively evaluate CAS, we conduct experiments on seven diverse open-domain Question Answering (QA) datasets covering both single-hop and complex reasoning capabilities: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023), HotpotQA (Yang et al., 2018), 2WikiMultihopQA (2Wiki) (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). For training our RL framework, we construct a mixed training corpus using the training splits of NQ and HotpotQA.

Baselines. We compare our method against a comprehensive suite of competitive baselines, which can be logically categorized into four groups: inference without retrieval, including Direct Inference and CoT reasoning (Wei et al., 2023); inference with retrieval, comprising standard RAG (Lewis et al., 2021), IRCoT (Trivedi et al., 2023), and Search-o1 (Li et al., 2025); fine-tuning based methods, which involve Supervised Fine-Tuning (SFT) (Chung et al., 2022), RL-based fine-tuning without search (R1) (Guo et al., 2025), and rejection sampling with a search engine (Ahn et al., 2024); and finally, our reference backbone, Search-R1 (Jin et al., 2025), along with Search-R2 (He et al., 2026).

Implementation Details. We initialize our policy model using Qwen2.5-3B (Qwen et al., 2025) and Qwen3-8B (Yang et al., 2025). For the retrieval module, we utilize the dense retriever E5 (Wang et al., 2024), paired with the 2018 Wikipedia dump (Karpukhin et al., 2020) as the external knowledge base. During the GRPO training phase, we set the group rollout size to $G = 5$ and sample 512 prompts per training step. To prevent infinite generation, the maximum number of assistant-search interaction rounds is capped at 4. The learning rate is set to 1×10^{-6}. For the ACI module, the step size is empirically set to $\gamma = 0.005$. All models are evaluated using the Exact Match (EM) metric. We provide more details in Appendix C.

Conformal Settings. We set $\alpha_{\text{APS}} = 0.2$ and the target error rate $\rho = 0.25$ for ACI. The ACI calibration set $\mathcal{C}$ comprises 150 randomly sampled queries from a mixture of NQ and HotpotQA training splits. For ACI, we evaluate the untrained backbone on $\mathcal{C}$ to compute the initial non-conformity scores (Eq. (6)), establishing the base empirical quantile q_{α_0} to bootstrap the dynamic tracking process. For APS calibration, we construct a distinct calibration set $\mathcal{C}_{\text{APS}}$. We utilize DeepSeek-V3.2 (DeepSeek-AI et al., 2025) to decompose multi-hop queries from the original 150 samples. These decomposed sub-queries, alongside the original single-hop questions, form $\mathcal{C}_{\text{APS}}$, which totals 239 queries. Given the target $\alpha_{\text{APS}} = 0.2$, this sample size provides a statistical error margin of $\epsilon = 0.026$ for the actual coverage (Angelopoulos and Bates, 2022). DeepSeek-V3.2 then acts as the judge to locate the ground-truth documents within $\mathcal{C}_{\text{APS}}$, yielding the calibrated threshold $\hat{q}_{\text{APS}}$ as formulated in Section 3.2.

4.2 Main Results

Table 1 and Table 2 present the comprehensive evaluation results of our method against all baselines across the seven datasets. We summarize the key findings as follows:

Table 1: The main results on seven datasets. †/⋆ represents in-domain/out-of-domain datasets. The best and second best performances are set as bold and underlined, respectively.

Methods	General QA			Multi-Hop QA				Average
	NQ†	TriviaQA⋆	PopQA⋆	HotpotQA†	2WikiMultiHopQA⋆	Musique⋆	Bamboogle⋆	
Direct Inference	0.106	0.288	0.108	0.149	0.244	0.020	0.024	0.134
CoT	0.023	0.032	0.005	0.021	0.021	0.002	0.000	0.015
RAG	0.348	0.544	0.387	0.255	0.226	0.047	0.080	0.270
IRCoT	0.111	0.312	0.200	0.164	0.171	0.067	0.240	0.181
Search-o1	0.238	0.472	0.262	0.221	0.218	0.054	0.320	0.255
SFT	0.249	0.292	0.104	0.186	0.248	0.044	0.112	0.176
R1-base	0.226	0.455	0.173	0.201	0.268	0.055	0.224	0.229
R1-instruct	0.210	0.449	0.171	0.208	0.275	0.060	0.192	0.224
Rejection Sampling	0.294	0.488	0.332	0.240	0.233	0.059	0.210	0.265
Search-R1 (Qwen2.5-3B-Instruct)	0.397	0.565	0.391	0.331	0.310	0.124	0.232	0.336
Search-R1 (Qwen3-8B)	0.440	0.631	0.418	0.372	0.355	0.157	0.430	0.400
Search-R2 (Qwen3-8B)[1]	<u>0.477</u>	**0.676**	**0.466**	<u>0.412</u>	<u>0.405</u>	<u>0.172</u>	<u>0.512</u>	<u>0.446</u>
Ours (Qwen2.5-3B-Instruct)	0.441	0.607	0.447	0.407	0.385	0.169	0.352	0.401
Ours (Qwen3-8B)	**0.490**	<u>0.668</u>	<u>0.465</u>	**0.463**	**0.431**	**0.213**	**0.520**	**0.464**

Table 2: Performance improvements of our method compared to Search-R1 and Search-R2 on General QA (single-hop) and Multi-Hop QA. Δ denotes the absolute performance gain.

Methods	General QA	Multi-Hop QA	Overall
Qwen2.5-3B-Instruct			
Search-R1	0.451	0.249	0.336
Ours	**0.498**	**0.328**	**0.401**
Improvement (Δ)	*+0.047*	***+0.079***	*+0.065*
Qwen3-8B			
Search-R1	0.496	0.329	0.400
Search-R2	0.540	0.375	0.446
Ours	**0.541**	**0.407**	**0.464**
Improvement (Δ) vs. Search-R1	*+0.045*	***+0.078***	*+0.064*
Improvement (Δ) vs. Search-R2	*+0.001*	*+0.032*	*+0.018*

CAS achieves superior performance across all evaluated settings. Specifically, on the Qwen3-8B backbone, our method achieves the highest overall average score of 0.464, outperforming the strong baseline Search-R2 (0.446) and substantially surpassing Search-R1 (0.400). A similar trend is observed on the Qwen2.5-3B-Instruct backbone, where our method achieves an average score of 0.401, improving upon Search-R1 by an absolute margin of +0.065. This consistent superiority across different model scales highlights the generalizability and robustness of CAS.

Our approach demonstrates exceptional performance across both complex multi-step and straightforward single-hop scenarios. As shown in Table 2, on Multi-Hop QA datasets, our method yields massive gains, outperforming Search-R1 by +0.079 (3B) and +0.078 (8B), and surpassing the highly optimized Search-R2 by +0.032. On General QA (single-hop) tasks, our models also achieve highly competitive accuracy, with the 8B backbone significantly outperforming Search-R1 (+0.045) and slightly edging out Search-R2 (+0.001). This demonstrates the improvements of our framework: APS provides a retrieval set with marginal coverage and approximate conditional coverage to prevent the model from encountering hallucinations due to excessive context in simple queries or missing answers in complex ones; meanwhile, ACI effectively ensures high-confidence model outputs, preventing hallucinations and redundant tool invocations. Further experimental analysis regarding the Qwen3-8B backbone is deferred to Appendix E.

