GROUNDED-R1: REINFORCEMENT LEARNING FOR PREEMPTIVE HALLUCINATION INTERVENTION AND GROUNDED GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Hallucination in large language models (LLMs) often begins locally: an ungrounded claim can enter the generation trajectory, steer subsequent reasoning, and compound into a global failure. Retrieval-augmented generation (RAG) can reduce such failures by supplying external evidence, but to prevent local error propagation, evidence must arrive before risky claims propagate. However, existing methods do not directly learn retrieval trigger timing from local hallucination risk. Retrieval is driven either by heuristic rules or fixed procedures in training-free methods or by delayed, coarse indirect rewards in reinforcement learning (RL)-based approaches, causing evidence to arrive too late to prevent local error propagation while unnecessary calls waste the retrieval budget. In this paper, we propose **Grounded-R1**, a dynamic RAG framework for preemptive hallucination intervention. Grounded-R1 introduces the Real-time Hallucination Risk Detection (RHRD) module to estimate sentence-level hallucination risk from generation-time signals and uses two-phase RL to align retrieval trigger timing with this risk. Across eight knowledge-intensive benchmarks, Grounded-R1 improves final-answer exact match (EM) by 18.0% over the strongest external baseline with comparable retrieval calls. On five open-ended benchmarks, it raises sentence-level non-hallucination rate (NHR) by 12.7 points. Delivering the same evidence one sentence earlier reduces next-sentence hallucination rate by 11.1 points; under equal retrieval budgets, Grounded-R1 targets positions with an observed 8.3-point downstream hallucination reduction. Grounded-R1 thus improves grounded generation not by retrieving more, but by retrieving where one sentence of delay matters. Our code and models are available at `https://sites.google.com/view/grounded-r1`.

1 INTRODUCTION

Large language models (LLMs) excel at reasoning and knowledge-intensive tasks, yet frequently generate fluent but ungrounded claims, known as *hallucination*. Because generation is autoregressive, hallucination can emerge before the final answer. A local risky claim can enter the trajectory, contaminate subsequent reasoning, and ultimately distort the global output. Consequently, a model might produce a plausible final answer while leaving intermediate claims unsupported, masking the fragility of its reasoning path. Final-answer correctness is therefore an incomplete proxy for grounded generation (Ji et al., 2023; Huang et al., 2025; Min et al., 2023).

Retrieval-augmented generation (RAG) grounds LLMs with external evidence (Lewis et al., 2020; Klesel & Wittmann, 2025), but evidence access alone does not guarantee grounded generation throughout the reasoning process. To keep a generation trajectory grounded, external evidence must support not only the final answer but also the intermediate claims that arise during generation. Static RAG retrieves only before generation, so its fixed evidence cannot cover later-emerging information needs or intervene when local risk appears mid-trajectory (Lewis et al., 2020; Jiang et al., 2023; Su et al., 2024). While dynamic RAG permits mid-trajectory retrieval, it shifts the core bottleneck from evidence *access* to retrieval *timing*, requiring the model to identify when an emerging local risk demands external evidence.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Existing dynamic RAG has not fully addressed this timing problem. Training-free or procedure-driven methods rely on heuristic signals, empirical thresholds, fixed schedules, or prompted workflows to trigger retrieval (Jiang et al., 2023; Su et al., 2024; Yao et al., 2025; Li et al., 2026; Trivedi et al., 2023; Li et al., 2025). Although these methods show that retrieval timing matters, their trigger mechanisms are not directly learned from downstream grounded generation and are therefore difficult to recalibrate under context shift or noisy evidence. A threshold rule over a risk score can partially localize high-risk positions, but it still reduces retrieval trigger timing to myopic scalar gating rather than a policy over evolving generation states, retrieved evidence, and retrieval cost. Conversely, reinforcement learning-based RAG (RL-based RAG) makes retrieval behavior learnable through interaction (Tan et al., 2026; Jin et al., 2025; Song et al., 2025; Zhang et al., 2025). However, their rewards target answer outcomes or intermediate think-and-search quality rather than retrieval trigger timing aligned with local hallucination risk. Such supervision provides only indirect credit for when retrieval should be triggered or suppressed. Consequently, both paradigms remain weakly aligned with local hallucination risk, causing retrieval to either arrive too late to prevent local error propagation or trigger unnecessarily and waste the retrieval budget.

Motivated by this gap, we propose **Grounded-R1**, a dynamic RAG framework that reframes retrieval as a risk-aware intervention for grounded generation. The key shift moves beyond outcome-level supervision to optimize whether retrieval is invoked at each sentence-end decision point, allowing evidence requests before a high-risk local claim shapes subsequent reasoning. Grounded-R1 has two components. First, **Real-time Hallucination Risk Detection (RHRD)** estimates sentence-level hallucination risk from generation-time signals. Second, a two-phase reinforcement learning (RL) framework uses this risk signal for policy learning. Phase I learns high-quality, low-cost generation with an episode-level quality-cost reward; Phase II uses RHRD-aligned sentence-level rewards to optimize retrieval trigger timing while preserving the base generation capability learned in Phase I.

Because our claim concerns local grounded generation rather than final-answer accuracy alone, we evaluate Grounded-R1 along three complementary axes: final-answer quality, sentence-level groundedness, and causal retrieval trigger timing. Across eight benchmarks, Grounded-R1 improves macro-average final-answer exact match (EM) by 18.0% over the strongest external baseline with comparable retrieval calls. On five benchmarks under open-ended generation, it improves sentence-level non-hallucination rate (NHR) by 12.7 points. Delivering the same evidence one sentence earlier cuts the first downstream sentence hallucination rate by 11.1 points; under equal retrieval budgets, Grounded-R1 targets positions with an observed 8.3-point downstream hallucination reduction.

Our contributions are fourfold. **First,** we formulate hallucination-risk-aligned retrieval trigger timing as an explicit sentence-level RL optimization target. **Second,** we propose RHRD, which provides estimated sentence-level hallucination risk signals for local retrieval decisions. **Third,** we introduce **Grounded-R1**, a two-phase RL framework that combines episode-level quality-cost learning with RHRD-aligned sentence-level credit assignment to optimize retrieval trigger timing. **Fourth,** we design three complementary evaluations showing that Grounded-R1 improves grounded generation not by retrieving more, but by retrieving where one sentence of delay matters.

2 Related Work

Hallucination mitigation spans training-time alignment, post-hoc verification and correction, and inference-time evidence augmentation (Pan et al., 2024; Kamoi et al., 2024). Training-time alignment methods such as reinforcement learning from human feedback (RLHF) improve instruction following and preference alignment, but lack generation-time access to external evidence (Ouyang et al., 2022). Post-hoc methods detect, filter, or revise outputs after generation, making them poorly suited to preventing local errors from propagating through the trajectory (Manakul et al., 2023; Zhang et al., 2023; Pan et al., 2024; Kamoi et al., 2024). RAG offers a more direct evidence-based path for grounded generation by injecting external knowledge (Lewis et al., 2020; Klesel & Wittmann, 2025). Static RAG improves evidence access, but cannot respond to new evidence needs that emerge during generation, motivating dynamic RAG, which shifts the problem from retrieval availability to retrieval trigger timing (Lewis et al., 2020; Su et al., 2025; 2024; Trivedi et al., 2023; Jiang et al., 2023). Training-free dynamic RAG methods such as FLARE, DRAGIN, SeaKR, and ETC trigger retrieval from information needs, self-awareness, or uncertainty trends (Jiang et al., 2023; Su et al., 2024; Yao et al., 2025; Li et al., 2026). These methods show that timing matters, but their triggers still rely

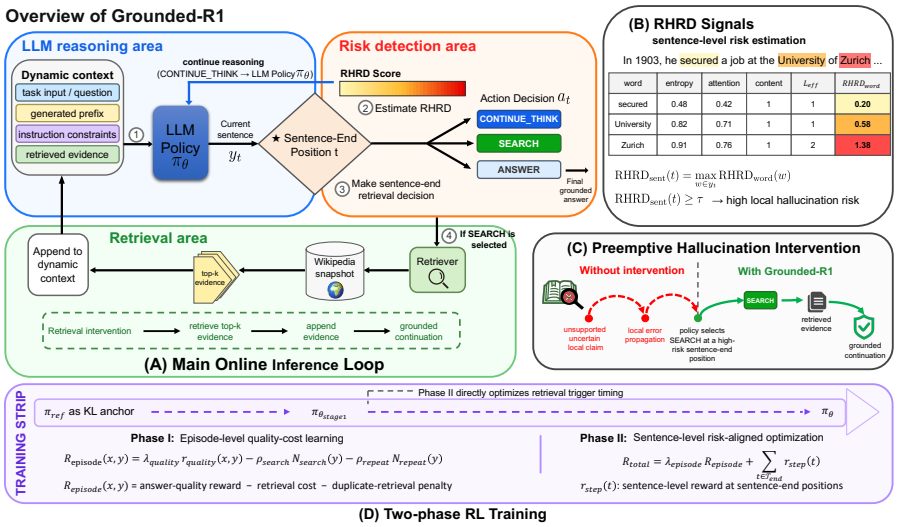


Figure 1: **Overview of Grounded-R1.** (A) The learned policy makes sentence-end retrieval decisions from the dynamic context. (B) RHRD estimates sentence-level hallucination risk from generation-time signals. (C) Retrieval intervenes before risky local claims propagate. (D) Two-phase RL first learns high-quality, low-cost generation, then aligns retrieval trigger timing with hallucination risk.

