Evaluate-as-Action: Self-Evaluated Process Rewards for Retrieval-Augmented Agents

Anonymous ACL submission

Abstract

Retrieval-augmented agents can query external evidence, yet their reliability in multi-step reasoning remains limited: noisy retrieval may derail multi-hop question answering, while outcome-only reinforcement learning provides credit signals that are too coarse to optimize intermediate steps. To address these limitations, we propose EvalAct (Evaluate-as-Action), which converts implicit retrieval quality assessment into an explicit action and enforces a coupled Search-to-Evaluate protocol so that each retrieval is immediately followed by an explicit evaluation score, yielding process signals aligned with the interaction trajectory. To leverage these signals, we introduce Process-Calibrated Advantage Rescaling (PCAR), a GRPO-based optimization method that rescales advantages at the segment level according to evaluation scores, emphasizing reliable segments while updating uncertain ones conservatively. Experiments on seven open-domain QA benchmarks show that EvalAct achieves the best average EM across multiple model scales, with the largest gains on multi-hop tasks. Ablations verify that the explicit evaluation loop drives the primary improvements, while PCAR provides consistent additional benefits. Code is available at https://anonymous.4open.science/r/EvalAct-Anonymized-9614.

1 Introduction

Large language model (LLM) agents have shifted automated reasoning from passive response generation to autonomous problem solving, where models plan, interact with external tools, and iteratively refine their beliefs across multi-step trajectories (Yao et al., 2022; Schick et al., 2023). Retrieval-augmented generation (RAG) further extends this capability by grounding decisions in external evidence, enabling open-domain question answering beyond the limits of parametric knowledge (Lewis et al., 2020; Guu et al., 2020). However, as queries shift from single-hop factoids to multi-hop narratives, the central bottleneck is no longer tool access itself, but the agent's ability to navigate, verify, and synthesize evidence over long-horizon, noise-prone interaction sequences (Trivedi et al., 2023; Asai et al., 2024).

Despite substantial progress, ensuring reliable intermediate reasoning remains a key challenge. Existing agentic baselines, from prompting methods that interleave retrieval and reasoning (Trivedi et al., 2023) to RL-based search agents such as Search-R1 (Jin et al., 2025) and refinement frameworks such as AutoRefine (Shi et al., 2025), still rely primarily on implicit internal reasoning for noise suppression and self-correction. This paradigm suffers from two fundamental limitations. First, **error propagation**: without an explicit, immediate mechanism for evidence verification, a single irrelevant document can derail downstream reasoning, causing irreversible trajectory drift in multi-hop settings. Second, **coarse credit assignment**: standard RL optimization, including PPO-based RLHF (Ouyang et al., 2022; Schulman et al., 2017) and outcome-reward post-training methods such as GRPO (Shao et al., 2024), typically relies on sparse signals tied to final-answer correctness. Such outcome-only supervision cannot distinguish informative retrieval steps from redundant or misleading actions within long trajectories; as a result, the optimizer often reinforces or penalizes an entire trajectory nearly uniformly, degrading sample efficiency and causing performance saturation as task complexity grows.

To address these challenges, we introduce EvalAct, a reinforcement learning framework that transforms the agent's implicit self-assessment of retrieval quality into an explicit, policy-selectable action. EvalAct enforces a strictly coupled search-then-evaluate protocol: each `Search` action must be immediately followed by an `Evaluate` action that produces a structured self-assessment score reflecting the utility of the retrieved evidence. This design directly addresses the two limitations

identified above. At inference time, the evaluation output provides actionable control signals that facilitate early pruning of unproductive branches, reducing error propagation without external oracle supervision. During training, it produces dense, trajectory-aligned process signals that make intermediate reliability directly optimizable and enable finer-grained credit assignment.

To leverage these process signals effectively, we further propose **Process-Calibrated Advantage Rescaling (PCAR)** built upon Group Relative Policy Optimization (GRPO) (Shao et al., 2024). Instead of broadcasting a single trajectory-level advantage to all tokens, PCAR uses step-wise self-evaluation scores to modulate updates at the segment level, amplifying gradients for reliable, progress-making steps while applying conservative updates to uncertain segments. Importantly, this provides process-level guidance without requiring expensive human-annotated process reward models (Lightman et al., 2023), while complementing prior verification-oriented supervision that does not explicitly target retrieval behavior (Ma et al.). Together, EVALACT and PCAR convert introspection into an executable action space with trainable process signals, improving learning stability and multi-hop generalization.

Our contributions are as follows:

- We propose EVALACT, an RL framework that transforms implicit retrieval quality evaluation into an explicit `Evaluate` action and enforces a coupled `Search`→`Evaluate` protocol, producing dense, trajectory-aligned self-evaluation rewards for tool-using agents.
- We introduce Process-Calibrated Advantage Rescaling (PCAR), a GRPO-based optimization strategy that leverages step-wise self-evaluation scores to refine credit assignment and stabilize learning in long-horizon retrieval trajectories.
- We achieve the best average performance across seven open-domain QA benchmarks with two backbone scales, with particularly strong gains on multi-hop tasks; extensive ablations show that the explicit evaluation loop accounts for the dominant improvements, while PCAR provides consistent additional benefits.

2 Methodology

We present our approach in three parts. First, we formulate retrieval-augmented multi-hop question answering as sequential decision-making under partial observability, providing a unified view for both inference-time interaction and RL training (§2.1). Second, we introduce EVALACT (Evaluate-as-Action), which transforms implicit retrieval quality evaluation into an explicit, policy-selectable action and enforces a coupled `Search`→`Evaluate` interaction protocol (§2.2). Third, we propose Process-Calibrated Advantage Rescaling (PCAR), a GRPO-based optimization method that rescales segment-wise policy gradients using self-evaluation scores, improving credit assignment and stabilizing learning (§2.3) throughout training. Figure 1 illustrates the coupled `Search`→`Evaluate` loop and the PCAR-weighted GRPO update.

2.1 Problem Formulation

Let $\mathcal{M}_\theta$ be an LLM parameterized by θ, inducing a stochastic policy π_θ over textual tokens and tool-mediated actions. We model retrieval-augmented multi-hop question answering as a POMDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P} \rangle$. At time $t = 0, \ldots, T$, the agent samples an action $a_t \sim \pi_\theta(\cdot \mid h_t)$ conditioned on the observable history

$$h_t = [\, x, a_0, o_0, \ldots, a_{t-1}, o_{t-1} \,], \tag{1}$$

where x is the input query and $o_t \in \mathcal{O}$ is the observation returned by the environment after executing a_t. A trajectory is $\tau = \{(a_t, o_t)\}_{t=0}^{T}$, while the underlying state $s_t \in \mathcal{S}$ is unobserved. The transition function $\mathcal{P}(s_{t+1} \mid s_t, a_t)$ governs state evolution and is implicitly defined by the retrieval environment and the agent's reasoning process.