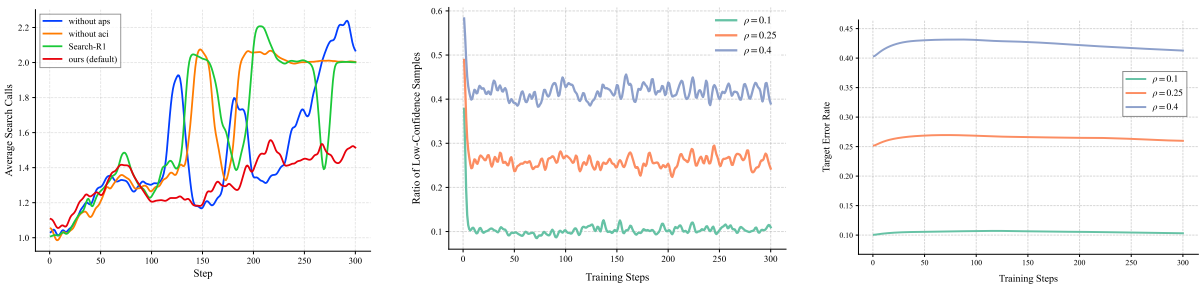
[1]As Search-R2 is closed-source, we are unable to evaluate it on Qwen2.5-3B-Instruct.

Table 3: Ablation study on Qwen2.5-3B-Instruct. The table presents the unablated framework, component-wise ablations, and sensitivity analyses for ρ and α_{APS}.Full results across all individual datasets are detailed in Table 8.

Methods	General QA	Multi-Hop QA	Overall
Ours (Default)	0.498	0.328	0.401
-ACI	0.481	0.311	0.384
-APS	0.490	0.313	0.389
$\rho = 0.1$	0.497	0.302	0.386
$\rho = 0.4$	0.495	0.273	0.368
$\alpha_{\text{APS}} = 0.35$	0.476	0.294	0.372
$\alpha_{\text{APS}} = 0.05$	0.504	0.234	0.350

4.3 Ablation Study

To evaluate the individual contributions of our proposed modules, we conducted a component-wise ablation study on the Qwen2.5-3B-Instruct backbone. For the configuration where the APS is disabled (-APS), the retrieval mechanism falls back to



(a) Average Search Calls (b) Ratio of Low-Confidence Samples (c) Adaptive Target Error Rate (α_t)

Figure 2: Training dynamics and sensitivity analyses on the Qwen2.5-3B-Instruct backbone. **(a)** Evolution of average search calls during training across different ablation configurations. **(b)** The ratio of low-confidence samples penalized by the ACI weight under varying target error rates (ρ). **(c)** The dynamic adaptation of the target error rate (α_t), demonstrating stable convergence to the preset ρ values.

a fixed top-k ($k = 3$) setting. As shown in Table 3, removing either component leads to a notable degradation in both General QA and Multi-Hop QA tasks.

Impact of the ACI Weight. Removing the ACI weight (-ACI) decreases the overall score from 0.401 to 0.384. This decline is intrinsically linked to the model's search behavior. As illustrated in the tool usage trajectories (Figure 2(a)), the baseline and the -ACI variant exhibit significantly higher and more fluctuating tool calls. Without confidence constraints, blind overconfidence causes the model to hallucinate, initiating searches that deviate from the target question. Notably, some trajectories, due to a lack of confidence, conversely resort to secondary searches to verify answers. In summary, through the ACI constraint, the model avoids not only overconfidence but also blind underconfidence, thereby maintaining a stable and efficient search frequency (as denoted by the default trajectory).

Impact of the Adaptive Prediction Set. Disabling APS (-APS) drops accuracy to 0.389 by restricting retrieval to a fixed-length context. This rigid setup degrades performance via two paths: in simple queries, fixed top-k retrieval introduces noise through redundant documents; in complex multi-hop queries, the static window often misses critical facts. Notably, the resulting information scarcity in complex scenarios forces the model to issue additional tool calls to compensate, even with the ACI weight active. Consequently, its tool usage frequency falls between our full framework and the baseline (Figure 2a). This dynamic corroborates the complementarity of the two modules: APS provides an adaptive, noise-free context in a single step, while the ACI weight suppresses unnecessary exploratory searches.

4.4 Sensitivity Analysis

To verify the robustness and controllability of our framework, we conduct a sensitivity analysis on the ACI target error rate $\rho \in \{0.1, 0.25, 0.4\}$ and the APS significance level $\alpha_{\text{APS}} \in \{0.05, 0.2, 0.35\}$. All experiments in this section are performed on the Qwen2.5-3B-Instruct backbone, with calibration set configurations consistent with the Conformal Settings. The performance results are summarized in the bottom sections of Table 3.

ACI Weight under Different ρ. Figures 2b and 2c illustrate the dynamic characteristics of the ACI mechanism during RL fine-tuning. The ACI weight effectively maintains the ratio of low-confidence samples within expected ranges and ensures that the dynamically adjusted α_t closely tracks the target error rate ρ. Notably, a pronounced spike is observed in the ratio of low-confidence samples at the very first step (figure 2b). This phenomenon is directly attributed to the surge of trajectories as the policy π_θ begins to update, introducing highly non-i.i.d. data into the stream. The rapid stabilization of α_t following this shock demonstrates ACI's robust adaptability, highlighting the fundamental inadequacy of Static CP in dynamic RL environments.

Furthermore, as shown in Table 3, a strict target ($\rho = 0.4$) classifies nearly 40% of the reasoning trajectories as low-confidence. While enforcing rigorous quality constraints, this over-penalization severely dilutes the RL reward signals, diminishing training efficiency. Conversely, a relaxed target

($\rho = 0.1$) applies the ACI weight to only 10% of the samples. With such lenient filtering, the performance degenerates toward the unconstrained baseline due to insufficient confidence guidance.

Table 4: Average number of retrieved documents under different APS significance levels (α_{APS}).

α_{APS}	Avg. Retrieved Documents
0.20 (Default)	3.4
0.35	2.4
0.05	4.8

APS Retrieval under Different α_{APS}. The significance level α_{APS} dictates the aggressiveness of the dynamic context truncation. Table 4 presents the average number of retrieved documents under different α_{APS} settings. A high-guarantee setting ($\alpha_{\text{APS}} = 0.05$) yields an average of 4.8 documents, ensuring a 95% marginal coverage. While this extensive context significantly benefits single-hop General QA by minimizing the risk of omitting critical evidence, the excessive information introduces substantial noise, which severely impairs the reasoning quality in complex Multi-Hop QA (see Table 3). In contrast, a low-guarantee setting ($\alpha_{\text{APS}} = 0.35$) returns only 2.4 documents on average. This aggressive truncation fails to provide sufficient supporting facts, resulting in suboptimal performance across both tasks. Consequently, our default configuration ($\alpha_{\text{APS}} = 0.20$) strikes the optimal balance between comprehensive information retrieval and effective noise reduction.