on heuristic signals, thresholds, or fixed procedures. Such signals are easy to deploy but difficult to recalibrate under context shift, noisy evidence, or cases where the model can safely continue without retrieval. IRCoT and Search-o1 further interleave retrieval with reasoning for complex question answering (Trivedi et al., 2023; Li et al., 2025). Their strength lies in organizing multi-step search and reasoning, but their retrieval boundaries remain governed by prompted workflows rather than an explicit objective for deciding when evidence should intervene. Other work makes retrieval learnable. Self-RAG uses reflection tokens to learn when to retrieve, generate, and critique retrieved evidence (Asai et al., 2024). In parallel, RL-based RAG optimizes retrieval behavior through interaction. Search-R1 and R1-Searcher optimize interleaved reasoning-retrieval trajectories with outcome-based rewards (Jin et al., 2025; Song et al., 2025). RAG-R1 extends this line with multi-query parallelism, and LeTS combines step-level and outcome-based rewards for think-and-search trajectories (Tan et al., 2026; Zhang et al., 2025). These methods improve retrieval use, but their learning signals mainly reward self-reflection, trajectory quality, or final task success. They give limited credit to when retrieval should be triggered or suppressed, leaving retrieval timing weakly aligned with local hallucination risk. Overall, optimizing RAG retrieval behavior explicitly for preemptive hallucination intervention during generation remains underexplored.

3 GROUNDED-R1

Grounded-R1 treats dynamic retrieval as a sentence-end decision problem for preemptive hallucination intervention. Given an input question x, the policy generates an interleaved think–search–answer trajectory, where y_t denotes the t-th reasoning sentence and its end defines decision point t. At each such point, the current dynamic context comprises the input, generation history, and previously retrieved evidence, and retrieval can add new evidence before generation proceeds. The two-phase RL framework first learns high-quality, low-cost generation in Phase I. During Phase II training, it uses the generation-time sentence-level hallucination risk signal from RHRD as a reward signal to align the policy's sentence-end retrieval decisions with local hallucination risk. Figure 1 summarizes the overall framework.

3.1 RHRD: REAL-TIME HALLUCINATION RISK DETECTION

Word-level hallucination risk score. We construct a word-level risk score from four signals. Word-internal averages of entropy $\overline{H}(w)$ and attention $\overline{A}(w)$ provide internal uncertainty and attention-

based salience signals, respectively. The content-word indicator $s(w) \in \{0, 1\}$ suppresses function-word noise. The effective subword token count $L_{\text{eff}}(w)$ serves as an auxiliary heuristic for lexical rarity and morphological complexity. We define:

$$\text{RHRD}_{\text{word}}(w) = \overline{H}(w) \cdot \overline{A}(w) \cdot s(w) \cdot L_{\text{eff}}(w). \tag{1}$$

Here, $\overline{H}(w)$ averages subword entropy, $\overline{A}(w)$ averages token-level maximum-attention scores, and $L_{\text{eff}}(w)$ denotes the effective subword token count. Full implementation details are given in Appendix A. The design of Eq. 1 is inspired by RIND in DRAGIN (Su et al., 2024), but repurposes it from information need detection to hallucination risk estimation.[1]

Sentence-level hallucination risk score. After computing $\text{RHRD}_{\text{word}}(w)$ for all words in sentence y_t, we take the maximum as the sentence-level hallucination risk score:

$$\text{RHRD}_{\text{sent}}(t) = \max_{w \in y_t} \text{RHRD}_{\text{word}}(w). \tag{2}$$

As illustrated in Figure 1(B), RHRD approximates hallucination risk from word-level signals and aggregates them into a sentence-level risk score. We use max aggregation to preserve the strongest local risk signal within each sentence. We emphasize that $\text{RHRD}_{\text{sent}}(t)$ is an operational proxy for local hallucination risk used for sentence-level reward supervision, not a post-hoc hallucination detector. Internal uncertainty is mainly reflected by word-level average entropy, while the evidence gap can be indirectly represented by the joint rise of entropy and attention on relevant words under the current dynamic context. Our empirical analysis shows that elevated RHRD scores are associated with both factuality hallucination and faithfulness hallucination. We therefore use a unified operational proxy rather than separate estimators for the two hallucination types.

3.2 Two-phase RL for Preemptive Hallucination Intervention

As shown in Figure 1(C), Grounded-R1 treats retrieval as an intervention action for hallucination risk. We model retrieval trigger timing as a learnable temporal decision problem and optimize it with a two-phase RL framework in Figure 1(D). Phase I learns a strong base policy for high-quality, low-cost generation. Phase II uses the sentence-level risk signal from RHRD to directly optimize retrieval trigger timing. We choose sentence-level intervention because it avoids prematurely interrupting local semantics, as token-level intervention would, while still reacting earlier than episode-level intervention. It thus provides a better trade-off among timeliness, semantic completeness, and implementation practicality.

3.2.1 Phase I: Episode-level Reward Learning for High-quality, Low-cost Generation

RAG Rollout and Action Space. Both phases are built on the same dynamic RAG scaffold as Search-R1 (Jin et al., 2025). Given an input question x, the policy generates an output sequence y over an interleaved think-search-answer trajectory. It reasons in the `<think>` segment, triggers retrieval in the `<search>` segment, receives returned evidence in the `<information>` segment, and finally outputs the answer in the `<answer>` segment. Rollout details are given in Appendix B.1.

Episode-level Reward Design. We define an episode-level quality-cost reward consisting of an answer-quality reward and retrieval-cost penalties. We use a continuous semantic score $s_{\text{semantic}}(x, y) \in [0, 1]$ and fold format compliance into answer quality:

$$r_{\text{quality}}(x, y) = (1 - \lambda_{\text{format}}) \cdot s_{\text{semantic}}(x, y) + \lambda_{\text{format}} \cdot \mathbf{1}_{\text{format}}(y). \tag{3}$$

Let $N_{\text{search}}(y)$ denote the number of effective retrieval calls, and let $N_{\text{repeat}}(y)$ denote the duplicate-retrieval count for repeated evidence. The Phase I episode-level quality-cost reward is:

$$R_{\text{episode}}(x, y) = \lambda_{\text{quality}} \cdot r_{\text{quality}}(x, y) - \rho_{\text{search}} \cdot N_{\text{search}}(y) - \rho_{\text{repeat}} \cdot N_{\text{repeat}}(y). \tag{4}$$

Here, λ_{quality}, ρ_{search}, and ρ_{repeat} control the relative weights of answer quality, over-retrieval penalty, and duplicate-retrieval penalty. Implementation details and hyperparameter settings are given in Appendices B.3 and F.3.

[1] RIND combines uncertainty, attention-based importance, and semantic filtering. RHRD builds on this idea by averaging subword entropy and attention within words, adding an effective subword token count, and using the resulting sentence-level score as an operational proxy for local hallucination risk in Phase II.

Training Objective and Policy Optimization. The training objective of Phase I is:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}}\Big[\, \mathbb{E}_{y\sim\pi_\theta(\cdot|x;\mathcal{R})}[R_{\text{episode}}(x,y)] \;-\; \beta\mathbb{D}_{\text{KL}}\big(\pi_\theta(\cdot \mid x;\mathcal{R}) \,||\, \pi_{\text{ref}}(\cdot \mid x;\mathcal{R})\big)\Big]. \tag{5}$$

The base policy obtained from this stage is denoted by $\pi_{\theta_{\text{stage1}}}$.

3.2.2 Phase II: Step-level Reward Optimization for Retrieval Trigger Timing

The Phase I reward provides delayed, indirect supervision for retrieval timing. A rollout usually contains multiple retrieval decisions, yet one terminal reward cannot identify which decision dominated the final outcome. It also cannot provide timely feedback when local hallucination risk emerges during generation. Therefore, Phase II introduces the sentence-level risk signal from RHRD and applies a step-level reward only at sentence-end decision points.

RAG Rollout and Action Space. As shown in Figure 1(A), Phase II models retrieval trigger timing as a ternary decision process at sentence-end positions. The policy generates the next reasoning sentence y_t and selects $a_t \in \{\texttt{CONTINUE_THINK}, \texttt{ANSWER}, \texttt{SEARCH}\}$ from the evolving dynamic context. The trainer then computes $\text{RHRD}_{\text{sent}}(t)$ for y_t and assigns the corresponding sentence-level reward for a_t to the sentence-ending token. Here, `CONTINUE_THINK` generates the next reasoning sentence in the `<think>` segment. `SEARCH` ends the current thinking segment and triggers retrieval through the `<search>` segment. `ANSWER` ends reasoning and outputs the final answer through the `<answer>` segment. When retrieval is triggered, newly retrieved evidence is written back to the dynamic context. Otherwise, the policy continues or answers under the current evidence state.