Action space. We partition $\mathcal{A}$ into (i) reasoning tokens $\mathcal{A}_{\text{think}}$, (ii) tool actions $\mathcal{A}_{\text{tool}}$, and (iii) a terminal answer action $\mathcal{A}_{\text{answer}}$. Tool actions include retrieval $\texttt{Search}(q)$ and self-evaluation $\texttt{Evaluate}(c, z)$, where q is a query string, c is a textual assessment, and $z \in [0, 10]$ is a scalar confidence score reported by the policy.

Observations. For tool actions, the environment returns

$$o_t \sim \begin{cases} \mathcal{R}(q), & \text{if } a_t = \texttt{Search}(q), \\ \Phi(z), & \text{if } a_t = \texttt{Evaluate}(c, z), \\ \emptyset, & \text{otherwise}, \end{cases} \tag{2}$$

where $\mathcal{R}(q)$ denotes the top-k retrieved documents and $\Phi(\cdot)$ maps the reported score to a discrete feedback cue used for subsequent decision-making.

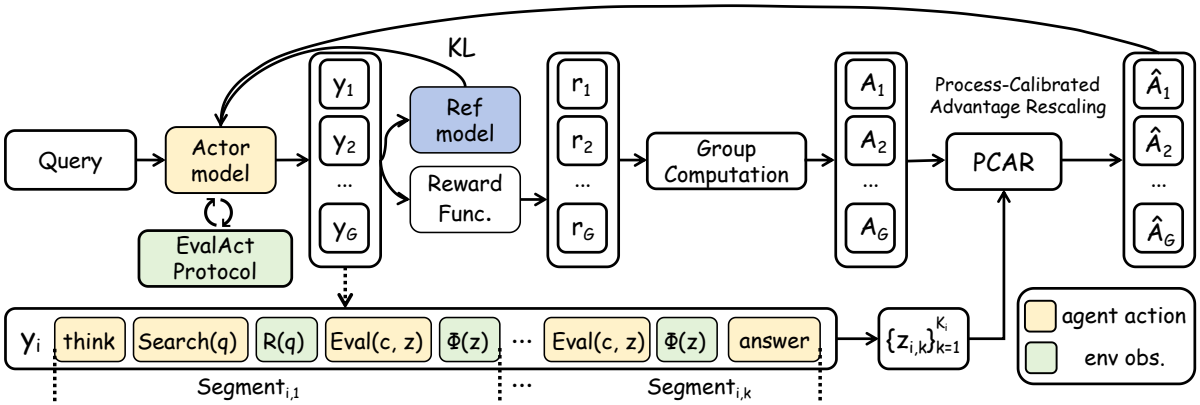

Figure 1: **Overview of EvalAct with PCAR.** The agent follows a coupled `Search`→`Evaluate` protocol, producing segment-wise self-evaluation scores $\{z_{i,k}\}$ that PCAR uses to rescale GRPO advantages.

2.2 EvalAct: Evaluate-as-Action

EvalAct transforms implicit retrieval quality evaluation into an executable action and couples each retrieval step with immediate self-assessment. Specifically, after any retrieval action $a_t = \texttt{Search}(q)$ with observation $o_t = \mathcal{R}(q)$, the agent must invoke exactly one evaluation action $a_{t+1} = \texttt{Evaluate}(c, z)$. The assessment c is conditioned on the retrieved documents, and $z \in [0, 10]$ is a self-reported confidence score. This one-to-one coupling aligns each retrieval result with an explicit reliability assessment, thereby enabling segment-wise training signals.

Inference-time control without oracle signals. To avoid external supervision, the environment-side evaluator is deliberately non-interpretive: it neither parses c nor inspects retrieved documents. Instead, it deterministically maps z to a discrete control cue $\mathcal{I} = \Phi(z)$:

$$\Phi(z) = \begin{cases} \mathcal{I}_{\text{low}}, & z \in [0, 3], \\ \mathcal{I}_{\text{mid}}, & z \in (3, 7], \\ \mathcal{I}_{\text{high}}, & z \in (7, 10]. \end{cases} \tag{3}$$

The cue $\mathcal{I}$ is appended to the context and modulates subsequent actions via instruction conditioning. For completeness, Appendix C specifies the Evaluate format and feedback templates, and Appendix D presents a multi-hop trajectory showing how the agent assigns calibrated self-evaluation scores to guide iterative evidence gathering.

2.3 Reinforcement Learning with PCAR

We optimize π_θ to maximize expected reward, using GRPO as the backbone and PCAR to incorporate process signals from `Evaluate`.

Gated outcome reward. To enforce protocol compliance while optimizing answer quality, we use a gated reward:

$$\mathcal{G}(y, y^*) = \begin{cases} \text{F1}(y_{\text{ans}}, y^*), & \text{if } \mathbb{I}_{\text{fmt}}(y) = 1, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where y_{ans} is the final answer extracted from `<answer>` tags and $\mathbb{I}_{\text{fmt}}(y)$ indicates whether (i) reasoning is enclosed by `<think>` tags and (ii) every `Search` is immediately followed by `Evaluate`.

GRPO. Given G rollouts $\{y_1, \ldots, y_G\}$ sampled from $\pi_{\theta_{\text{old}}}$ for the same input x, we compute group-normalized advantages

$$A_i = \frac{r_i - \mu_{\text{group}}}{\sigma_{\text{group}} + \varepsilon}, \tag{5}$$

where $r_i = \mathcal{G}(y_i, y^*)$ and $\mu_{\text{group}}, \sigma_{\text{group}}$ are the within-group mean and standard deviation.

PCAR: segment-wise advantage rescaling. Standard GRPO applies the same A_i to all tokens, which can inadvertently reinforce unreliable intermediate steps. PCAR instead rescales advantages at the segment level using the self-evaluation scores. Let y_i contain K_i segments, each associated with a score $z_{i,k} \in [0, 10]$. We first compute an intra-trajectory standardized reliability signal

$$\tilde{z}_{i,k} = \frac{z_{i,k} - \mu_i}{\sigma_i + \varepsilon}, \tag{6}$$

where μ_i and σ_i are the mean and standard deviation of $\{z_{i,k}\}_{k=1}^{K_i}$. This normalization makes $\tilde{z}_{i,k}$ reflect relative reliability within the trajectory and suppress trivial constant scoring.