5 Related Works

5.1 Retrieval in LLMs

Traditional RAG (Lewis et al., 2021; Gao et al., 2024) significantly expands the knowledge boundaries of LLMs by prepending retrieved external documents to the input context. With the continuous evolution of RAG, the emergence of frameworks such as Adaptive RAG (Jeong et al., 2024), Search-o1 (Li et al., 2025), and SAKI-RAG (Tao et al., 2025) has highlighted the inherent challenges of determining when to trigger retrieval in static paradigms. Concurrently, approaches that integrate retrieval with Reinforcement Learning (Jin et al., 2025; He et al., 2026; Zhao et al., 2025; Singh et al., 2026), have been introduced. However, these methods universally rely on fixed Top-K truncation. This static constraint fails to guarantee the marginal coverage of the retrieved knowledge.

5.2 Reinforcement Learning for Agents

Reinforcement Learning has fundamentally transformed the capabilities of LLMs, evolving them from passive generators into autonomous agents (Luo et al., 2025), such as SWE-agents (Zhang et al., 2026; Wei et al., 2025), Web Agents (Ding et al., 2026; Guo et al., 2026), and Search Agents (Jin et al., 2025; Zhao et al., 2025; He et al., 2026; Singh et al., 2026). However, they universally face the problem of overconfidence driven by sparse, binary rewards, which subsequently leads to hallucination issues (Leng et al., 2025). Recent work proposes training models to explicitly verbalize their confidence scores alongside their answers (Damani et al., 2025). Yet, this approach struggles in Search Agents, as the discontinuous generation process caused by continuous tool invocations prevents the model from explicitly expressing its confidence.

5.3 Conformal Prediction and Uncertainty Quantification

CP (Vovk et al., 2005) offers highly reliable marginal coverage guarantees without distributional assumptions. Due to its theoretical rigor, CP has been widely applied in traditional classification and detection tasks (Wu et al., 2026; andéol et al., 2025). Recent works have integrated CP into LLMs (Kumar et al., 2023; Quach et al., 2024; Su et al., 2025) and further into RL fine-tuning for robust alignment (Chen et al., 2026). However, while CP guarantees marginal coverage, it fails to guarantee conditional coverage, exhibiting a lack of adaptability when faced with complex queries (Romano et al., 2020). Furthermore, CP methods fundamentally rely on a strict independent and identically distributed (i.i.d.) assumption, which is inherently violated in reinforcement learning where the model and policy are continuously updating. Therefore, breaking these constraints is essential for reliable agentic search.

6 Conclusion

We propose CAS, a framework that integrates Conformal Prediction to resolve the reliability crisis in RL-trained search agents. The synergy between APS and ACI ensures reliable document retrieval and mitigates model overconfidence. Empirically, CAS significantly enhances reasoning accuracy and reduces redundant tool invocations. By balancing theoretical rigor with practical performance, this work establishes a principled foundation for future

reliable autonomous agents.

7 Limitations

Although CAS demonstrates significant potential in improving the accuracy and efficiency of Search Agents, several limitations remain. First, our empirical validation is primarily focused on general open-domain Question Answering (QA) tasks. While CAS provides robust statistical guarantees within these general information-seeking contexts, its applicability in highly specialized professional domains remains unexplored. Second, the framework relies heavily on a strong external teacher model (e.g., DeepSeek-V3.2) to construct the calibration set for the APS by decomposing queries and acting as a relevance judge. Finally, CAS primarily focuses on outcome reliability without extending statistical guarantees to the intermediate reasoning process. Future research should explore verifiable process reliability.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2024YFF0907802 and 2024YFF0907803) and the National Natural Science Foundation of China (62276230).

References

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges.

Léo andéol, Luca Mossina, Adrien Mazoyer, and Sébastien Gerchinovitz. 2025. Conformal object detection by sequential risk control.

Anastasios N. Angelopoulos and Stephen Bates. 2022. A gentle introduction to conformal prediction and distribution-free uncertainty quantification.

Debashish Chakraborty, Eugene Yang, Daniel Khashabi, Dawn Lawrie, and Kevin Duh. 2026. *Principled Context Engineering for RAG: Statistical Guarantees via Conformal Prediction*, page 537–546. Springer Nature Switzerland.

Tiejin Chen, Xiaoou Liu, Vishnu Nandam, Kuan-Ru Liou, and Hua Wei. 2026. Conformal feedback alignment: Quantifying answer-level reliability for robust llm alignment.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. 2025. Beyond binary rewards: Training lms to reason about their uncertainty.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Erhang Li, Fangqi Zhou, Fangyun Lin, Fucong Dai, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoran Wei, Haowei Zhang, Haowen Luo, Haozhe Ji, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, Jialiang Huang, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jingchang Chen, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jinhua Zhu, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexin Huang, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Liang Zhao, Liangsheng Yin, Lihua Guo, Lingxiao Luo, Linwang Ma, Litong Wang, Liyue Zhang, M. S. Di, M. Y Xu, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Panpan Huang, Peixin Cong, Peiyi Wang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, S. H. Liu, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Songyang Zhou, Tao Ni, Tao Yun, Tian Pei, Tian Ye, Tianyuan Yue, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjun Gao, Wentao Zhang, Xi Gao, Xiangwen Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyuan Li, Xu Chen, Xuecheng Su, Xuehai Pan, Xuheng Lin, Xuwei Fu, Y. Q. Wang, Yang Zhang, Yanhong Xu, Yanru Ma, Yao Li, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yiliang Xiong, Ying He, Ying Zhou, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuduan Wang, Yue Gong, Yuhan Wu, Yuheng Zou, Yukun Li, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan

Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehua Zhao, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhiyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Zizheng Pan, Zongqing Yao, Bei Feng, Hui Li, J. L. Cai, Jiaqi Ni, Lei Xu, Meng Li, Ning Tian, R. J. Chen, R. L. Jin, S. S. Li, Shuang Zhou, Tianyu Sun, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xinnan Song, Xinyi Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, Dongjie Ji, Jian Liang, Jianzhong Guo, Jin Chen, Leyi Xia, Miaojun Wang, Mingming Li, Peng Zhang, Ruyi Chen, Shangmian Sun, Shaoqing Wu, Shengfeng Ye, T. Wang, W. L. Xiao, Wei An, Xianzu Wang, Xiaowen Sun, Xiaoxiang Wang, Ying Tang, Yukun Zha, Zekai Zhang, Zhe Ju, Zhen Zhang, and Zihua Qu. 2025. Deepseek-v3.2: Pushing the frontier of open large language models.

Hang Ding, Peidong Liu, Junqiao Wang, Ziwei Ji, Meng Cao, Rongzhao Zhang, Lynn Ai, Eric Yang, Tianyu Shi, and Lei Yu. 2026. Dynaweb: Model-based reinforcement learning of web agents.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Isaac Gibbs and Emmanuel Candès. 2021. Adaptive conformal inference under distribution shift.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Yuyu Guo, Wenjie Yang, Siyuan Yang, Ziyang Liu, Cheng Chen, Yuan Wei, Yun Hu, Yang Huang, Guoliang Hao, Dongsheng Yuan, Jianming Wang, Xin Chen, Hang Yu, Lei Lei, and Peng Di. 2026. Opagent: Operator agent for web navigation.

Bowei He, Minda Hu, Zenan Xu, Hongru Wang, Licheng Zong, Yankai Chen, Chen Ma, Xue Liu, Pluto Zhou, and Irwin King. 2026. Search-r2: Enhancing search-integrated reasoning via actor-refiner collaboration.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering.