Step-level Reward Design. We define the following step-level reward:

$$r_{\text{step}}(t) = \begin{cases} +\alpha, & \text{if } \text{RHRD}_{\text{sent}}(t) \geq \tau \wedge a_t = \texttt{SEARCH} \\ -\alpha, & \text{if } \text{RHRD}_{\text{sent}}(t) \geq \tau \wedge a_t \in \{\texttt{CONTINUE_THINK}, \texttt{ANSWER}\} \\ +\gamma, & \text{if } \text{RHRD}_{\text{sent}}(t) < \tau \wedge a_t \in \{\texttt{CONTINUE_THINK}, \texttt{ANSWER}\} \\ -\gamma, & \text{if } \text{RHRD}_{\text{sent}}(t) < \tau \wedge a_t = \texttt{SEARCH} \end{cases}. \tag{6}$$

Here, a_t is the action taken at sentence-end time step t, $\alpha, \gamma \in \mathbb{R}^+$ are the reward magnitudes under high-risk and low-risk states, and τ is the risk threshold. When $\text{RHRD}_{\text{sent}}(t) \geq \tau$, the current sentence is treated as exhibiting potentially high local hallucination risk. Otherwise, it is treated as relatively low risk. This reward directly aligns risk state with action preference. It encourages `SEARCH` under high risk and encourages `CONTINUE_THINK` or `ANSWER` under low risk. We further set $\alpha > \gamma$ because missing a retrieval under high risk is more costly than triggering an unnecessary retrieval under low risk. On the held-out validation set, RHRD achieves an AUROC of 0.7607 and an AUPRC of 0.7625; $\tau = 0.56$ achieves a recall of 0.9047 (Appendix B.8). Importantly, RHRD serves as Phase II reward supervision rather than an inference-time hard-threshold retrieval rule. The threshold τ only converts the continuous risk score into a high-risk or low-risk state for sentence-level credit assignment; it does not directly prescribe the deployed action. Instead, this supervision shapes the policy to learn risk-aware retrieval behavior. At inference, the learned policy chooses among $\{\texttt{CONTINUE_THINK}, \texttt{ANSWER}, \texttt{SEARCH}\}$ based on the evolving dynamic context; neither the binary risk state nor the continuous RHRD score is a policy input. We test this distinction in Section 4.2 using a same-checkpoint inference-time hard-threshold control.

Mixed Reward and Training Objective. We combine the step-level reward with the episode-level reward and define the mixed reward as:

$$R_{\text{total}}(x,y) = \lambda_{\text{episode}} \cdot R_{\text{episode}}(x,y) + \sum_{t\in\mathcal{T}_{\text{end}}} r_{\text{step}}(t). \tag{7}$$

Here, $r_{\text{step}}(t)$ is assigned at each sentence-end decision point $t \in \mathcal{T}_{\text{end}}$, while $R_{\text{episode}}(x,y)$ is assigned once at the end of the episode. Based on this mixed reward, the Phase II optimization objective is:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}}\Big[\, \mathbb{E}_{y\sim\pi_\theta(\cdot|x;\mathcal{R})}[R_{\text{total}}(x,y)] \;-\; \beta\mathbb{D}_{\text{KL}}\big(\pi_\theta(\cdot \mid x;\mathcal{R}) \,||\, \pi_{\theta_{\text{stage1}}}(\cdot \mid x;\mathcal{R})\big)\Big]. \tag{8}$$

Phase II is initialized from the Phase I policy $\pi_{\theta_{\text{stage1}}}$. This initialization, the KL constraint anchored to the same policy, and the mixed reward jointly help preserve the base generation capability learned in Phase I while optimizing retrieval trigger timing.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks and Metrics. We evaluate Grounded-R1 on eight knowledge-intensive benchmarks, including HotpotQA, 2WikiMultiHopQA, MuSiQue, Bamboogle, PopQA, TriviaQA, ASQA, and HaluEval (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022; Press et al., 2023; Mallen et al., 2023; Joshi et al., 2017; Stelmakh et al., 2022; Li et al., 2023). They cover multi-hop reasoning, open-domain factual QA, long-form generation, and hallucination-related task settings. Experiments draw from this shared benchmark pool according to their evaluation goals. We report three evaluations. Episode-level evaluation measures final answer quality and retrieval cost, reporting EM, F1, LLM-as-a-judge score, non-hallucination rate (NHR), and the average number of retrieval calls. Sentence-level hallucination analysis reports overall NHR under open-ended generation. Causal retrieval trigger timing evaluation comprises two experiments. The matched-prefix intervention measures the downstream benefit of timely evidence using the first downstream sentence hallucination rate, suffix NHR, and final EM. The equal-budget retrieval benefit evaluation tests whether selectors prioritize positions where timely evidence yields larger hallucination reductions, using the Immediate − Delayed difference in suffix NHR. Dataset statistics and metric definitions appear in Appendices C and D.

Baselines. We group baselines into three categories. Direct generation and static RAG baselines include Direct Inference and static RAG (Lewis et al., 2020), corresponding to no-retrieval generation and one-shot retrieve-then-generate, respectively. Training-free dynamic RAG baselines include FLARE, DRAGIN, SeaKR, ETC, IRCoT, and Search-o1 (Jiang et al., 2023; Su et al., 2024; Yao et al., 2025; Li et al., 2026; Trivedi et al., 2023; Li et al., 2025). Learning-based dynamic RAG baselines include Self-RAG, Search-R1,[2] and R1-Searcher (Asai et al., 2024; Jin et al., 2025; Song et al., 2025). Implementation details and evaluation scopes are in Appendix E.

Training and Implementation Details. We follow the Search-R1 retrieval and training setup (Jin et al., 2025). We use the 2018 Wikipedia dump as the external knowledge base and perform top-3 dense retrieval with E5 and FAISS-GPU (Karpukhin et al., 2020; Wang et al., 2022; Johnson et al., 2021). For fair comparison, retrieval-based methods share the same knowledge base, retriever, and top-3 setting whenever possible. RL training uses about 170k examples from Natural Questions (NQ) and HotpotQA (Kwiatkowski et al., 2019; Yang et al., 2018) and optimizes Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct with PPO and GRPO (Schulman et al., 2017; Shao et al., 2024). Full training details are reported in Appendix F.

4.2 MAIN RESULTS, REWARD ABLATIONS, AND RHRD HARD-THRESHOLD CONTROL

Table 1 shows that Grounded-R1 (full) achieves the best results on all 16 metrics, outperforming every external baseline. The gains in LLM-as-a-judge score indicate improvements beyond surface answer matching. On HaluEval, joint EM and NHR gains indicate higher correctness with fewer hallucinations, while on ASQA, higher EM and F1 show benefits for long-form generation.

The three trained variants show progressive gains. Replacing terminal EM with a continuous semantic reward improves final-answer quality beyond lexical matching in Phase I, while Phase II further translates step-level retrieval optimization into stronger episode-level performance.

The RHRD hard-threshold control tests whether the validation-fixed threshold alone is sufficient as an inference-time retrieval policy. It reuses the Grounded-R1 (episode-semantic) checkpoint that initializes Grounded-R1 (full), performs no Phase II policy optimization, and forces `SEARCH` if and only if $\mathrm{RHRD}_{\mathrm{sent}}(t) \geq \tau$ and retrieval budget remains, with $\tau = 0.56$ fixed on the validation set. Its underperformance relative to Grounded-R1 (full) across all 16 metrics shows that deterministic threshold gating does not reproduce the gains of Phase II policy optimization. This supports using RHRD as a sentence-level learning signal rather than a deterministic retrieval rule, allowing Phase II to condition retrieval on the evolving dynamic context and accumulated evidence while remaining aligned with the episode-level quality-cost objective. Complete ablations and hard-threshold-control details are provided in Appendix L.1 and Appendix K.

[2]We use the publicly released Search-R1 v0.3 setting.

Table 1: Main results, reward-design ablations, and RHRD hard-threshold control across eight benchmarks. All Grounded-R1 rows use the unified 7B GRPO setting. Best and second-best results are bold and underlined, respectively. LLM and NHR denote LLM-as-a-judge score and episode-level non-hallucination rate, both evaluated by Qwen2.5-72B-Instruct.