We then define a score-scaled gain

$$\lambda_{i,k} = \lambda_{\text{base}} + (\lambda_{\text{max}} - \lambda_{\text{base}}) \cdot \frac{z_{i,k}}{10}, \tag{7}$$

and compute the token-level calibrated advantage for any token t belonging to segment k:

$$\hat{A}_{i,t} = A_i \cdot \mathrm{clamp}(1 + \lambda_{i,k}\tilde{z}_{i,k},\, \delta,\, \infty)\,, \tag{8}$$

where $\delta > 0$ prevents gradient inversion; we set $\delta = 10^{-6}$ in all experiments unless otherwise specified.

Finally, we maximize the GRPO-style clipped objective with the calibrated advantages:

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{L_i}\Big(\mathcal{L}_{i,t}^{\text{CLIP}} - \beta\,\mathbb{D}_{\text{KL}}(\pi_\theta(\cdot \mid h_t)\,\|\,\pi_{\theta_{\text{ref}}}(\cdot \mid h_t))\Big)\Bigg], \tag{9}$$

where L_i is the length of y_i and

$$\mathcal{L}_{i,t}^{\text{CLIP}} = \min\Big(\rho_{i,t}\hat{A}_{i,t},\, \mathrm{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}\Big), \rho_{i,t} = \frac{\pi_\theta(y_{i,t} \mid h_t)}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid h_t)}. \tag{10}$$

By steering updates toward segments that are outcome-aligned and process-reliable, PCAR improves credit assignment in long-horizon retrieval trajectories. Unlike process reward models (PRMs) that require step-level annotations (Lightman et al., 2023; Wang et al., 2024), PCAR uses self-generated evaluation scores to modulate gradients without external supervision, making it more scalable for retrieval-augmented agents. For reproducibility, Appendix A provides pseudocode for the EvalAct training loop with PCAR, including protocol-compliant rollouts, GRPO advantage estimation, and segment-wise advantage rescaling.

3 Experiments

3.1 Experimental Setup

Datasets. We evaluate open-domain question answering performance on seven widely-used benchmarks spanning both single-hop and multi-hop settings: Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023) as single-hop datasets, and HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023) as multi-hop datasets that typically require complex iterative reasoning and evidence acquisition. For training, we use the publicly released ASearcherBase35K corpus (Gao et al., 2025) and remove invalid or non-actionable samples via lightweight filtering, resulting in 27k instances for RL training. For supervised warm-up, we synthesize 2k protocol-compliant trajectories by prompting DeepSeek-V3.2 (Non-thinking Mode) (Liu et al., 2025) to follow the EvalAct interaction format.

Baselines. We compare EvalAct against representative baselines spanning direct answering, single-pass retrieval augmentation, and multi-step retrieval–reasoning. (1) **Direct Generation** uses the instruction-tuned backbone model to answer using only parametric knowledge, without any retrieval. (2) **Naïve RAG** retrieves documents once and concatenates them with the query, then generates the answer in a single forward pass. (3) **IRCoT** (Trivedi et al., 2023) interleaves retrieval and chain-of-thought prompting for multi-hop reasoning. (4) **Search-o1** (Li et al., 2025) and (5) **Search-R1** (Jin et al., 2025) represent recent search-augmented agentic baselines with iterative retrieval and reasoning. (6) **AutoRefine** (Shi et al., 2025) is an iterative refinement baseline that alternates between evidence gathering and answer refinement. For all retrieval-enabled baselines, we use the same retrieval environment as EvalAct, including the corpus, retriever, returned top-k documents, and search budget, ensuring controlled comparison under matched external evidence access.

Evaluation Metrics. We report Exact Match (EM) as the primary evaluation metric on all benchmarks, computed via exact string matching between the normalized prediction and the reference answer. During RL training, the outcome-level reward is defined as the token-level F1 score between the generated answer and the ground-truth reference (cf. Eq. 4). At test time, performance is evaluated using EM to align with standard open-domain QA evaluation protocols.

3.2 Implementation Details

We conduct experiments with two backbones, Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct. Unless otherwise specified, training uses 8 NVIDIA A100 GPUs with full-parameter optimization and gradient checkpointing. We use a fixed open-domain retrieval environment built on the December 2018 Wikipedia dump with a standard BM25 retriever, without reranking or post-retrieval filtering. At each retrieval step, the top-$k = 3$ documents are returned and appended to the dialogue

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	PopQA	TriviaQA	2Wiki	Bamboogle	HotpotQA	MuSiQue	
Backbone: Qwen2.5-3B-Instruct								
Direct Generation	10.6	10.8	28.8	24.4	2.4	14.9	2.0	13.4
IRCoT	11.1	20.0	31.2	17.1	24.0	16.4	6.7	18.1
Search-o1	23.8	26.2	47.2	21.8	32.0	22.1	5.4	25.5
Naïve RAG	34.8	38.7	54.4	22.6	8.0	25.5	4.7	27.0
Search-R1	34.1	37.8	54.5	31.9	26.4	32.4	10.3	32.5
AutoRefine	**46.7**	**45.0**	62.0	39.4	34.4	40.5	15.7	40.5
EvalAct-3B(ours)	38.5	43.2	**62.1**	**50.0**	**48.0**	**44.3**	**21.6**	**44.0**
Backbone: Qwen2.5-7B-Instruct								
Direct Generation	13.4	14.0	40.8	25.0	12.0	18.3	3.1	18.1
IRCoT	22.4	30.1	47.8	14.9	22.4	13.3	7.2	22.6
Search-o1	15.1	13.1	44.3	17.6	29.6	18.7	5.8	20.6
Naïve RAG	34.9	39.2	58.5	23.5	20.8	29.9	5.8	30.4
Search-R1	39.3	39.7	61.0	41.4	36.8	37.0	14.6	38.5
AutoRefine	**48.4**	**48.7**	**65.9**	40.5	51.2	45.1	18.7	45.5
EvalAct-7B(ours)	38.5	43.6	65.6	**52.1**	**56.0**	**48.8**	**25.3**	**47.1**

Table 1: **Main results (EM, %) on seven open-domain QA benchmarks. Bold** and underlined values indicate the best and second-best performance, respectively.

context. The maximum allowable tool budget is capped at 20 `Search` calls per question.

For RL optimization, we implement the GRPO-based training described in §2.3 with the following default hyperparameters: learning rate 1×10^{-6}, global batch size 256, 2 epochs, 5 rollouts per prompt, rollout temperature 1.0, KL coefficient $\beta = 0.001$, and clip ratio $\epsilon = 0.2$. For PCAR, we set the score-based modulation parameters $(\lambda_{\text{base}}, \lambda_{\text{max}}) = (0.1, 0.5)$ in Eq. 7, which determine the minimum and maximum strength of segment-wise rescaling.