Bhawesh Kumar, Charlie Lu, Gauri Gupta, Anil Palepu, David Bellamy, Ramesh Raskar, and Andrew Beam. 2023. Conformal prediction with large language models for multi-choice question answering.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. 2017. Distribution-free predictive inference for regression.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. 2025. Taming overconfidence in llms: Reward calibration in rlhf.

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. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models.

Tianle Liang, Yifu Chen, Shengpeng Ji, Yijun Chen, Zhiyang Jia, Jingyu Lu, Fan Zhuo, Xueyi Pu, Yangzhuo Li, and Zhou Zhao. 2026. Voxmind: An end-to-end agentic spoken dialogue system.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts.

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, Rongcheng Tu, Xiao Luo, Wei Ju, Zhiping Xiao, Yifan Wang, Meng Xiao, Chenwu Liu, Jingyang Yuan, Shichang Zhang, Yiqiao Jin, Fan Zhang, Xian Wu, Hanqing Zhao, Dacheng Tao, Philip S. Yu, and Ming Zhang. 2025. Large language model agent: A survey on methodology, applications and challenges.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. 2024. Conformal language modeling.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report.

Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. 2020. Classification with valid and adaptive coverage.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools.

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.

Aditi Singh, Abul Ehtesham, Saket Kumar, Tala Talaei Khoei, and Athanasios V. Vasilakos. 2026. Agentic retrieval-augmented generation: A survey on agentic rag.

Jiayuan Su, Fulin Lin, Zhaopeng Feng, Han Zheng, Teng Wang, Zhenyu Xiao, Xinlong Zhao, Zuozhu Liu, Lu Cheng, and Hongwei Wang. 2025. Cp-router: An uncertainty-aware router between llm and lrm.

Wenyu Tao, Xiaofen Xing, Zeliang Li, and Xiangmin Xu. 2025. SAKI-RAG: Mitigating context fragmentation in long-document RAG via sentence-level attention knowledge integration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1195–1213, Suzhou, China. Association for Computational Linguistics.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multihop questions via single-hop question composition.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. 2005. *Algorithmic Learning in a Random World*. Springer-Verlag, Berlin, Heidelberg.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I. Wang. 2025. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution.

Zixuan Wu, So Won Jeong, Yating Liu, Yeo Jin Jung, and Claire Donnat. 2026. Filtering with confidence: When data augmentation meets conformal prediction.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report.

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.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models.

Chenlong Yin, Zeyang Sha, Shiwen Cui, Changhua Meng, and Zechao Li. 2026. The reasoning trap: How enhancing llm reasoning amplifies tool hallucination.

Zhaoxi Zhang, Yitong Duan, Yanzhi Zhang, Yiming Xu, Zhixiang Wang, Kun Liang, Yang Li, Jiahui Liang, Deguo Xia, Jizhou Huang, Jiyan He, and Yunfang Wu. 2026. One tool is enough: Reinforcement learning for repository-level llm agents.

Shu Zhao, Tan Yu, Anbang Xu, Japinder Singh, Aaditya Shukla, and Rama Akkiraju. 2025. Parallelsearch: Train your llms to decompose query and search subqueries in parallel with reinforcement learning.

A Pseudocode

We provide the pseudocode of CAS in Algorithm 1

Algorithm 1: CAS

Input: Dataset $\mathcal{D}$, Initial policy π_θ, Search engine $\mathcal{S}$
Parameters: Group size G, ACI step size γ, Target error rate ρ, Initial calibration set size N_{cal}
Output: Optimized policy π_θ

Initialize ACI threshold $\alpha_1 \leftarrow \rho$;
Initialize calibration set $\mathcal{C}$ with N_{cal} scores from untrained policy;
for *each RL training iteration* $t = 1, 2, \ldots$ **do**
  Sample a prompt $x_t \sim \mathcal{D}$;
  for *trajectory* $i = 1$ **to** G **do**
    while *trajectory* y_i *not terminated* **do**
      Generate tokens $y_{next} \sim \pi_\theta(\cdot \mid x_t, y_i)$;
      if *search query* q *generated* **then**
        $D_{\text{APS}} \leftarrow \text{APS}(\mathcal{S}(q))$ (Sec. 3.2);
        $y_i \leftarrow y_i \oplus D_{\text{APS}}$;
      end
    end
  end
  for *trajectory* $i = 1$ **to** G **do**
    Compute reward $R_i = r_{\text{acc}} + r_{\text{fmt}}$;
    Compute NLL score $s(x_t, y_i)$ for answer tokens (Eq. (6));
  end
  Compute advantages $A_1, \ldots, A_G$ from rewards $\{R_i\}$;
  Compute quantile threshold $\hat{q}_t \leftarrow \hat{Q}_t(1 - \alpha_t)$ over $\mathcal{C}$;
  for *trajectory* $i = 1$ **to** G **do**
    Determine ACI confidence weight ω_i using $\hat{q}_t$ (Eq. (9));
  end
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{s(x_t, y_1), \ldots, s(x_t, y_G)\}$;
  Update π_θ by minimizing the ACI-guided objective (Eq. (8));
  Update ACI threshold α_{t+1} (Eq. (7));
end

B Mathematical Proofs

In this section, we provide the formal mathematical proofs for the statistical guarantees of the CP methods used in CAS. We begin with the fundamental marginal coverage guarantee of standard Split CP.

B.1 Marginal Coverage Guarantee of Conformal Prediction

Following the standard proof of validity for split-conformal prediction (Angelopoulos and Bates, 2022), we demonstrate that the prediction sets constructed via CP possess a strict, finite-sample marginal coverage guarantee.

Theorem 1 *(Conformal calibration coverage guarantee). Suppose the calibration data* $(X_i, Y_i)_{i=1,\ldots,n}$ *and the test point* (X_{test}, Y_{test}) *are independent and identically distributed (i.i.d.). Define the conformal quantile* $\hat{q}$ *as:*

$$\hat{q} = \inf\left\{ q : \frac{|\{i : s(X_i, Y_i) \le q\}|}{n} \ge \frac{\lceil (n+1)(1-\alpha) \rceil}{n} \right\} \tag{10}$$

and the resulting prediction sets as:

$$\mathcal{C}(X) = \{y : s(X, y) \le \hat{q}\} \tag{11}$$

Then, the marginal coverage satisfies:

$$\mathbb{P}(Y_{\text{test}} \in \mathcal{C}(X_{\text{test}})) \ge 1 - \alpha \tag{12}$$

Proof of Theorem 1. Let $s_i = s(X_i, Y_i)$ for $i = 1, \ldots, n$ and $s_{\text{test}} = s(X_{\text{test}}, Y_{\text{test}})$. To avoid handling ties, we consider the case where the non-conformity scores s_i are distinct with probability 1.

Without loss of generality, we assume the calibration scores are sorted such that $s_1 < s_2 < \cdots < s_n$. In this case, the quantile $\hat{q}$ can be explicitly written as:

$$\hat{q} = s_{\lceil (n+1)(1-\alpha) \rceil} \tag{13}$$

when $\alpha \ge \frac{1}{n+1}$, and $\hat{q} = \infty$ otherwise.

Note that in the case where $\hat{q} = \infty$, the prediction set includes the entire label space, i.e., $\mathcal{C}(X_{\text{test}}) = \mathcal{Y}$, so the coverage property is trivially satisfied. Thus, we only need to handle the case when $\alpha \ge \frac{1}{n+1}$.