Methods	HotpotQA		2Wiki		MuSiQue		Bamboogle		PopQA		TriviaQA		ASQA		HaluEval	
	EM	LLM	EM	LLM	EM	LLM	EM	LLM	EM	LLM	EM	LLM	EM	F1	EM	NHR
Direct Inference	0.120	0.257	0.136	0.261	0.040	0.122	0.280	0.432	0.147	0.241	0.400	0.614	0.220	0.325	0.270	0.490
Static RAG	0.164	0.329	0.032	0.166	0.042	0.140	0.168	0.327	0.275	0.521	0.392	0.766	0.230	0.391	0.245	0.277
DRAGIN	0.138	0.270	0.102	0.266	0.066	0.141	0.208	0.344	0.160	0.260	0.542	0.623	0.330	0.426	0.188	0.329
FLARE	0.130	0.245	0.116	0.243	0.038	0.081	0.240	0.356	0.112	0.192	0.396	0.556	0.220	0.301	0.168	0.314
R1-Searcher	0.078	0.154	0.078	0.119	0.024	0.047	0.072	0.123	0.137	0.173	0.608	0.713	0.100	0.150	0.106	0.189
ETC	0.164	0.270	0.162	0.263	0.044	0.111	0.240	0.342	0.166	0.234	0.597	0.721	0.350	0.448	0.198	0.073
SeaKR	0.000	0.282	0.000	0.099	0.000	0.110	0.000	0.296	0.271	0.478	0.574	0.691	0.360	0.540	0.004	0.395
IRCoT	0.216	0.411	0.078	0.239	0.068	0.198	0.160	0.339	0.358	0.496	0.532	0.749	0.430	0.560	0.319	0.526
Search-R1	0.272	0.398	0.282	0.339	0.096	0.150	0.208	0.321	0.393	0.468	0.596	0.726	0.580	0.663	0.357	0.497
Grounded-R1 (episode-EM)	0.163	0.252	0.220	0.253	0.032	0.084	0.120	0.201	0.175	0.225	0.428	0.521	0.280	0.372	0.206	0.306
Grounded-R1 (episode-semantic)	0.180	0.273	0.228	0.274	0.050	0.107	0.128	0.205	0.202	0.268	0.474	0.599	0.370	0.463	0.221	0.336
Grounded-R1 (hard-threshold control)	0.124	0.249	0.108	0.191	0.032	0.070	0.200	0.306	0.132	0.201	0.386	0.549	0.230	0.346	0.173	0.314
Grounded-R1 (full)	**0.346**	**0.488**	**0.292**	**0.382**	**0.160**	**0.238**	**0.368**	**0.516**	**0.417**	**0.525**	**0.620**	**0.774**	**0.660**	**0.754**	**0.423**	**0.591**

4.3 Retrieval Cost and Quality-Cost Trade-off

Together, Tables 1 and 2 show that Grounded-R1 improves the quality-cost trade-off without materially increasing retrieval volume. Phase II raises the macro-average number of retrieval calls only from 2.00 to 2.04, while Grounded-R1 (full) has a lower macro average than all external baselines in Table 2. In contrast, the same-checkpoint RHRD hard-threshold control averages 4.12 calls, more than twice as many, yet performs worse on all episode-level task metrics. These results show that Phase I learns retrieval restraint, whereas Phase II improves episode-level outcomes with nearly unchanged retrieval volume.

Table 2: Average number of retrieval calls across seven benchmarks. Macro averages are unweighted over datasets.

Methods	Hotpot QA	2Wiki	MuSi Que	Bam boogle	Pop QA	Halu Eval	Trivia QA	Macro Avg.
FLARE	2.91	3.19	2.93	2.84	2.63	2.83	2.52	2.84
DRAGIN	3.75	3.39	3.92	2.97	3.62	3.70	3.70	3.58
Search-R1	2.13	2.50	2.44	2.19	**1.84**	2.12	2.00	2.17
R1-Searcher	3.64	3.72	3.89	3.70	3.26	3.62	3.00	3.55
Grounded-R1 (hard-threshold control)	4.19	4.01	4.31	3.86	4.13	4.20	4.13	4.12
Grounded-R1 (episode-semantic)	**2.00**	**2.04**	**2.00**	**2.00**	2.00	**2.01**	**1.95**	**2.00**
Grounded-R1 (full)	2.08	2.07	2.06	2.02	2.02	2.04	2.02	2.04

4.4 Sentence-level Hallucination Analysis in Open-ended Generation

To evaluate local grounded generation under standard model rollouts, we split model outputs into sentences on five representative benchmarks and perform automatic sentence-level hallucination annotation. The main paper reports overall NHR; factuality and faithfulness breakdowns appear in Appendix H. For compactness, Grounded-R1 (episode) denotes Grounded-R1 (episode-semantic).

Table 3 shows that Grounded-R1 (full) achieves the highest overall NHR on all five benchmarks, improving the macro average by 12.7 points over the strongest external baseline, Search-o1. This suggests that Grounded-R1 (full) reduces local hallucinations in open-ended generation, with larger gains on ASQA and PopQA indicating stronger benefits in long-form answer generation and long-tail knowledge settings.

Table 3: Sentence-level overall NHR (%) on five open-ended benchmarks. Higher is better. A sentence is non-hallucinated only if it contains neither factuality nor faithfulness hallucination; macro averages are unweighted.

Methods	ASQA	Bam boogle	Halu Eval	Pop QA	Trivia QA	Macro Avg.
Search-R1	48.40	40.94	29.46	37.38	46.32	40.50
R1-Searcher	41.34	48.41	31.50	33.99	48.67	40.78
Self-RAG	60.10	44.65	41.12	60.83	64.95	54.33
Search-o1	58.22	62.68	47.00	53.68	61.33	56.58
IRCoT	54.17	65.44	49.34	46.51	61.02	55.30
ETC	54.84	59.92	42.48	46.05	59.43	52.54
Grounded-R1 (episode)	68.81	57.59	46.27	58.80	69.53	60.20
Grounded-R1 (full)	**79.63**	**67.02**	**52.96**	**73.52**	**73.44**	**69.31**

Table 4: Matched-prefix intervention on retrieval trigger timing. Absolute results are macro averages across five datasets; differences are paired percentage-point estimates with 95% confidence intervals.

	Absolute performance (%)			Paired difference (pp; 95% CI)	
Outcome	**No retrieval**	**Delayed**	**Immediate**	**Immediate − No**	**Immediate − Delayed**
First downstream hallucination rate ↓	54.84	54.84	**43.74**	-11.10 [-13.90, -8.20]	-11.10 [-13.90, -8.20]
Suffix NHR ↑	52.02	52.70	**59.84**	$+7.82$ [$+5.60$, $+9.80$]	$+7.14$ [$+5.00$, $+9.10$]
Final EM ↑	48.02	50.06	**52.64**	$+4.62$ [$+2.70$, $+6.80$]	$+2.58$ [$+0.80$, $+4.40$]

We validate the automated hallucination annotations on 100 claim-bearing sentences spanning five datasets and multiple model families, with model identities hidden from annotators. Human–human agreement is 96.0%/95.0%, and tool–human agreement is 88.0%/91.0%, for factuality and faithfulness, respectively; all 33 human-positive factuality cases are detected. This sanity check supports the consistency of the automated annotations for relative model comparisons. Detailed agreement statistics and annotation rules are reported in Appendices H.7 and H.1.

4.5 Causal Evaluation of Retrieval Trigger Timing

Earlier Evidence Reduces Downstream Hallucinations. To isolate retrieval trigger timing, we sample 5,000 sentence-end `SEARCH` positions from standard Grounded-R1 (full) trajectories and construct three paired continuations from the same prefix at each position. Immediate retrieval inserts the original query and top-3 evidence before the first downstream sentence. Delayed and No retrieval first share the same sentence generated without evidence; Delayed then inserts the same query and evidence, while No retrieval skips the retrieval. Subsequent retrieval is disabled in all branches. Appendix I provides the full protocol and per-dataset results.

Table 4 shows that Immediate retrieval outperforms No retrieval, confirming the aggregate retrieval value of the selected sentence-end positions. More importantly, compared with Delayed retrieval, delivering the same evidence one sentence earlier reduces the first downstream sentence hallucination rate by 11.1 percentage points (pp), raises suffix NHR by 7.1 pp, and improves final EM by 2.6 pp; all 95% confidence intervals exclude zero. Delayed and No retrieval share the first downstream sentence by construction and therefore have the same hallucination rate, yet Delayed retrieval yields higher suffix NHR and final EM point estimates. Thus, late evidence may improve subsequent generation but cannot prevent an error already emitted. These results provide direct causal evidence that timely retrieval prevents local error propagation and acts as a preemptive hallucination intervention.

Grounded-R1 Prioritizes Positions with Larger Downstream Hallucination Reductions.

We construct a source-balanced shared pool of 2,000 sentence-end states from unmodified trajectories of Grounded-R1 (full), Search-R1, and Grounded-R1 (episode). Each frozen selector ranks all states by its deterministic `SEARCH` score and selects the top 200 under identical prespecified dataset budgets, so the selectors use the same retrieval budget but choose different positions. At each selected position, we reuse the Immediate–Delayed intervention with the same prefix, query, and top-3 evidence, and measure suffix NHR. We refer to this Immediate–Delayed difference as the *retrieval timing benefit*: a larger value means that providing the same evidence one sentence earlier produces a larger reduction in downstream hallucination. Appendix J provides the full protocol.

Table 5: Equal-budget comparison of retrieval timing benefit across selectors. Panel A reports Immediate − Delayed suffix NHR over all selected positions; Panels B and C report the same contrast over pairwise-exclusive positions; values are percentage-point differences with 95% confidence intervals.