3.3 Main Results

Table 1 summarizes the overall performance across the seven open-domain QA benchmarks. Across both backbone scales, EVALACT consistently achieves the highest average Exact Match (EM) score among all compared methods. Specifically, it reaches an average EM of 44.0% with the EvalAct-3B backbone and 47.1% with the EvalAct-7B backbone. Notably, in both scale settings, it outperforms the strongest competitive baseline by a substantial margin of 3.5 and 1.6 points, respectively.

Comparison with Baselines. Compared with Search-o1, Search-R1, IRCoT, and Naïve RAG, EVALACT consistently outperforms these baselines on the majority of benchmarks, with the largest gains appearing in multi-hop settings. This trend is consistent across both 3B and 7B backbones, indicating that the improvements are robust across model scales. Unlike prior approaches that rely on implicit self-correction within free-form reasoning, EVALACT explicitly models evaluation as a discrete action, enabling segment-level credit assignment during RL optimization.

Multi-Hop Benchmarks. The strongest gains of EVALACT emerge on multi-hop datasets. Across both backbone scales, EVALACT achieves the best performance on all four multi-hop benchmarks: 2WikiMultihopQA, Bamboogle, HotpotQA, and MuSiQue. The gains are especially large on 2WikiMultihopQA and Bamboogle, where EvalAct-3B improves over AutoRefine by 10.6 and 13.6 points, respectively, and EvalAct-7B improves over the strongest baseline by 10.7 and 4.8 points. Consistent improvements are also observed on HotpotQA and MuSiQue. These results suggest that explicit intermediate evaluation is particularly beneficial for tasks requiring iterative evidence aggregation and long-horizon reasoning, where the coupled evaluation loop helps control error propagation across extended interaction sequences. The larger gains on multi-hop tasks can be attributed to more retrieval steps providing more segments for PCAR to reweight, whereas single-retrieval scenarios offer

limited opportunities for segment-level rescaling. We further validate these findings with extended scaling experiments (Appendix E) and real web search evaluation (Appendix F).

Single-Hop Benchmarks. On these single-hop datasets, EVALACT remains competitive but does not consistently outperform AutoRefine, which performs better on NQ and PopQA. This is expected: AutoRefine is designed for iterative answer refinement, which is particularly effective when the main challenge is answer polishing rather than multi-step evidence accumulation in single-hop settings. Nevertheless, the substantial gains on multi-hop benchmarks outweigh these gaps, resulting in the best overall average performance.

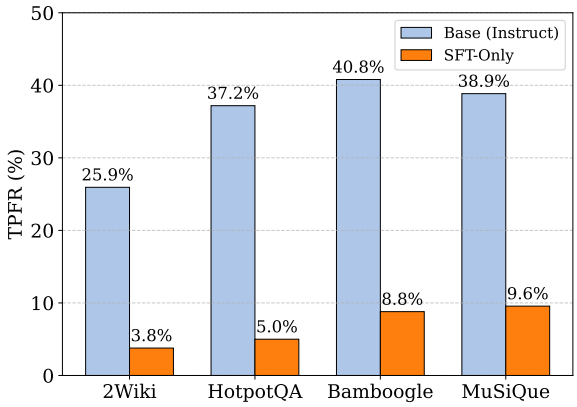

Figure 2: **Effect of SFT on format alignment.** SFT warm-up substantially reduces tool parsing failure rates (TPFR), stabilizing structured tool use for RL.

4 Analysis

We conduct ablation studies to examine the contribution of each component in EVALACT. Figure 3 provides an overview: (a) training curves showing stable convergence for both 3B/7B backbones, (b) model ablation across training variants, (c) method ablation isolating the evaluation loop and PCAR, and (d) sensitivity analysis of PCAR hyperparameters. The following subsections provide details.

4.1 Model Ablation: Disentangling Format Alignment from Reasoning

A prerequisite for EVALACT is protocol compliance: the agent must reliably produce well-formed tool calls and adhere to the strictly coupled `Search`→`Evaluate` loop. To disentangle the contributions of format acquisition (via SFT) from reasoning capability (via RL), we construct six training variants based on the Qwen2.5-3B-Instruct backbone and systematically evaluate them on four multi-hop benchmarks.

The variants are defined as follows:

- **Base (Instruct) / SFT-Only:** the backbone model evaluated without and with supervised warm-up, respectively.
- **Base + RL / SFT + RL (Vanilla):** standard GRPO optimizing for answer correctness without enforcing the explicit `Evaluate` loop; retrieval tools are invoked freely.
- **Base EvalAct:** applied directly to the Base model without SFT warm-up.
- **EvalAct (Ours):** the full pipeline, i.e., SFT warm-up followed by EVALACT RL training.

Model Variant	SFT	RL Paradigm	Avg. EM
Base (Instruct)	✗	-	14.2
SFT-Only	✓	-	24.8
Base + RL	✗	GRPO	33.1
SFT + RL	✓	GRPO	33.5
Base EvalAct	✗	**EvalAct**	17.1
EvalAct (Ours)	✓	**EvalAct**	**41.0**

Table 2: Ablation of training stages and paradigms on multi-hop benchmarks (Avg. EM, %). *Vanilla* denotes standard RL without the explicit evaluation loop.

SFT for Format Alignment. We first examine the role of supervised warm-up in establishing protocol compliance. As shown in Figure 2, SFT substantially reduces tool-formatting and parsing failures, providing a stable initialization for structured tool use. Consistent with this observation, SFT alone improves the multi-hop average from 14.2% to 24.8% (Table 2). Under vanilla RL without the explicit evaluation loop, performance is only weakly affected by SFT initialization, reaching 33.1% from the Base model and 33.5% from the SFT-initialized model. This suggests that standard RL can eventually recover a functional tool-calling policy, whereas SFT primarily stabilizes early optimization by aligning the model with the required format.

Effectiveness of the EvalAct Paradigm. We next isolate the effect of enforcing the coupled `Search`→`Evaluate` loop. With the same backbone and training budget, EVALACT (OURS) attains

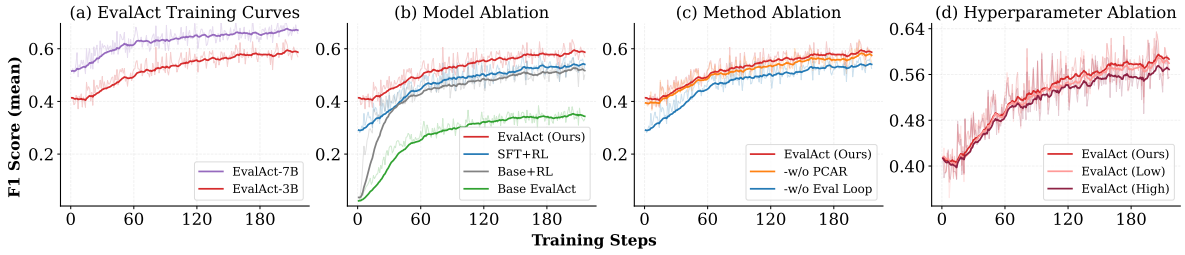

Figure 3: **Training curves and ablation overview.** (a) Training curves of EVALACT with 3B/7B backbones. (b) Model ablation across training variants. (c) Method ablation isolating the evaluation loop and PCAR. (d) Sensitivity to PCAR rescaling intensity.