We proceed by noticing the strict equality of the two following events:

$$\{Y_{\text{test}} \in \mathcal{C}(X_{\text{test}})\} = \{s_{\text{test}} \le \hat{q}\} \tag{14}$$

Combining this with our definition of the sorted quantile $\hat{q}$ yields:

$$\{Y_{\text{test}} \in \mathcal{C}(X_{\text{test}})\} = \{s_{\text{test}} \le s_{\lceil (n+1)(1-\alpha) \rceil}\} \tag{15}$$

Now comes the crucial insight: by the exchangeability of the random variables $(X_1, Y_1), \ldots, (X_{\text{test}}, Y_{\text{test}})$, their corresponding non-conformity scores $s_1, \ldots, s_n, s_{\text{test}}$ are also exchangeable. Because they are exchangeable, s_{test} is equally likely to fall anywhere between the sorted calibration points $s_1, \ldots, s_n$. Therefore, the probability that s_{test} is less than or equal to the k-th sorted score is exactly:

$$\mathbb{P}(s_{\text{test}} \le s_k) = \frac{k}{n+1} \tag{16}$$

for any integer k. (Note that here, the randomness is over all variables $s_1, \ldots, s_n, s_{\text{test}}$).

From this property, we substitute $k = \lceil (n+1)(1-\alpha) \rceil$ to conclude:

$$\mathbb{P}\left(s_{\text{test}} \leq s_{\lceil (n+1)(1-\alpha) \rceil}\right) = \frac{\lceil (n+1)(1-\alpha) \rceil}{n+1} \geq 1-\alpha \tag{17}$$

which implies the desired result: $\mathbb{P}(Y_{\text{test}} \in \mathcal{C}(X_{\text{test}})) \geq 1-\alpha$.

Theorem 2 *(Conformal calibration upper bound). Additionally, if the scores $s_1, \ldots, s_n$ have a continuous joint distribution (i.e., avoiding ties), the coverage is tightly bounded from above:*

$$\mathbb{P}(Y_{\text{test}} \in \mathcal{C}(X_{\text{test}})) \leq 1-\alpha+\frac{1}{n+1} \tag{18}$$

(Proof deferred to Theorem 2.2 of Lei et al. (2017)).

B.2 Statistical Guarantees of Adaptive Prediction Sets

As established in the foundational literature, achieving exact finite-sample conditional coverage is theoretically impossible without strong distributional assumptions. However, the APS framework (Romano et al., 2020) effectively circumvents this limitation. It provides a rigorous marginal coverage guarantee while sensibly approximating conditional coverage by adapting the prediction set size to the local uncertainty of the input.

To construct the adaptive sets, APS introduces a generalized inverse quantile conformity score. Given a base model's probability estimate $\hat{\pi}$ and a uniform random variable $U \sim \text{Uniform}(0,1)$ for tie-breaking, the conformity score function E is defined as:

$$E(x, y, u; \hat{\pi}) = \min\{\tau \in [0,1] : y \in \mathcal{S}(x, u; \hat{\pi}, \tau)\} \tag{19}$$

where $\mathcal{S}$ is the generalized conditional quantile function that includes classes in descending order of their estimated probabilities until the cumulative mass reaches τ.

Using this conformity score, APS achieves the following rigorous marginal guarantee:

Theorem 3 *(Marginal coverage of APS). If the calibration samples $(X_i, Y_i)_{i \in \mathcal{I}_2}$ and the test sample (X_{test}, Y_{test}) are exchangeable, and the conformity scores are calculated using a model trained on a disjoint split $\mathcal{I}_1$, the APS prediction set $\hat{\mathcal{C}}_{APS}$ satisfies:*

$$\mathbb{P}\left(Y_{\text{test}} \in \hat{\mathcal{C}}_{\text{APS}}(X_{\text{test}})\right) \geq 1-\alpha \tag{20}$$

Furthermore, if the scores E_i are almost surely distinct, the coverage is bounded tightly from above by $1-\alpha+1/(|\mathcal{I}_2|+1)$.

Proof of Theorem 3. Let $E_i = E(X_i, Y_i, U_i; \hat{\pi})$ denote the conformity score for the i-th calibration sample in $\mathcal{I}_2$, and $E_{\text{test}} = E(X_{\text{test}}, Y_{\text{test}}, U_{\text{test}}; \hat{\pi})$ for the test point.

By the construction of the APS prediction set, a label y is included in $\hat{\mathcal{C}}_{\text{APS}}(X_{\text{test}})$ if and only if its requisite cumulative mass τ is less than or equal to the calibrated threshold $\hat{Q}_{1-\alpha}$. Mathematically, we know that:

$$Y_{\text{test}} \in \hat{\mathcal{C}}_{\text{APS}}(X_{\text{test}}) \iff E_{\text{test}} \leq \hat{Q}_{1-\alpha}(\{E_i\}_{i \in \mathcal{I}_2}) \tag{21}$$

where $\hat{Q}_{1-\alpha}(\{E_i\}_{i \in \mathcal{I}_2})$ is defined as the $\lceil (1-\alpha)(1+|\mathcal{I}_2|) \rceil$-th smallest value in the calibration score set $\{E_i\}_{i \in \mathcal{I}_2}$.

Because the data points (X, Y) are exchangeable and the uniform variables U are i.i.d., all the evaluated conformity scores E_{test} and $\{E_i\}_{i \in \mathcal{I}_2}$ are completely exchangeable. Under the property of exchangeability, the rank of E_{test} is uniformly distributed among the $|\mathcal{I}_2|+1$ scores. Therefore, the probability of the event that E_{test} falls below the empirical $(1-\alpha)$-quantile is bounded from below by the nominal level:

$$\mathbb{P}\left(E_{\text{test}} \leq \hat{Q}_{1-\alpha}(\{E_i\}_{i \in \mathcal{I}_2})\right) \geq 1-\alpha \tag{22}$$

which immediately establishes $\mathbb{P}(Y_{\text{test}} \in \hat{\mathcal{C}}_{\text{APS}}(X_{\text{test}})) \geq 1-\alpha$.

Asymptotic Conditional Coverage. While Theorem 3 guarantees marginal coverage, the structural design of APS provides an asymptotic approximation of conditional coverage. Consider an Oracle model with perfect knowledge of the true conditional distribution $\pi_y(x) = \mathbb{P}(Y = y | X = x)$. The Oracle's prediction set $\mathcal{C}_\alpha^{\text{oracle}}(x)$ naturally attains exact conditional coverage.

According to Romano et al. (2020), as the sample size increases and if the base predictive model is consistent (i.e., $\hat{\pi}_y(x) \approx \pi_y(x)$), the constructed sets $\mathcal{S}(X, U; \hat{\pi}, \tau)$ will converge to contain the true labels for exactly a fraction τ of the points. In this limit, the threshold $\hat{Q}_{1-\alpha} \approx 1-\alpha$, and the decision rule approaches:

$$\hat{\mathcal{C}}_{\text{APS}}(X_{\text{test}}) \approx \{y \in \mathcal{Y} : E(X_{\text{test}}, y, U_{\text{test}}; \pi) \leq 1-\alpha\} \tag{23}$$

which mathematically equates to the exact output of the Oracle procedure, thereby closely approximating optimal conditional coverage in complex data scenarios.