Analysis set / Selector	**Immediate − Delayed suffix NHR (pp) ↑**
Panel A: All selected positions	
Grounded-R1 (full)	$\mathbf{+8.3}$ [$\mathbf{+2.3}$, $\mathbf{+14.5}$]
Search-R1	$+7.0$ [-0.2, $+14.3$]
Grounded-R1 (episode)	$+3.4$ [-2.8, $+9.2$]
Panel B: Pairwise-exclusive, Grounded-R1 (full) vs. Search-R1	
Grounded-R1 (full) only	$\mathbf{+5.3}$ [$\mathbf{-2.3}$, $\mathbf{+13.1}$]
Search-R1 only	$+4.4$ [-4.8, $+12.9$]
Panel C: Pairwise-exclusive, Grounded-R1 (full) vs. Grounded-R1 (episode)	
Grounded-R1 (full) only	$\mathbf{+8.4}$ [$\mathbf{+0.8}$, $\mathbf{+16.2}$]
Grounded-R1 (episode) only	$+3.8$ [-4.7, $+11.6$]

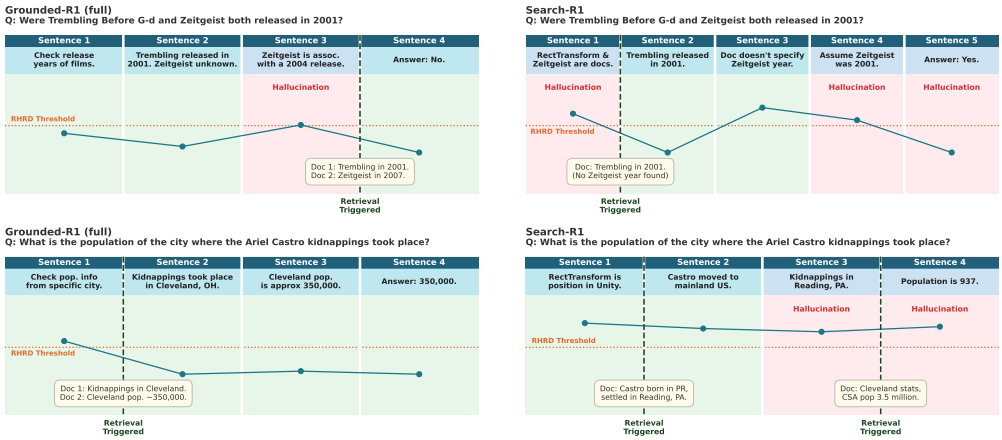


Figure 2: Case studies of retrieval triggering and hallucination intervention. Green and red blocks denote human-verified non-hallucinated and hallucinated sentences; blue curves show sentence-level RHRD scores (overlaid post hoc and not used at inference); orange horizontal and green vertical dashed lines denote the threshold $\tau = 0.56$ and retrieval positions; yellow boxes show evidence.

Table 5 shows that, in Panel A, Grounded-R1 (full) has the largest point estimate and the only 95% confidence interval entirely above zero. Its point estimate also exceeds that of Search-R1 on pairwise-exclusive positions (Panel B). More diagnostically, excluding positions shared with Grounded-R1 (episode) leaves the full-only benefit significant and virtually unchanged (+8.4 vs. +8.3 pp), while the episode-only effect is much smaller and its confidence interval includes zero (Panel C). Together with the nearly unchanged retrieval calls, this pattern indicates that Phase II shifts a similar retrieval budget toward positions where earlier evidence more strongly reduces downstream hallucination. Thus, the matched-prefix intervention establishes the causal benefit of earlier evidence, while the equal-budget comparison shows that Grounded-R1 (full) tends to trigger retrieval where this benefit is larger.

5 Case Studies

Figure 2 aligns sentence-level generation trajectories with interpretable signals, illustrating that effective hallucination intervention depends on retrieval trigger timing and evidence quality rather than retrieval frequency. Across the two cases, Grounded-R1 retrieves near risk-critical points before local errors propagate, while Search-R1 shows decisions misaligned with hallucination risk, with missed retrieval under high risk, delayed retrieval after a wrong trajectory forms, and retrieval that leaves the key evidence gap unresolved. In the movie-year case, Grounded-R1 retrieves after generating an unsupported 2004 claim; the post-hoc RHRD trace marks this position as high risk. The retrieved evidence supplies the key facts, "Trembling in 2001" and "Zeitgeist in 2007," preventing local error propagation. Search-R1 also retrieves, but fails to resolve the Zeitgeist year, continues with insufficient evidence, and ends with an incorrect answer. In the city-population case, Grounded-R1 quickly narrows the trajectory to the minimal necessary evidence chain of the specific city and its population, staying grounded with one timely retrieval. Search-R1 performs two misaligned retrievals, drifts to the wrong location and mismatched population scope, and ends with an incorrect answer.

6 Conclusion

In this paper, we presented **Grounded-R1**, a dynamic RAG framework that reframes preemptive hallucination intervention as a sentence-end retrieval trigger timing problem. Through two-phase RL, Grounded-R1 aligns retrieval timing with generation-time hallucination risk, improving final-answer performance and sentence-level grounded generation. Controlled timing interventions show that delivering the same evidence one sentence earlier cuts next-sentence hallucination rate by 11.1 points; under equal retrieval budgets, Grounded-R1 prioritizes positions where earlier evidence yields an observed 8.3-point reduction in downstream hallucination. With nearly unchanged retrieval calls, Grounded-R1 improves grounded generation by retrieving where one sentence of delay matters.

AI USE STATEMENT

In this work, ChatGPT was used to generate the 5,000 low-risk common-sense questions that constitute the low-risk subset of the validation set used to calibrate the RHRD threshold τ. Qwen2.5-72B-Instruct was used as an LLM-as-a-judge model to compute the semantic score in the Phase I answer-quality reward, evaluate final answers, and perform automatic sentence-level hallucination annotation. Generative AI tools were also used to assist with literature search and summarization and to draft or edit parts of the manuscript for clarity, terminology consistency, structure, and LaTeX formatting. Generative AI tools were not used to formulate mathematical claims or proofs. The authors designed and supervised all AI-assisted research workflows and reviewed their use and outputs using the validation and quality-control procedures described in the paper. All experiments, statistical analyses, and reported numerical results were executed or independently verified by the authors. AI-assisted manuscript content was checked against the cited sources, implementation, and experimental records and corrected where necessary. The authors take responsibility for every research decision and for the final content of this work, including all text, claims, citations, code, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Grounded-R1 is designed to improve grounded generation, but it does not eliminate hallucination risk. Incomplete, biased, outdated, or maliciously manipulated retrieval corpora can cause incorrect answers to appear supported by evidence, thereby increasing unwarranted user trust. Uneven corpus coverage may also result in unequal reliability across topics, languages, or populations. When dynamic RAG is connected to private or untrusted knowledge bases, retrieval queries, retrieved passages, and intermediate reasoning traces may expose confidential information, while adversarial retrieved content may manipulate generation. Automated factuality and faithfulness annotations can inherit errors or biases from judge models and the evidence used for verification. We therefore use these annotations only for threshold calibration and research evaluation and assess the annotation protocol through the human sanity check reported in Appendix H.7. Grounded-R1 should not be used as an autonomous decision-maker in medical, legal, financial, security-critical, or other high-stakes settings. Any such deployment would require domain-specific validation, access controls, data minimization and filtering, corpus provenance and update governance, retrieval-quality and security monitoring, audit logs, and meaningful human oversight. Our experiments use established public benchmarks, publicly available base and judge models, and a fixed Wikipedia retrieval corpus. We do not introduce a dataset involving human subjects or collect new personal or sensitive data. The human sanity check was conducted by members of the research team solely to evaluate model-output annotations; no external participants were recruited, and no information about the raters was collected or analyzed. We cite the external datasets, models, and corpus resources used in our experiments, whose use and redistribution remain subject to their applicable licenses and terms. The released code and Grounded-R1 checkpoints are intended for research on dynamic RAG and retrieval trigger timing; downstream use should account for the risks described above and the limitations in Appendix M.

REPRODUCIBILITY STATEMENT

Section 3 specifies RHRD, the two-phase RL framework, and the training objectives, while Section 4 defines the benchmarks, baselines, metrics, and evaluation scope. Appendices A and B document the implementation of RHRD, the dynamic RAG rollout, the reward functions, the sentence-end decision construction, and the sentence-end risk-threshold calibration. Appendices C, D, and E describe the datasets and preprocessing, evaluation definitions, judge prompts, macro-averaging procedures, and baseline implementations. Appendix F reports the training data, unified retrieval environment, PPO and GRPO hyperparameters, reward weights, training schedules, compute resources, checkpoint-selection criteria, and leakage-control procedures. Appendices H, I, and J provide the complete protocols for sentence-level hallucination annotation, the matched-prefix intervention, and the equal-budget evaluation, including their integrity checks and statistical procedures. The anonymous project page linked in the abstract provides the source code and Grounded-R1 model checkpoints.

REFERENCES

A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

Y. Bang, Z. Ji, A. Schelten, A. Hartshorn, T. Fowler, C. Zhang, N. Cancedda, and P. Fung. HalluLens: LLM hallucination benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24128–24156, 2025.

J. Ferrando, O. Balcells Obeso, S. Rajamanoharan, and N. Nanda. Do I know this entity? Knowledge awareness and hallucinations in language models. In *International Conference on Learning Representations*, 2025.

X. Ho, A.-K. Duong Nguyen, S. Sugawara, and A. Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):42:1–42:55, 2025.

Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, A. Madotto, and P. Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):248:1–248:38, 2023.

Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, 2023.

B. Jin, H. Zeng, Z. Yue, J. Yoon, S. O. Arik, D. Wang, H. Zamani, and J. Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Proceedings of the Second Conference on Language Modeling (COLM)*, 2025.