41.0% average EM, exceeding the strongest vanilla baseline (33.5%) by +7.5 points (Table 2). This gain supports the hypothesis that converting intermediate evaluation into an explicit action yields more informative process signals than implicit verification under outcome-only optimization. By contrast, applying EVALACT directly without supervised warm-up yields only 17.1%, highlighting the difficulty of learning a structured action protocol from scratch.

4.2 Method Ablation: Dissecting Structural and Optimization Components

We further decompose the performance gains of EVALACT into two sources: the structural contribution of the explicit evaluation loop and the optimization contribution of Process-Calibrated Advantage Rescaling (PCAR). We compare the full model against two ablated variants: (1) **w/o Eval Loop:** removing `Evaluate` and reverting to a standard retrieval policy optimized via vanilla GRPO (equivalent to *SFT+RL* in §4.1); (2) **w/o PCAR:** retaining the `Evaluate` structure and confidence scores z, but optimizing with standard GRPO without confidence-based advantage rescaling.

Method	2Wiki	Bam.	Hotpot	Mus.	Avg.
EvalAct	**50.0**	**48.0**	**44.3**	**21.6**	**41.0**
w/o PCAR	48.2	46.2	44.1	20.8	39.8
w/o Eval Loop	41.4	37.2	38.9	16.5	33.5

Table 3: Method ablation isolating the evaluation loop and PCAR on multi-hop benchmarks (EM, %).

Structural Contribution of the Evaluation Loop. Removing the explicit evaluation mechanism causes the largest overall performance degradation. As shown in Table 3, eliminating `Evaluate` lowers average EM from 41.0% to 33.5%, a drop of 7.5 points. This degradation is consistent across all four benchmarks, with especially pronounced declines on 2WikiMultihopQA (−8.6) and Bamboogle (−10.8). These empirical results suggest that the primary benefit of EVALACT lies in its explicit evaluation loop, which enforces intermediate verification and thereby reduces error propagation in multi-hop reasoning.

Optimization Contribution of PCAR. Beyond the structural benefit of the explicit evaluation loop, PCAR provides additional optimization gains. Compared with standard GRPO applied to the same evaluation-augmented framework, PCAR raises average EM from 39.8% to 41.0%, a gain of 1.2 points. Improvements are observed on all four benchmarks, with gains of 1.8 points on both 2WikiMultihopQA and Bamboogle, and smaller but consistent improvements on HotpotQA and MuSiQue. These results indicate that confidence-aware advantage rescaling complements the explicit evaluation structure by providing more informative gradient signals for segments with varying reliability estimates.

4.3 Hyperparameter Sensitivity

We analyze the sensitivity of PCAR to the rescaling intensity governed by λ_{base} and $\lambda_{\max}$. To characterize the strength of reliability-aware modulation, we define the **Relative Importance Ratio (RIR)** as the ratio between the maximum and minimum attainable advantage multipliers under full-confidence conditions. When the standardized reliability score satisfies $\tilde{z} \in [-1, 1]$, the unclamped multiplier spans $1 \pm \lambda_{\max}$; after applying the lower-bound clamp with threshold δ, the effective ratio is approximated by $\text{RIR} \approx (1 + \lambda_{\max}) / \max(\delta,\, 1 - \lambda_{\max})$.

Table 4 reports three representative configurations corresponding to low, moderate, and high rescaling intensity. Performance remains relatively stable across settings, with average EM ranging

Setting	Params	RIR	Avg.	Δ
EvalAct (Low)	0.05/0.25	1.67	39.8	-0.3
EvalAct (Mid)	**0.10/0.50**	**3.0**	**40.1**	-
EvalAct (High)	0.20/1.00	200	39.4	-0.7

Table 4: Sensitivity to PCAR rescaling intensity. **RIR**: ratio of max to min advantage multipliers.

from 39.4% to 40.1%. The moderate configuration (*EvalAct (Mid)*, RIR $= 3.0$) achieves the best overall performance.

Under conservative rescaling (*Low*, RIR $= 1.67$), the separation between high- and low-reliability segments is limited, potentially reducing the effectiveness of segment-level credit assignment. Conversely, aggressive rescaling (*High*, RIR $= 200$) slightly degrades performance. In this regime, the minimum multiplier approaches zero and is clipped by the lower-bound constraint, leading to highly imbalanced gradient magnitudes across segments. Such extreme modulation can restrict corrective updates on low-reliability steps and destabilize optimization.

Overall, these results suggest moderate reliability rescaling provides a balanced trade-off between emphasizing high-confidence segments and preserving sufficient gradient flow for error correction.

4.4 Self-Evaluation Reliability

A core assumption of EVALACT is that self-evaluation scores reflect actual retrieval quality. We validate this through groundtruth recall analysis: for each retrieval step, we check whether the reference answer appears in the retrieved documents and correlate this with the self-reported score. Across 22,084 retrieval steps from the 7B-RL model on seven QA benchmarks, high-confidence retrievals (score ≥ 7) contain the groundtruth answer 54.68% of the time, compared to only 24.51% for low-confidence retrievals (score ≤ 4)—a 30.17 percentage point gap ($\chi^2 = 1829.4$, $p < 0.001$). This strong association indicates that self-scores are meaningfully aligned with evidence relevance. Furthermore, behavioral analysis shows that trajectories ending with high self-scores use significantly fewer searches (4.86 vs. 13.01 on average, $t = -39.0$, $p < 0.001$), suggesting models stop searching when sufficient evidence is gathered. Complete calibration data, correlation analyses, and statistical details are provided in Appendix G.

4.5 Efficiency Analysis

Despite introducing explicit `Evaluate` actions, RL training yields *dual improvements*: higher accuracy with reduced computational overhead. Comparing SFT and RL stages on the 7B backbone across all benchmarks (32,625 predictions), RL achieves +15.6pp EM accuracy while reducing average searches by 11.5% (5.2→4.6) and generation tokens by 10.4% (1321→1183). Similar patterns hold for 3B and 14B models (Appendix H). This efficiency gain arises because PCAR incentivizes strategic search behavior: models learn to stop when evidence is sufficient rather than exhausting the search budget.