B.3 Statistical Guarantees of Adaptive Conformal Inference

Standard CP fundamentally relies on the exchangeability of the data. In online settings, the policy

continuously evolves, leading to severe distribution shifts that violate the i.i.d. assumption. To maintain rigorous coverage, we employ ACI (Gibbs and Candès, 2021).

ACI guarantees the target coverage frequency over long-time intervals irrespective of the true data-generating process by dynamically adjusting the nominal error level. Following Gibbs and Candès (2021), let $\rho \in (0, 1)$ be the target miscoverage rate. At each time step t, the algorithm uses a parameter α_t to construct the prediction set $\hat{\mathcal{C}}_t(\alpha_t)$, and records the miscoverage event:

$$\text{err}_t := \begin{cases} 1, & \text{if } Y_t \notin \hat{\mathcal{C}}_t(\alpha_t) \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

The parameter α_t is recursively updated using a step size $\gamma > 0$:

$$\alpha_{t+1} := \alpha_t + \gamma(\rho - \text{err}_t) \tag{25}$$

To establish the distribution-free guarantee, we assume that with probability one, $\alpha_1 \in [0, 1]$ and the quantile function $\hat{Q}_t(x)$ is non-decreasing with $\hat{Q}_t(x) = -\infty$ for $x < 0$ and $\hat{Q}_t(x) = \infty$ for $x > 1$.

Lemma 4 *(Boundedness of α_t, Lemma 4.1 in Gibbs and Candès (2021)). With probability one, we have that for all $t \in \mathbb{N}$, $\alpha_t \in [-\gamma, 1 + \gamma]$.*

Proof of Lemma 4. Assume by contradiction that with positive probability, the sequence $\{\alpha_t\}_{t\in\mathbb{N}}$ is such that $\inf_t \alpha_t < -\gamma$ (the case for $\sup_t \alpha_t > 1 + \gamma$ is symmetric). Notice that the maximum change in one step is bounded: $\sup_t |\alpha_{t+1} - \alpha_t| = \sup_t \gamma|\rho - \text{err}_t| < \gamma$. Thus, with positive probability, we may find a specific time step $t \in \mathbb{N}$ such that $\alpha_t < 0$ and $\alpha_{t+1} < \alpha_t$.

However, by the boundary definition of the quantile function:

$$\alpha_t < 0 \implies \hat{Q}_t(1 - \alpha_t) = \infty \implies \text{err}_t = 0 \tag{26}$$

Substituting $\text{err}_t = 0$ into the update rule gives:

$$\alpha_{t+1} = \alpha_t + \gamma(\alpha - 0) \geq \alpha_t \tag{27}$$

This contradicts the assumption that $\alpha_{t+1} < \alpha_t$. Thus, $\mathbb{P}(\exists t \text{ such that } \alpha_{t+1} < \alpha_t < 0) = 0$, establishing the lower bound.

Theorem 5 *(Distribution-free asymptotic coverage, Proposition 4.1 in Gibbs and Candès (2021)). With probability one, for all horizon lengths $T \in \mathbb{N}$, the empirical miscoverage rate satisfies:*

$$\left| \frac{1}{T} \sum_{t=1}^{T} \text{err}_t - \rho \right| \leq \frac{\max\{\alpha_1, 1 - \alpha_1\} + \gamma}{T\gamma} \tag{28}$$

In particular, as $T \to \infty$, the average miscoverage converges almost surely to the target rate α:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \text{err}_t \overset{a.s.}{=} \rho \tag{29}$$

Proof of Theorem 5. By recursively expanding the update rule $\alpha_{t+1} = \alpha_t + \gamma(\alpha - \text{err}_t)$ from $t = 1$ to T, we obtain the telescoping sum:

$$\alpha_{T+1} = \alpha_1 + \sum_{t=1}^{T} \gamma(\rho - \text{err}_t) \tag{30}$$

Rearranging the terms to isolate the empirical average of err_t, we get:

$$\frac{1}{T} \sum_{t=1}^{T} (\text{err}_t - \rho) = \frac{\alpha_1 - \alpha_{T+1}}{T\gamma} \tag{31}$$

Taking the absolute value on both sides yields:

$$\left| \frac{1}{T} \sum_{t=1}^{T} \text{err}_t - \rho \right| = \frac{|\alpha_1 - \alpha_{T+1}|}{T\gamma} \tag{32}$$

From Lemma 4, we know that $\alpha_{T+1} \in [-\gamma, 1 + \gamma]$. Given that the initialization $\alpha_1 \in [0, 1]$, the maximum possible distance between α_1 and α_{T+1} is bounded by:

$$|\alpha_1 - \alpha_{T+1}| \leq \max\{\alpha_1 - (-\gamma), 1 + \gamma - \alpha_1\} \tag{33}$$

which simplifies to $\max\{\alpha_1, 1 - \alpha_1\} + \gamma$. Substituting this upper bound into the absolute difference completes the proof for Equation 28.

Taking the limit as $T \to \infty$, the right-hand side of Equation 28 diminishes to zero (since γ is a fixed positive constant), proving that ACI flawlessly achieves the exact marginal coverage frequency over time, without making any assumptions on the nature of the data distribution shift.

C Supplementary Implementation Details

Environment Our framework operates on a dual-service architecture developed based on the VeRL distributed reinforcement learning framework. The Training Service executes GRPO using Python 3.12, PyTorch 2.8.0 (CUDA 12.9), and is distributed across 4 GPUs via Ray (v2.49.2). To accelerate asynchronous multi-turn rollouts, it leverages sglang (v0.5.3rc0) equipped with the flashinfer backend and flash-attn (v2.8.3). The Retrieval Service operates independently as a FastAPI-based REST endpoint using Python 3.10 and PyTorch

2.4.0 (CUDA 12.1). It utilizes faiss-gpu (v1.8.0) and the e5-base-v2 embedding model, performing high-throughput dense retrieval via mean pooling on 256-token inputs with FP16 precision. The retrieval backend is configured to handle a peak rate of 120 queries per second (QPS) with a 30-second timeout.

Configurations This encompasses our data processing, optimization, and CP settings. Data & Rollout: Models are trained on a unified search-integrated reasoning dataset in Parquet format. We set the maximum prompt, response, and context lengths to 4096, 3000, and 15,000 tokens, respectively, filtering out prompts that exceed the limit. During the GRPO step, we sample $G = 5$ trajectories per prompt with a maximum of 4 assistant turns. Optimization: The Actor is optimized with a learning rate of 1×10^{-6} and a warmup ratio of 0.285 (100 steps), while the Critic uses 1×10^{-5}. Training employs a global batch size of 512, a low-variance KL penalty coefficient of 0.001, and Fully Sharded Data Parallel (FSDP) with tensor model parallelism set to 1. Reward Design: The rule-based reward comprises an EM accuracy score (weight 1.0) and format rewards (0.2 for structural integrity, 0.1 for the final answer boundary). CAS: On the retrieval side, APS are applied with a significance level $\alpha_{APS} = 0.20$ and a temperature of 0.01, dynamically restricting the retrieved subset to between 1 and 5 documents. On the training side, ACI is initialized with a target error rate $\rho = 0.25$ and an update step size $\gamma = 0.005$. We apply a discount factor $\eta = 0.5$ to penalize low-confidence trajectories ($s_i > \hat{q}_t$), while empirical error tracking utilizes an Exponential Moving Average (EMA) ratio of 0.05 to maintain quantile stability.