J. Johnson, M. Douze, and H. Jégou. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2021.

M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. 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)*, pp. 1601–1611, 2017.

R. Kamoi, Y. Zhang, N. Zhang, J. Han, and R. Zhang. When can LLMs actually correct their own mistakes? A critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024.

G. Kaplan, M. Oren, Y. Reif, and R. Schwartz. From tokens to words: On the inner lexicon of LLMs. In *International Conference on Learning Representations*, 2025.

V. Karpukhin, B. Oguz, S. Min, P. S. H. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

M. Klesel and H. F. Wittmann. Retrieval-augmented generation (RAG). *Business & Information Systems Engineering*, 67(4):551–561, 2025.

T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M. Chang, A. Dai, J. Uszkoreit, Q. Le, and S. Petrov. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

B. Li, T. Tian, Z. Xu, H. Cheng, S. Zhang, and W. Ye. Modeling uncertainty trends for timely retrieval in dynamic RAG. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(37): 31527–31535, 2026.

J. Li, X. Cheng, X. Zhao, J.-Y. Nie, and J.-R. Wen. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6449–6464, 2023.

X. Li, G. Dong, J. Jin, Y. Zhang, Y. Zhou, Y. Zhu, P. Zhang, and Z. Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438, 2025.

S. Lin, J. Hilton, and O. Evans. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, 2022.

A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi. 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)*, pp. 9802–9822, 2023.

P. Manakul, A. Liusie, and M. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9004–9017, 2023.

N. McKenna, T. Li, L. Cheng, M. Hosseini, M. Johnson, and M. Steedman. Sources of hallucination by large language models on inference tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2758–2774, 2023.

S. Min, K. Krishna, X. Lyu, M. Lewis, W.-t. Yih, P. W. Koh, M. Iyyer, L. Zettlemoyer, and H. Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100, 2023.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

L. Pan, M. Saxon, W. Xu, D. Nathani, X. Wang, and W. Y. Wang. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *Transactions of the Association for Computational Linguistics*, 12:484–506, 2024.

O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, 2016.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, and D. Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

H. Song, J. Jiang, Y. Min, J. Chen, Z. Chen, W. X. Zhao, L. Fang, and J.-R. Wen. R1-Searcher: Incentivizing the search capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.

I. Stelmakh, Y. Luan, B. Dhingra, and M.-W. Chang. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8273–8288, 2022.

W. Su, Y. Tang, Q. Ai, Z. Wu, and Y. Liu. DRAGIN: Dynamic Retrieval Augmented Generation based on the Information Needs of Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12991–13013, 2024.

W. Su, Q. Ai, J. Zhan, Q. Dong, and Y. Liu. Dynamic and parametric retrieval-augmented generation. *arXiv preprint arXiv:2506.06704*, 2025.

Z. Tan, J. Huang, Q. Wu, H. Zhang, C. Zhuang, and J. Gu. RAG-R1: Incentivizing the search and reasoning capabilities of LLMs through multi-query parallelism. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(39):33187–33195, 2026.

H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. 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)*, pp. 10014–10037, 2023.

D. Wang, Y. Li, J. Jiang, Z. Ding, Z. Luo, G. Jiang, J. Liang, and D. Yang. Tokenization matters! Degrading large language models through challenging their tokenization. *arXiv preprint arXiv:2405.17067*, 2024.

L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Z. Yao, W. Qi, L. Pan, S. Cao, L. Hu, W. Liu, L. Hou, and J. Li. SeaKR: Self-aware knowledge retrieval for adaptive retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27022–27043, 2025.

J. Zhang, Z. Li, K. Das, B. Malin, and S. Kumar. SAC^3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15445–15458, 2023.

Q. Zhang, S. Yang, L. Gao, H. Chen, X. Hu, J. Chen, J. Wang, S. Guo, B. Zheng, H. Wang, and J. Zhao. LeTS: Learning to think-and-search via process-and-outcome reward hybridization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5109–5122, 2025.

APPENDIX

A REAL-TIME HALLUCINATION RISK DETECTION DETAILS

This section gives implementation details for Real-time Hallucination Risk Detection (RHRD). The main text defines the RHRD word-level score in Eq. 1 and the RHRD sentence-level score in Eq. 2. We expand the subword-to-word aggregation, filtering rules, effective subword token count, and the role of the RHRD score as an operational proxy for local hallucination risk.

A.1 SUBWORD-TO-WORD AGGREGATION AND WORD-INTERNAL SIGNALS

RHRD computes a hallucination risk score at the word level rather than directly at the subword level. This choice aligns the risk score with a natural semantic unit by combining signals across its constituent subwords. Let a word w consist of a sequence of subword tokens:

$$w = (u_1, u_2, \ldots, u_{|w|}), \tag{9}$$

where $|w|$ is the number of subword tokens in w. For each subword token u_j, we compute the entropy of the next-token predictive distribution at the generation step that produces u_j:

$$H(u_j) = -\sum_{v \in \mathcal{V}} p_v(u_j) \log p_v(u_j), \tag{10}$$

where $p_v(u_j)$ denotes the probability assigned to vocabulary item v when generating u_j.

A direct sum of subword entropies would make longer segmented words appear riskier even when their uncertainty is not intrinsically higher. We therefore use word-internal average entropy:

$$\overline{H}(w) = \frac{1}{|w|} \sum_{u \in w} H(u). \tag{11}$$

Here and below, $\sum_{u \in w}$ denotes aggregation over all subword tokens belonging to w. Averaging entropy removes the direct accumulation with subword count that a sum would introduce.

We also aggregate the attention signal at the word level. For attention indexing, let u_i denote the subword at position i in the generated segment being scored, with n tokens available at the current sentence-end decision point. Let $A_{j,i}^{(h)}$ denote the last-layer attention weight from query position j to token u_i under head h, and let N_{head} be the number of attention heads. Following DRAGIN's implementation, we take the maximum over query positions within each head and then average these maxima across heads:

$$a_{\max}(u_i) = \frac{1}{N_{\text{head}}} \sum_{h=1}^{N_{\text{head}}} \max_{i \le j \le n} A_{j,i}^{(h)}. \tag{12}$$

The range includes the token's own position, matching the unmasked diagonal in DRAGIN's causal-attention implementation. Only positions already available at the current sentence-end are used. We then average the token-level scores across the subwords of each word:

$$\overline{A}(w) = \frac{A(w)}{|w|} = \frac{1}{|w|} \sum_{u \in w} a_{\max}(u). \tag{13}$$

Here, $A(w)$ denotes the unnormalized sum of token-level maximum-attention scores, and $\overline{A}(w)$ is their arithmetic mean. Dividing by $|w|$ removes the direct accumulation with subword count while preserving the scale of the token-level scores. For a single-subword word, $\overline{A}(w) = a_{\max}(u)$, retaining its attention magnitude.

A.2 CONTENT-WORD FILTERING AND EFFECTIVE SUBWORD TOKEN COUNT

RHRD suppresses function-word noise through a content-word indicator. We define:

$$s(w) = \begin{cases} 1, & \text{if } w \text{ is a content word, e.g., NOUN, PROPN, VERB, ADJ, or NUM,} \\ 0, & \text{if } w \text{ is a function or stop word.} \end{cases} \tag{14}$$

This rule keeps the risk score concentrated on words that carry semantic or factual content.

The effective subword token count $L_{\text{eff}}(w)$ serves as an auxiliary heuristic for rare entities, domain terms, and morphologically complex words. Word-internal averaging of entropy and attention removes the direct accumulation of these signals with subword count. This normalization and $L_{\text{eff}}(w)$ serve different purposes: averaging controls aggregation-induced length effects, while $L_{\text{eff}}(w)$ deliberately retains effective subword count as a weak lexical-rarity prior. With byte-pair encoding and related subword tokenizers, low-frequency or unseen words are often segmented into multiple subword units (Sennrich et al., 2016). In knowledge-intensive generation, such words are frequently associated with lower entity familiarity or weaker knowledge recall (Ferrando et al., 2025; McKenna et al., 2023). Prior work also suggests that LLMs may encode whether an entity lies within their recallable knowledge range, and entities outside this range are more likely to induce hallucinated completions (Ferrando et al., 2025). At the same time, models can form stable word-level representations from multiple subword tokens, while tokenization perturbations can still degrade understanding and output accuracy (Kaplan et al., 2025; Wang et al., 2024). We therefore treat the effective subword token count as an auxiliary weak signal, not as a direct cause or observation of hallucination.

We define:

$$L_{\text{eff}}(w) = |w| - \#\text{punct}(w) - \#\text{prefix}(w) - \#\text{suffix}(w). \tag{15}$$

Here, $\#\text{punct}(w)$ removes punctuation-only subword tokens, while $\#\text{prefix}(w)$ and $\#\text{suffix}(w)$ remove frequent morphological prefixes and suffixes, such as "un-", "re-", "-ing", and "-tion". This filtering avoids treating common morphological fragments as evidence of semantic rarity. The resulting $L_{\text{eff}}(w)$ provides a more targeted estimate of the complexity and rarity of the core lexical content. The averaging denominator remains $|w|$; these exclusions apply only to $L_{\text{eff}}(w)$.