5 Conclusion

We presented EVALACT, a framework that elevates retrieval evaluation from an implicit reasoning behavior to an explicit policy action. This design enables retrieval-augmented agents to generate structured process signals during interaction for more fine-grained reinforcement learning. Built on this framework, PCAR further improves optimization by aligning policy updates with segment-level reliability estimates.

Across seven open-domain QA benchmarks and multiple model scales (3B/7B/14B), EVALACT delivers the best average results with the largest advantages on multi-hop reasoning tasks. Experiments on real web search benchmarks confirm that the benefits generalize beyond static retrieval environments. Notably, the explicit evaluation loop also yields efficiency gains: RL models achieve higher accuracy while using fewer searches, as PCAR incentivizes strategic evidence gathering.

These findings highlight the value of converting intermediate evaluation into a trainable action for multi-step retrieval-augmented reasoning.

Limitations

While EVALACT and PCAR yield consistent gains in our experiments, the current empirical study is restricted to a relatively compact model regime. Owing to the cost of full-parameter RL training and systematic ablation, we have not yet evaluated the framework on substantially larger language models. Overall, our findings provide empirical evidence that introducing explicit self-evaluation as process-level supervision can offer clear benefits over outcome-only GRPO within the model regime studied in this work.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, and 1 others. 2025. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. 2025. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl. *arXiv preprint arXiv:2508.07976*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

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.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pages 874–880.

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.

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. *arXiv preprint arXiv:2503.09516*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4745–4759.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

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. *arXiv preprint arXiv:2501.05366*.

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

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

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

Ruotian Ma, Peisong Wang, Cheng Liu, Xingyan Liu, Jiaqi Chen, Bang Zhang, Xin Zhou, Nan Du, and Jia Li. S2r: Teaching llms to self-verify and selfcorrect via reinforcement learning, 2025. *URL https://arxiv.org/abs/2502.12853*, 4:17.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

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. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

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

Yaorui Shi, Sihang Li, Chang Wu, Zhiyuan Liu, Junfeng Fang, Hengxing Cai, An Zhang, and Xiang Wang. 2025. Search and refine during think: Facilitating knowledge refinement for improved retrieval-augmented reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. volume 36, pages 8634–8652.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher Manning, and Chelsea Finn. 2023. Fine-tuning language models for factuality. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

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.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 10014–10037.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Xie. 2023. Self-evaluation guided beam search for reasoning. *Advances in Neural Information Processing Systems*, 36:41618–41650.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Qi Zhang, Shouqing Yang, Lirong Gao, Hao Chen, Xiaomeng Hu, Jinglei Chen, Jiexiang Wang, Sheng Guo, Bo Zheng, Haobo Wang, and 1 others. 2025a. Lets: Learning to think-and-search via process-and-outcome reward hybridization. *arXiv preprint arXiv:2505.17447*.

Yuxiang Zhang, Jiangming Shu, Ye Ma, Xueyuan Lin, Shangxi Wu, and Jitao Sang. 2025b. Memory as action: Autonomous context curation for long-horizon agentic tasks. *arXiv preprint arXiv:2510.12635*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In

International conference on machine learning, pages 12697–12706. PMLR.

A Pseudocode for EVALACT with PCAR

Algorithm 1 EVALACT Training with PCAR (GRPO Backbone)

Input: policy π_θ, reference $\pi_{\theta_{\text{ref}}}$, dataset $\mathcal{D}$, environment $\mathcal{E}$,
  rollouts per input G, clip ϵ, KL weight β, constant ϵ_0,
  PCAR params $(\lambda_{\text{base}}, \lambda_{\text{max}}, \delta)$
Output: optimized policy π_θ
while not converged **do**
  Sample a batch of queries $\mathcal{X} \sim \mathcal{D}$
  Initialize training buffer $\mathcal{B} \leftarrow \emptyset$
  for all $x \in \mathcal{X}$ **do**
    // Protocol-compliant rollouts
    for $i = 1$ **to** G **do**
      Sample trajectory $y_i \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$ under coupled `Search`$\rightarrow$`Evaluate`
      Compute gated reward $r_i \leftarrow \mathcal{G}(y_i, y^*)$ (Eq. (4))
      Segment y_i into $\{\sigma_{i,k}\}_{k=1}^{K_i}$ aligned with `Search`$\rightarrow$`Evaluate`, record $\{z_{i,k}\}_{k=1}^{K_i}$
    end for
    // GRPO advantage
    $\mu_{\text{group}} \leftarrow \text{mean}(\{r_i\}_{i=1}^{G})$, $\sigma_{\text{group}} \leftarrow \text{std}(\{r_i\}_{i=1}^{G}) + \epsilon_0$
    for $i = 1$ **to** G **do**
      $A_i \leftarrow (r_i - \mu_{\text{group}})/\sigma_{\text{group}}$
      Compute calibrated advantages $\{\hat{A}_{i,t}\}$ for tokens in y_i via PCAR:
        $\tilde{z}_{i,k} \leftarrow (z_{i,k} - \mu_i)/(\sigma_i + \epsilon_0)$, $\lambda_{i,k} \leftarrow \lambda_{\text{base}} + (\lambda_{\text{max}} - \lambda_{\text{base}}) \cdot z_{i,k}/10$
        $\hat{A}_{i,t} \leftarrow A_i \cdot \text{clamp}(1 + \lambda_{i,k}\tilde{z}_{i,k}, \delta, \infty)$ for $t \in \sigma_{i,k}$
      Add token-level instances from (x, y_i) with advantages $\{\hat{A}_{i,t}\}$ to $\mathcal{B}$
    end for
  end for
  Update π_θ by maximizing the clipped objective with KL regularization (Eq. (9)) on $\mathcal{B}$
end while
return π_θ

B Related Work

B.1 Retrieval-Augmented Language Models

Retrieval-augmented generation (RAG) enhances LLMs by grounding generation in externally retrieved knowledge (Lewis et al., 2020; Guu et al., 2020). Early work focused on improving retrieval quality through dense encoders (Karpukhin et al., 2020; Izacard and Grave, 2021) or neural rerankers (Nogueira and Cho, 2019). More recent approaches integrate retrieval directly into the reasoning process, enabling models to iteratively query external sources (Yao et al., 2022; Trivedi et al., 2023; Jiang et al., 2023). Self-RAG (Asai et al., 2024) introduces special tokens for assessing retrieval utility, though these remain implicit signals rather than structured actions. Our work builds on this line by converting retrieval evaluation into an explicit, structured action with discrete scores that can serve as training signals.