Hardware All experiments were conducted on a single server node. The server is configured with dual-socket AMD EPYC 9454 48-Core processors, providing a total of 96 physical cores and 192 threads, organized into two NUMA nodes. The server is equipped with four NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs and 755 GiB of system memory. Storage infrastructure includes a 446.6 GB drive for the OS and environment, alongside a 14.6 TB enterprise-grade drive for high-throughput data caching. The software environment is built on Ubuntu 24.04.3 LTS.

D Prompts

In this section, we present the detailed prompt templates utilized across different stages of CAS. The configuration of the reasoning template in Table 5 is adapted from Search-R1 (Jin et al., 2025) to maintain consistency in agentic behavior. Additionally, the query decomposition prompt in Table 6 and the retrieval relevance judge prompt in Table 7 are specifically employed to construct the calibration set for the retrieval-side APS.

E Additional Experimental Analysis on Qwen3-8B

In Table 1, it is observed that our method's performance on Qwen3-8B consistently outperforms that on Qwen2.5-3B-Instruct. This improvement is primarily attributed to the inherent model capacity of Qwen3-8B, which exhibits a significant advantage over the 3B-Instruct variant, as illustrated in Figure 3a. Regarding the search behavior shown in Figure 3b, we note that the average search calls for Qwen3-8B remain lower than those of Qwen2.5-3B-Instruct during approximately the first 80 training steps. This phenomenon occurs because Qwen2.5-3B-Instruct, as a smaller model, tends to exhibit erratic and indiscriminate tool invocation during the early stages of training. In contrast, the larger parameter scale of Qwen3-8B ensures more efficient search calls from the beginning. This efficiency is further evidenced by comparing Figure 3a and Figure 3b, where Qwen3-8B achieves substantially higher EM scores despite a noticeably lower frequency of search invocations. Furthermore, as depicted in Figure 3a, although the number of search calls for Qwen3-8B increases slightly relative to Qwen2.5-3B-Instruct after convergence, it remains significantly more efficient than baseline methods lacking ACI constraints. This demonstrates that the ACI mechanism effectively modulates low-confidence trajectories even when applied to the 8B model.

For the experiments involving Qwen3-8B, the thinking mode is disabled by default, as enabling this feature leads to a drastic reduction in training effectiveness, as shown in Figure 3c. The underlying cause is revealed in Figure 3d: after enabling the thinking mode, the model initially tends towards aggressive search calls due to the interleaving of internal reasoning with our prescribed reasoning grammar. However, the model rapidly discovers that many single-hop problems can be

resolved solely through internal reasoning. Consequently, it gradually ceases to invoke the search tool, leading to a complete cessation of active information gathering and rendering the training process ineffective for the intended search-integrated tasks.

F Detailed Results for Ablation and Sensitivity Analysis

Corresponding to Table 3 in the main text, we provide the complete results across all individual datasets in Table 8.

G Case Study

In this section, we present representative qualitative cases to illustrate the core behavioral patterns of CAS.

Table 10 illustrates a straightforward single-hop scenario. The model directly addresses the factual query by formulating a precise search action. Because the retrieved documents are clean and highly relevant, the agent swiftly concludes its reasoning and extracts the correct answer without unnecessary actions.

Table 9 presents a more challenging single-hop case characterized by high retrieval noise. Although the search results contain highly distracting entities with similar names, the agent successfully evaluates the contextual relevance of each document, filters out the irrelevant distractors, and accurately grounds its final answer on the correct source.

Table 11 demonstrates the framework's capability to handle multi-hop queries through interleaved reasoning and search. The agent dynamically decomposes the complex task in its initial `<think>` block, retrieves the missing bridge entity in the first hop, and uses this intermediate information to construct a targeted query for the subsequent hop. This iterative process highlights the effectiveness of allowing the model to flexibly transition between internal deliberation and external tool interaction.

Prompt 1: Template for CAS

You are Qwen, created by Alibaba Cloud. You are a helpful assistant. Answer the given question. You must conduct reasoning inside `<think>` and `</think>` first every time you get new information. After reasoning, if you find you lack some knowledge, you can call a search engine by `<search>query</search>` and it will return the top searched results between `<information>` and `</information>`. You can search as many times as you want. If you find no further external knowledge needed, you can directly provide the answer inside `<answer>` and `</answer>`, without detailed illustrations. For example, `<answer> Beijing </answer>`.

Question: [Question]

Table 5: Template for CAS reasoning process.

Prompt 2: Query Decomposition

You decompose QA tasks into hop-level search query-answer pairs. Return strict JSON. Given a QA sample, split it into single-hop query-answer pairs.

Rules:
1) If single-hop, return one pair.
2) If multi-hop, return one pair per hop.
3) answer should be concise and factual.
4) Output JSON only in schema: `{"pairs":[{"query":"...","answer":"..."}]}`

question: {question}
ground_truth_answers: {gt_answers}
metadata: {metadata}
extra_info: {extra_info}

Table 6: Prompt for decomposing multi-hop queries into single-hop sub-queries.

Prompt 3: Retrieval Relevance Judge

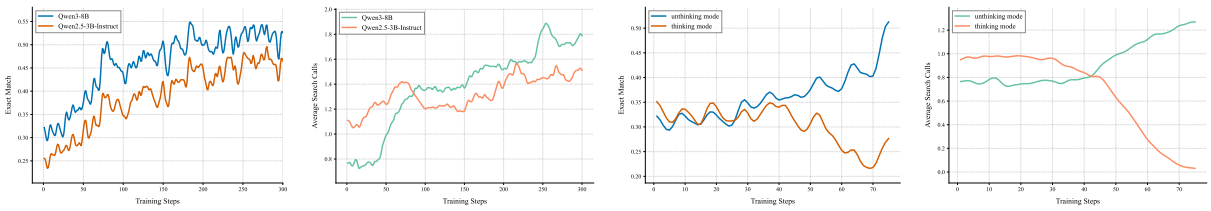
You are a precise retrieval relevance judge for QA reasoning. You are given a retrieval query, its target answer, and retrieved documents. Decide which document(s) can support reasoning to the target answer.

Return strict JSON with schema:
`{"golden_doc_indices":[0,1], "best_golden_doc_index":0, "reason":"..."}`

Rules:
1) If none can support the answer, return empty golden_doc_indices and -1 as best index.
2) best_golden_doc_index must be one item in golden_doc_indices, or -1.
3) Never output markdown.

query: {query}
answer: {answer}
retrieved_docs: {docs_for_llm}

Table 7: Judge prompt for locating the most relevant documents.



(a) EM score during training. (b) Average search calls. (c) EM (Thinking mode). (d) Calls (Thinking mode).

Figure 3: Visualization of training metrics and the impact of the thinking mode. Panels (a) and (b) illustrate the progression of accuracy and tool invocation frequency during training. Panels (c) and (d) demonstrate how the thinking mode influences the final EM scores and the efficiency of the search process.

Table 8: Detailed ablation study and sensitivity analysis on Qwen2.5-3B-Instruct. The table presents component-wise ablations and sensitivity analyses for ρ and α_{APS}.