A.3 RHRD Word-level and Sentence-level Score Computation

RHRD combines the four word-level components into the RHRD word-level score:

$$\text{RHRD}_{\text{word}}(w) = \overline{H}(w) \cdot \overline{A}(w) \cdot s(w) \cdot L_{\text{eff}}(w). \tag{16}$$

This score is suppressed for function words. Among content words, it increases with higher internal uncertainty, stronger average attention, and a larger effective subword token count.

For a generated sentence y_t, we compute the RHRD word-level score for each word and take the maximum as the RHRD sentence-level score:

$$\text{RHRD}_{\text{sent}}(t) = \max_{w \in y_t} \text{RHRD}_{\text{word}}(w). \tag{17}$$

Max aggregation reflects the fact that a single high-risk content word can dominate the local risk of the sentence. In this sense, the sentence-level score serves as a conservative proxy for sentence-level hallucination risk at the current sentence-end decision point.

Computation pipeline. For each generated sentence, the implementation first aligns subword tokens to words and computes token entropy from generation logits. For last-layer attention, it takes the maximum received by each token over available query positions at or after that token within each head, then averages these maxima across heads, as in Eq. 12. The token-level attention scores and entropies are averaged within each word to obtain $\overline{A}(w)$ and $\overline{H}(w)$, respectively. The implementation then applies the content-word indicator, computes $L_{\text{eff}}(w)$, and finally applies Eq. 16 and Eq. 17. This produces one scalar sentence-level hallucination risk score for each sentence-end decision point.

A.4 RHRD as an Operational Hallucination Risk Signal

The RHRD sentence-level score is an operational proxy for local hallucination risk and a training-time signal for Phase II reward assignment. When $\text{RHRD}_{\text{sent}}(t)$ meets or exceeds the sentence-end risk threshold τ, the current sentence-end state is treated as high risk for Phase II reward assignment.

This interpretation is empirical. High word-internal average entropy $\overline{H}(w)$ indicates internal uncertainty. When the current dynamic context may be insufficient to support an emerging claim, generation can exhibit elevated entropy together with stronger average attention on relevant words. Higher $\overline{H}(w)$ and $\overline{A}(w)$ can increase the corresponding word-level risk and, through max aggregation,

the sentence-level score $\mathrm{RHRD}_{\mathrm{sent}}(t)$. Thus, the score is used as an empirical risk signal related to a possible evidence gap, not as a direct judgment of factual correctness or evidence faithfulness.

At a sentence-end time step t, Eq. 18 defines a binary risk state for sentence-level reward assignment:

$$\mathrm{RHRD}_{\mathrm{sent}}(t) \geq \tau. \tag{18}$$

This condition does not directly prescribe the deployed retrieval action. During Phase II, it determines whether `SEARCH` or a non-retrieval action receives the preferred sentence-level reward. The resulting policy remains free to trade off this proxy signal against the dynamic context and the episode-level quality-cost objective. We separately evaluate a deterministic controller that forces `SEARCH` whenever this condition holds in Appendix K.

RHRD uses a unified operational proxy signal for factuality hallucination risk and faithfulness hallucination risk. For factuality hallucination, elevated risk often reflects uncertainty about world knowledge such as entities, dates, or attributes. For faithfulness hallucination, elevated risk often reflects a possible departure from the user instruction, retrieved evidence, instruction constraints, or local context. In both cases, RHRD does not directly verify factual consistency or evidence faithfulness. It provides a real-time and unified operational proxy signal that supports sentence-level reward assignment for learning retrieval trigger timing in Phase II.

B Two-phase Reinforcement Learning Details

This section gives implementation details for the dynamic RAG rollout and the two-phase RL training framework. The goal is to use the RHRD sentence-level score as a sentence-end learning signal for retrieval trigger timing. Phase I learns a base policy for high-quality, low-cost generation. Phase II uses RHRD-derived sentence-level rewards to optimize sentence-end retrieval triggering decisions while preserving the base capability learned in Phase I.

B.1 Dynamic RAG Rollout Algorithm

Both phases use the same reasoning-retrieval interleaving scaffold as Search-R1 (Jin et al., 2025). Given an input question x, the policy samples a complete trajectory $y \sim \pi_\theta(\cdot \mid x; \mathcal{R})$ while interacting with the retrieval environment $\mathcal{R}$. The trajectory contains model-generated reasoning, retrieval queries, returned evidence, and the final answer.

The interaction is controlled by four tag types. The policy writes reasoning in `<think>` and `</think>`. It triggers retrieval by producing a query inside `<search>` and `</search>`. After the closing `</search>` tag, the environment executes retrieval and inserts the returned evidence inside `<information>` and `</information>`. The policy writes the final answer inside `<answer>` and `</answer>`, which terminates the episode and triggers episode-level scoring.

The dynamic context is updated after each retrieval call. Newly retrieved evidence is appended to the current trajectory and becomes available for subsequent decoding. This allows the policy to generate the next reasoning step under an incrementally augmented evidence state. The episode stops when the policy produces a valid `<answer>` segment, reaches the maximum generation length, or exceeds the maximum retrieval budget.

B.2 Retrieved Token Masking

Tokens inside `<information>` and `</information>` are inserted by the retrieval environment rather than generated by the policy. We therefore apply retrieved-token loss masking to these tokens during policy optimization. This prevents external evidence from contributing gradients to the policy loss and keeps optimization focused on model-generated reasoning, retrieval queries, and answer tokens. The retrieved evidence still remains in the dynamic context and can affect subsequent policy decisions through conditioning.

B.3 Phase I Episode-level Quality-Cost Reward

Answer-quality reward. Phase I uses an episode-level quality-cost reward to learn a strong base policy. The answer-quality component combines a continuous semantic score with format compliance,

Table 6: Discriminative prompt used to compute the semantic score $s_{\text{semantic}}(x, y)$.

Prompt for semantic equivalence scoring

```
# Role
You are an objective evaluator comparing a candidate response to a
golden answer.
# Instructions
1. Rate on five dimensions with integer scores from 0 to 5.
Semantic Accuracy has weight 70%. Completeness, Logical Coherence,
Clarity, and Fluency each have weight 7.5%.
2. Compute overall similarity from 0 to 100 as
overall = (0.70*SA + 0.075*(CMP+LC+CLR+FL))*20.
3. Output only the overall score.
# Few-Shot
Golden: Apple founded by Jobs/Wozniak (1976)
Candidate: Jobs and Wozniak co-founded Apple (1976), launching the
PC era
Output: 100
Golden: Atmospheric layers: troposphere/stratosphere/mesosphere/
thermosphere/exosphere
Candidate: Atmosphere: troposphere and stratosphere
Output: 20
Golden: Deep learning stages: data preprocessing/model
design/training/evaluation
Candidate: DL workflow: data cleaning/network design/training/testing
Output: 100
Golden: EV benefits: zero emissions/high efficiency/low maintenance
Candidate: EVs produce no pollutants but require charging networks
Output: 67
# Evaluation
Candidate: {model_response}
Golden: {best_answer}
```

following Eq. 3 in the main text:

$$r_{\text{quality}}(x, y) = (1 - \lambda_{\text{format}}) \cdot s_{\text{semantic}}(x, y) + \lambda_{\text{format}} \cdot \mathbf{1}_{\text{format}}(y). \tag{19}$$

Here, $s_{\text{semantic}}(x, y) \in [0, 1]$ is a normalized LLM-as-a-judge score measuring semantic equivalence between the candidate answer and the reference answer. It provides denser feedback than exact match (EM) and is more semantically sensitive than token-overlap F1, especially when the answer is semantically correct but differs lexically from the reference. The format compliance indicator $\mathbf{1}_{\text{format}}(y) \in \{0, 1\}$ checks whether the output follows the required tag structure. This term improves downstream parsing and reduces failures caused by missing or malformed answer tags.

The semantic judge outputs an integer score in $[0, 100]$, which is linearly normalized to $s_{\text{semantic}}(x, y) \in [0, 1]$.[3] The prompt used to compute $s_{\text{semantic}}(x, y)$ is shown in Table 6.

Retrieval-cost penalties. A reward based only on final-answer quality cannot distinguish efficient trajectories from trajectories that rely on repeated or excessive retrieval. We therefore penalize both the number of effective retrieval calls and repeated evidence retrieval. Let $N_{\text{search}}(y)$ be the number of valid closed <search>...</search> pairs that invoke the retrieval environment. Let $K = N_{\text{search}}(y)$, and let S_k denote the set of retrieved passages returned by the k-th retrieval call. We define the duplicate-retrieval count as:

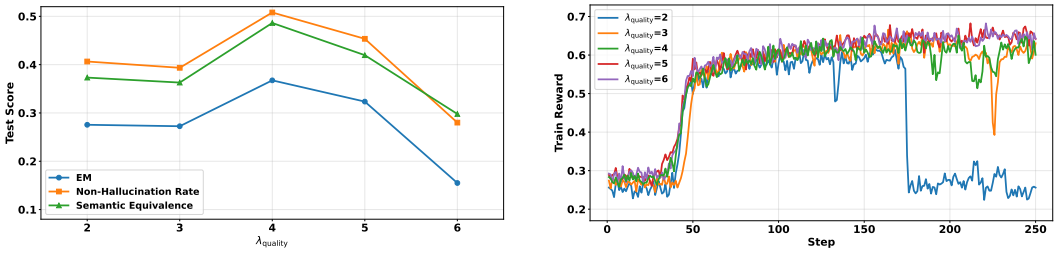
$$N_{\text{repeat}}(y) = \sum_{k=2}^{K} \mathbf{1}\left[\exists j < k : S_k = S_j\right]. \tag{20}$$

A retrieval call is counted as repeated when its returned passage set exactly matches that of an earlier call. This penalty encourages information gain rather than repeated access to the same evidence.