B.2 Reinforcement Learning for Tool-Using Agents

RL has emerged as a promising approach for training LLM agents to use tools effectively (Schick et al., 2023; Qin et al., 2023). Search-R1 (Jin et al., 2025) demonstrates that pure RL with outcome rewards can train effective retrieval policies without supervised fine-tuning. However, outcome-only rewards suffer from the credit assignment problem in multi-step trajectories. Process reward models (PRMs) address this by providing step-level supervision (Lightman et al., 2023; Wang et al., 2024), but typically require expensive human annotations or external verifiers that may not align with the target policy. LeTS (Zhang et al., 2025a) designs retrieval-specific process rewards based on knowledge redundancy and exact match, yet relies on heuristics that may not generalize.

Recent work has begun to expand the action space by converting traditionally implicit behaviors into explicit, learnable decisions. MemAct (Zhang et al., 2025b) formulates working memory management as policy actions for context deletion and insertion, enabling end-to-end RL over long-horizon context curation. EVALACT shares this action-centric perspective but targets a different behavior: instead of treating retrieval quality assessment as an implicit part of free-form reasoning, we convert evaluation into an explicit action that produces process signals for fine-grained credit assignment.

B.3 Self-Evaluation and Calibration

LLMs can evaluate their own outputs (Kadavath et al., 2022; Xie et al., 2023; Madaan et al., 2023), but such assessments vary in calibration (Tian et al., 2023). Uncalibrated evaluation during RL training risks reward hacking. Prior work addresses calibration through specialized training (Lin et al., 2022), prompting (Xiong et al., 2023), or post-hoc adjustments (Zhao et al., 2021). PCAR takes a different approach by designing the RL objective such that miscalibrated confidence incurs penalties through the advantage signal. Unlike self-reflection methods (Shinn et al., 2023; Madaan et al., 2023) that evaluate retrospectively, EVALACT performs inline evaluation after each retrieval. Unlike external pro-

cess supervision (Wang et al., 2024; Lightman et al., 2023) requiring oracle annotations, EVALACT generates self-supervised signals without external labels.

C Evaluate Specification and Feedback Templates

`Evaluate` Action Format. After each retrieval step, the agent invokes `Evaluate`(c, z), where c is a free-form textual assessment of the immediately preceding `Search` output, and $z \in [0, 10]$ denotes a scalar self-reported confidence score. The environment intercepts this action and returns a discrete control cue $\mathcal{I} = \Phi(z)$, which is appended to the context to modulate subsequent decision-making.

Discrete Feedback Mapping. We instantiate $\Phi(\cdot)$ as a deterministic, three-tier binning strategy:

$$\Phi(z) = \begin{cases} \mathcal{I}_{\text{low}}, & z \in [0, 3], \\ \mathcal{I}_{\text{mid}}, & z \in (3, 7], \\ \mathcal{I}_{\text{high}}, & z \in (7, 10]. \end{cases} \tag{11}$$

Feedback Templates. The control cue $\mathcal{I}$ is materialized as an instruction-style message that conditions the agent's next action. The exact textual templates returned by the environment are detailed below.

Environment Feedback Templates

$\mathcal{I}_{\text{low}}$ **(Low Quality** | $z \in [0, 3]$**)**
"The previous `Search` *results are largely irrelevant or unhelpful for answering the question. Do not rely on them. Reformulate the query (e.g., alternative keywords or a different angle) and issue a new* `Search`*."*

$\mathcal{I}_{\text{mid}}$ **(Medium Quality** | $z \in (3, 7]$**)**
"The previous `Search` *results contain partially useful evidence but may be incomplete or noisy. Use only clearly relevant excerpts. Consider an additional, more targeted* `Search` *to fill missing details, resolve remaining subproblems, or verify uncertain information."*

$\mathcal{I}_{\text{high}}$ **(High Quality** | $z \in (7, 10]$**)**
"The previous `Search` *results are highly relevant and constitute substantive progress toward answering the question (e.g., providing key facts or resolving an important subtask). Use them as primary evidence to construct the answer. Perform another* `Search` *only if a specific critical detail is still missing."*

D Case Study: Multi-Hop Reasoning Trajectory

We present a full multi-hop interaction trajectory to illustrate the EVALACT framework in practice. The example highlights calibrated self-evaluation: a partial retrieval receives 5/10, while a conclusive retrieval receives 10/10. These step-wise confidence signals are the same signals used by PCAR for segment-level advantage rescaling during RL.

EvalAct Trajectory Example

User: In between *Remember the Titans* and *My Favorite Martian* which film grossed $36.8 million domestically?

Agent:
`<think>`I need to find out which film grossed $36.8 million domestically between "Remember the Titans" and "My Favorite Martian". I should search for box office information for both films.`</think>`

Tool Call [Search]:
`{"query": "Remember the Titans domestic box office $36.8 million"}`

Observation [Search]:
[Doc 1 & 3 omitted for brevity]
Doc 2 (Title: "Remember the Titans"): "...Remember the Titans opened strongly at the U.S. box office... It eventually went on to gross an estimated $115,654,751 in the U.S., and..."

Tool Call [Evaluate]:
```
{
  "evaluation": "The search provided information about Remember the Titans' box office performance, but didn't specifically mention the $36.8 million figure. The results show it grossed $115,654,751 in the U.S., which is much higher than $36.8 million. This helps narrow down that Remember the Titans didn't gross $36.8 million, but I need more specific information to compare with My Favorite Martian.",
  "score": 5
}
```

Observation [Evaluate]:
Score 5/10 (Medium Quality): The previous Search results are partially helpful but may be incomplete or contain noise. Extract and use only the clearly relevant parts. Consider running another more focused Search to fill in missing details, resolve remaining subproblems, or verify uncertain information.

Agent:
`<think>`Now I need to search for My Favorite Martian's box office information.`</think>`

Tool Call [Search]:
`{"query": "\"My Favorite Martian\" domestic box office $36.8 million"}`

Observation [Search]:
[Doc 1 & 2 omitted for brevity]
Doc 3 (Title: "My Favorite Martian (film)"): "...The film grossed $36.8 million domestically against a budget of $65 million..."

Tool Call [Evaluate]:
```
{
  "evaluation": "This search was highly successful. Document 3 directly mentions
```

```
\"My Favorite Martian\" and states it grossed $36.8 million domestically against a budget of $65 million. This matches exactly the figure mentioned in the question and confirms this film's domestic gross. Combined with the previous search showing Remember the Titans grossed much more than $36.8 million ($115+ million), this gives me clear information to answer which film grossed the specified amount.",
  "score": 10
}
```

Observation [Evaluate]:
Score 10/10 (High Quality): The previous Search results are highly relevant and make strong progress toward answering the question... Use them as primary evidence to construct your answer. Only conduct another Search if you still need a very specific missing detail.