Methods	General QA			Multi-Hop QA				Average
	NQ†	TriviaQA*	PopQA*	HotpotQA†	2WikiMultiHopQA*	Musique*	Bamboogle*	
- ACI	0.417	0.598	0.430	0.378	0.372	0.143	0.352	0.384
- APS	0.435	0.603	0.433	0.393	0.374	0.143	0.344	0.389
$\rho = 0.1$	0.441	0.601	0.449	0.382	0.375	0.132	0.320	0.386
$\rho = 0.4$	0.438	0.596	0.452	0.356	0.351	0.113	0.272	0.368
$\alpha_{\text{APS}} = 0.35$	0.420	0.584	0.432	0.372	0.370	0.146	0.288	0.372
$\alpha_{\text{APS}} = 0.05$	0.442	0.613	0.459	0.328	0.285	0.076	0.248	0.350

Question: What BBC television drama serial starring Jim McManus was based on Sarah Waters' debut novel of the same name?
Golden Answer: ['Tipping the Velvet']

<think>
The question asks for a BBC television drama serial based on Sarah Waters' debut novel of the same name. I recall that Sarah Waters' debut novel is probably "Tipping the Velvet", but I need to confirm it, especially because Jim McManus may create ambiguity in search results.
</think>

<search> Jim McManus Tipping the Velvet </search>

<information>
Doc 1 (Title: "Jack McManus")
Jack McManus Jack McManus (born Thomas McManus, March 1862 – May 1905), also known as Eat 'Em Up, was a noted New York City gangster around the turn of the 20th century. Born in Boston, he was considered one of the premier boxers of the underworld, rivaled only by Monk Eastman.

Doc 2 (Title: "Jimmy Velvet")
toured in 1973-1978 with the Jimmy Velvet Show band. Band members included Steve Morgan (guitar), John Collins (bassist), Dennis Moog (drums), and Kathy Slinkard Velvet. Velvet was championed by TV host Dick Clark, who had him on his "American Bandstand" show more than once.

Doc 3 (Title: "Tipping the Velvet")
Tipping the Velvet is a historical novel published as Sarah Waters' debut novel in 1998. Set in England during the 1890s, it tells a coming of age story about a young woman named Nan who falls in love with a male impersonator, follows her to London, and finds various ways to support herself as she journeys through the city.

Doc 4 (Title: "James McManus")
James "Jim" McManus (born March 22, 1951) is an American teacher, writer and poker player living in Kenilworth, Illinois. McManus is best known as the author of the book "Positively Fifth".
</information>

<think>
The search results contain several noisy results about other people named McManus or Velvet, but Doc 3 directly confirms that "Tipping the Velvet" is Sarah Waters' debut novel. Since the question asks for the BBC drama serial based on that debut novel of the same name, the answer is "Tipping the Velvet".
</think>

<answer> Tipping the Velvet </answer>

Table 9: Case Study: Complex Single-Hop QA.

Question: Who won the Men's Singles French Open Tennis Tournament in 2018?
Golden Answer: ['Rafael Nadal']

<think>
I need to find out who won the Men's Singles French Open Tennis Tournament in 2018. I remember Rafael Nadal was dominant at the French Open around that time, but I should verify the exact 2018 result.
</think>

<search> 2018 French Open Men's Singles winner </search>

<information>
Doc 1 (Title: "2018 French Open – Men's Singles")
2018 French Open – Men's Singles Rafael Nadal was the defending champion and successfully defended his title, defeating Dominic Thiem in the final, 6–4, 6–3, 6–2. Nadal equaled Margaret Court's all-time record of 11 Grand Slam singles titles won at one major and became the first player to achieve that feat in the Open era. He dropped only one set during the entire tournament (losing the first set of his quarterfinal match against Diego Schwartzman). Nadal also retained the ATP No. 1 singles ranking by defending the title.

Doc 2 (Title: "2018 French Open – Men's Singles")
72nd in the ATP rankings, became the first unseeded men's singles semifinalist at the French Open since Gaël Monfils in 2008, the lowest-ranked male singles player to reach the French Open semifinals since Andriy Medvedev in 1999, and the first Italian man to reach a Grand Slam singles semifinal since Corrado Barazzutti at the 1978 French Open. 2018 French Open – Men's Singles Rafael Nadal was the defending champion and successfully defended his title, defeating Dominic Thiem in the final, 6–4, 6–3, 6–2.
</information>

<think>
The search results clearly state that Rafael Nadal won the 2018 French Open men's singles title by defeating Dominic Thiem in the final.
</think>

<answer> Rafael Nadal </answer>

Table 10: Case Study: Simple Single-Hop QA.

Question: Which hockey team does the younger brother of Sean James Pronger currently work for?
Golden Answer: ['the Florida Panthers']

<think>
I need to identify Sean James Pronger's younger brother first, then determine which hockey team that person currently works for. This requires at least two steps: family relation lookup, then current team lookup.
</think>

<search> Sean James Pronger younger brother </search>

<information>
Doc 1 (Title: "Jordan Reaves")
with Saskatchewan for 2018 training camp. Jordan is the youngest son of former Winnipeg Blue Bomber running back Willard Reaves, who played with the team for five years and was a member of the 72nd Grey Cup winning team. Jordan is also the younger brother of Ryan Reaves, a professional hockey player and right wing for the Vegas Golden Knights of the National Hockey League.

Doc 2 (Title: "Chris Pronger")
the 1990–91 season. One of his defence partners in Stratford was future NHL player Greg de Vries. In May 1991, Pronger indicated he was going to join his older brother Sean at Bowling Green State University (NCAA) instead of opting for the OHL. Regardless of his pre-draft indications, Pronger was selected in the 6th round by the Peterborough Petes in the OHL Priority Selection.

Doc 3 (Title: "Luke Schenn")
hits. He believes he is at his best when he is playing a tough game, while staying responsible. Growing up, Schenn patterned his play after fellow defencemen Chris Pronger and Rob Blake. Schenn was born on November 2, 1989, in Saskatoon, Saskatchewan. His younger brother Brayden was picked fifth overall in the 2009 NHL Entry Draft by the Los Angeles Kings.

Doc 4 (Title: "Jordan Eberle")
two sisters, Ashley and Whitney, and a younger brother, Dustin. He attended high school at the Athol Murray College of Notre Dame in Wilcox, Saskatchewan. His brother Dustin was drafted.
</information>

<think>
Doc 2 says Chris Pronger planned to join his older brother Sean at Bowling Green State University. Therefore, Sean James Pronger's younger brother is Chris Pronger. Now I need to find which hockey team Chris Pronger currently works for.
</think>

<search> Chris Pronger current team </search>

<information>
Doc 1 (Title: "Chris Pronger")
Chris Pronger Christopher Robert Pronger (born October 10, 1974) is a Canadian former professional ice hockey defenceman who is currently the senior advisor of hockey operations for the Florida Panthers. He had not played since November 2011 due to post-concussion syndrome.

Doc 2 (Title: "Chris Pronger")
Chris Pronger Christopher Robert Pronger is a Canadian former professional ice hockey defenceman who is currently the senior advisor of hockey operations for the Florida Panthers.
</information>

<think>
The second search confirms that Chris Pronger currently works as senior advisor of hockey operations for the Florida Panthers. Therefore, the hockey team is the Florida Panthers.
</think>

<answer> Florida Panthers </answer>

Table 11: Case Study: Multi-Hop QA.