Agent:
<think>From the information gathered:
1. Remember the Titans grossed $115,654,751 in the U.S. (much higher than $36.8 million)
2. My Favorite Martian grossed $36.8 million domestically
Therefore, My Favorite Martian is the film that grossed $36.8 million domestically.</think>

<answer>My Favorite Martian</answer>

E Extended Scaling Results

To validate the scalability of EvalAct, we extend experiments to Qwen2.5-14B-Instruct across all seven QA benchmarks. Table 5 reports the full results, demonstrating that RL benefits persist at larger scale.

Scale	Base	SFT	RL	Δ (RL−Base)
3B	16.7%	32.7%	43.7%	+27.0pp (+162%)
7B	31.4%	34.6%	46.5%	+15.1pp (+48%)
14B	39.8%	41.1%	45.8%	+6.0pp (+15%)

Table 5: Cross-scale comparison (average EM on 7 QA benchmarks). RL benefits persist at 14B; smaller relative gains reflect stronger baselines rather than method saturation.

While relative improvements decrease at 14B (15% vs. 162% at 3B), this reflects the substantially stronger baseline (39.8% vs. 16.7%) rather than method saturation. Notably, 7B achieves the highest absolute RL accuracy (46.5%), representing a favorable accuracy–efficiency trade-off for practical deployment.

F Real Web Search Experiments

To validate cross-environment robustness, we evaluate EvalAct on two challenging real web search benchmarks using the Jina Search API for live internet retrieval.

Datasets. **FRAMES** (Krishna et al., 2025) requires time-sensitive, multi-source reasoning over dynamic web content (200 samples). **BrowseComp+** (Chen et al., 2025) presents extreme-difficulty queries requiring 8–12 web searches (200 samples). Both datasets test generalization beyond the static Wikipedia corpus used during training.

Experimental Setup. All models use identical settings: maximum 20 searches per trajectory, Jina Search API with fixed parameters, no model-specific tuning. We report both LLM-judge accuracy (GPT-OSS-120B for semantic matching) and strict Exact Match (EM) to address concerns about evaluation bias.

Dataset	Scale	Base	SFT	RL	Δ
FRAMES (Judge)	3B	8.96	27.86	27.36	+18.40pp
	7B	28.36	36.82	43.78	+15.42pp
	14B	35.32	50.75	**60.80**	+25.48pp
FRAMES (EM)	3B	4.00	15.00	17.00	+13.00pp
	7B	14.00	22.00	25.00	+11.00pp
	14B	23.00	27.00	**30.50**	+7.50pp
BrowseComp+ (Judge)	3B	0.50	2.49	3.48	+2.98pp
	7B	1.99	3.50	4.50	+2.51pp
	14B	2.99	5.47	**7.00**	+4.01pp
BrowseComp+ (EM)	3B	0.50	1.50	2.50	+2.00pp
	7B	0.50	1.50	2.50	+2.00pp
	14B	2.50	3.50	**4.50**	+2.00pp

Table 6: Real web search results. RL benefits persist in dynamic retrieval environments under both LLM-judge and strict EM evaluation.

Results in Table 6 confirm that RL benefits generalize to dynamic web retrieval environments. The 14B-RL model achieves 60.80% (Judge) / 30.50% (EM) on FRAMES, demonstrating robust performance on time-sensitive multi-source reasoning. Gains persist under strict EM, ruling out LLM-judge bias as the source of improvement.

G Self-Evaluation Reliability Analysis

We provide comprehensive validation that self-evaluation scores reflect actual retrieval quality.

G.1 Calibration Analysis

We measure calibration as the Mean Absolute Error (MAE) between self-reported scores and external evaluator scores (GPT-OSS-120B), computed across 21,750 samples (143,000+ retrieval steps) spanning 6 configurations (3B/7B/14B × SFT/RL) on 7 datasets.

Model	SFT MAE	RL MAE	Improvement
3B	1.792	1.695	-5.4%
7B	1.583	1.469	-7.2%
14B	1.340	1.317	-1.7%

Table 7: Calibration MAE (lower is better). RL consistently reduces calibration error across all scales.

G.2 Correlation with Final Correctness

We compute point-biserial correlations between aggregated self-scores and binary EM correctness.

Model	SFT r	RL r	Significance
3B	0.319	0.345	$p < 0.0001$
7B	0.362	0.384	$p < 0.0001$
14B	0.392	0.413	$p < 0.0001$

Table 8: Point-biserial correlation between self-scores and EM correctness. All correlations exceed $r = 0.3$ (medium-to-strong) with $p < 0.0001$.

G.3 Groundtruth Recall Analysis

We analyze 22,084 retrieval steps from the 7B-RL model, checking whether groundtruth answers appear in retrieved documents via string matching.

Self-Score	Steps	GT Recall	Test
High ($\geq$7)	8,217	54.68%	$\chi^2 = 1829.4$
Low ($\leq$4)	11,025	24.51%	$p < 0.001$
Difference	—	**+30.17pp**	(highly significant)

Table 9: Groundtruth recall by self-score category. High-confidence retrievals are 2.2$\times$ more likely to contain the answer.

G.4 Behavioral Analysis

We compare search behavior across 2,784 trajectories grouped by final self-score.

Final Score	Traj.	Avg. Searches	Test
High ($\geq$8)	2,034	4.86	$t = -39.0$
Low ($\leq$5)	750	13.01	$p < 0.001$
Difference	—	**-8.15**	(significant)

Table 10: Search count by final self-score. High-confidence trajectories use 62% fewer searches.

H Efficiency Analysis

We report detailed efficiency metrics across all scales, measuring search count and generation tokens (Qwen2.5 tokenizer) on 32,625 predictions.

Scale	Metric	SFT	RL	Δ
3B	EM Accuracy	32.5%	43.6%	+11.1pp
	Avg. Searches	4.65	4.20	-9.7%
	Avg. Tokens	1190	984	-17.3%
7B	EM Accuracy	29.9%	45.5%	+15.6pp
	Avg. Searches	5.20	4.60	-11.5%
	Avg. Tokens	1321	1183	-10.4%
14B	EM Accuracy	39.9%	44.7%	+4.8pp
	Avg. Searches	5.25	5.00	-4.8%
	Avg. Tokens	1382	1217	-11.9%

Table 11: Efficiency analysis across scales. RL achieves dual improvements: higher accuracy with reduced search count and token generation.

Across all scales, RL achieves substantial accuracy gains (+4.8pp to +15.6pp) while simultaneously reducing both search count (-4.8% to -11.5%) and token generation (-10.4% to -17.3%). The additional `Evaluate` actions are offset by more efficient search behavior: PCAR incentivizes models to stop when evidence is sufficient rather than exhausting the search budget.