[3] This judge compares the candidate answer with the reference answer. It is not an external-evidence factuality judge and does not evaluate retrieved evidence. Format compliance is handled separately by $\mathbf{1}_{\text{format}}(y)$.

Table 7: Final-answer quality alone cannot distinguish retrieval efficiency. The retrieval-cost terms in $R_{\text{episode}}(x, y)$ separate these trajectories.

Trajectory	Description	$r_{\text{quality}}(x, y)$
1	One effective retrieval call followed by a correct and well-formatted answer.	1
2	The maximum allowed number of retrieval calls followed by a correct and well-formatted answer.	1



(a) Validation metrics under different λ_{quality} values.

(b) Normalized Phase I reward curves during training.

Figure 3: Sensitivity analysis of the answer-quality weight λ_{quality}. The left panel reports EM, NHR, and semantic equivalence for $\lambda_{\text{quality}} \in \{2, 3, 4, 5, 6\}$. The right panel reports normalized episode-level reward $R_{\text{episode}}(x, y)$ over training. The model is Qwen2.5-7B-Instruct and the RL algorithm is PPO.

The full Phase I episode-level reward follows Eq. 4:

$$R_{\text{episode}}(x, y) = \lambda_{\text{quality}} \cdot r_{\text{quality}}(x, y) - \rho_{\text{search}} \cdot N_{\text{search}}(y) - \rho_{\text{repeat}} \cdot N_{\text{repeat}}(y). \tag{21}$$

Here, $\rho_{\text{search}} > 0$ penalizes each effective retrieval call, while $\rho_{\text{repeat}} > 0$ adds a stronger penalty to duplicate retrieval. This separates potentially useful retrieval from duplicate retrieval with no new evidence.[4]

B.4 Phase I Reward Scaling Calibration

We calibrate the Phase I reward scale on a validation set by varying λ_{quality}. Unless otherwise stated, we fix $\rho_{\text{search}} = 0.1$, $\rho_{\text{repeat}} = 0.2$, and $\lambda_{\text{format}} = 0.4$. We scan $\lambda_{\text{quality}} \in \{2, 3, 4, 5, 6\}$ and select the value using validation EM, NHR, and semantic equivalence.

Figure 3 summarizes the calibration. Performance first improves as λ_{quality} increases and peaks at $\lambda_{\text{quality}} = 4$. Smaller values make retrieval penalties too dominant and can suppress necessary retrieval. Larger values weaken the cost constraint and can encourage less selective retrieval. The training reward curves show faster and more stable optimization when $\lambda_{\text{quality}} \geq 4$, but validation performance drops when λ_{quality} becomes too large. We therefore use $\lambda_{\text{quality}} = 4$ for the 7B setting.

B.5 Phase I Training Objective and Policy Optimization

Phase I optimizes the policy π_θ by maximizing the expected episode-level reward under the retrieval environment $\mathcal{R}$, with a KL penalty to the reference policy π_{ref}:

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}} \Big[\mathbb{E}_{y \sim \pi_\theta(\cdot|x;\mathcal{R})} [R_{\text{episode}}(x, y)] - \beta \mathbb{D}_{\text{KL}} \big(\pi_\theta(\cdot \mid x; \mathcal{R}) \| \pi_{\text{ref}}(\cdot \mid x; \mathcal{R}) \big) \Big]. \tag{22}$$

We optimize this objective with PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024), following the Search-R1 training setup (Jin et al., 2025). The resulting base policy is denoted by $\pi_{\theta_{\text{stage1}}}$.

[4]We set $\rho_{\text{repeat}} > \rho_{\text{search}}$ to distinguish retrieval with possible information gain from duplicate retrieval that returns the same passage set.

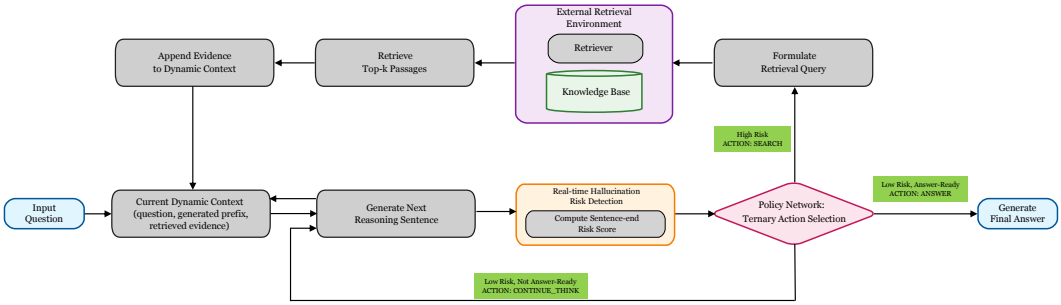


Figure 4: Phase II rollout and training-time reward assignment in Grounded-R1. The policy rollout is shared by training and inference, whereas RHRD provides reward supervision only during Phase II training and is not a policy input at inference.

B.6 Phase II Sentence-end Decision Construction

Phase II targets retrieval trigger timing directly by applying a step-level reward at sentence-end decision points. At each sentence-end time step t, given the current dynamic context, the policy generates y_t and selects an action from

$$\{\texttt{CONTINUE_THINK}, \texttt{ANSWER}, \texttt{SEARCH}\}.$$

For sentence-level reward assignment, generated text is first segmented using spaCy `en_core_web_sm` via `Doc.sents`, then refined at RAG control-tag boundaries. Closing tags are attached to the preceding span, while search-query, retrieved-information, final-answer, empty, and terminal-marker spans are excluded. Character offsets align each eligible reasoning span to the generated tokens, and its step reward is assigned to the last aligned token.

During Phase II training, the trainer then computes $\mathrm{RHRD}_{\mathrm{sent}}(t)$ for y_t, thresholds the score at τ, and uses the resulting risk state solely to assign the corresponding $r_{\mathrm{step}}(t)$ to the sentence-ending token of y_t. Through policy optimization, this training signal shapes the policy to associate evolving dynamic-context states with risk-aware retrieval actions. Neither the score nor the threshold is used during standard Grounded-R1 inference.

These actions are realized through the structured RAG generation interface: `SEARCH` and `ANSWER` are identified by their control-tag transitions, whereas `CONTINUE_THINK` denotes remaining in the current `<think>` segment. The action `CONTINUE_THINK` continues reasoning in the `<think>` segment without retrieval. The action `SEARCH` closes the current thinking segment and triggers retrieval through the `<search>` segment. The action `ANSWER` closes reasoning, writes the final answer through the `<answer>` segment, and terminates the episode.

Figure 4 separates the policy rollout, which is shared by training and inference, from the reward-assignment branch. After a retrieval call, the returned evidence is appended to the dynamic context through the `<information>` segment, and the next sentence is generated under the updated context. Retrieved-token loss masking is applied as in Phase I.

B.7 Phase II Sentence-level Reward and Hyperparameters

The Phase II reward uses the RHRD sentence-level score as a training signal. During Phase II training, the trainer compares $\mathrm{RHRD}_{\mathrm{sent}}(t)$ with the sentence-end risk threshold τ and assigns the reward for the selected action a_t as follows:

$$r_{\mathrm{step}}(t) = \begin{cases} +\alpha, & \text{if } \mathrm{RHRD}_{\mathrm{sent}}(t) \geq \tau \wedge a_t = \texttt{SEARCH}, \\ -\alpha, & \text{if } \mathrm{RHRD}_{\mathrm{sent}}(t) \geq \tau \wedge a_t \in \{\texttt{CONTINUE_THINK}, \texttt{ANSWER}\}, \\ +\gamma, & \text{if } \mathrm{RHRD}_{\mathrm{sent}}(t) < \tau \wedge a_t \in \{\texttt{CONTINUE_THINK}, \texttt{ANSWER}\}, \\ -\gamma, & \text{if } \mathrm{RHRD}_{\mathrm{sent}}(t) < \tau \wedge a_t = \texttt{SEARCH}. \end{cases} \tag{23}$$

Here, a_t is the action selected at sentence-end time step t, and $\alpha, \gamma \in \mathbb{R}^+$ are reward magnitudes. In our main setting, we use $\alpha = 2$ and $\gamma = 1$. This implements the asymmetric design $\alpha > \gamma$, since a

Table 8: Training-time sentence-level reward mapping used in Phase II.

Sentence-end risk state	Action	Reward	Decision meaning
$\mathrm{RHRD}_{\mathrm{sent}}(t) \geq \tau$	`SEARCH`	$+\alpha$	Correct retrieval under potentially high local hallucination risk.
$\mathrm{RHRD}_{\mathrm{sent}}(t) \geq \tau$	`CONTINUE_THINK` or `ANSWER`	$-\alpha$	False negative retrieval decision.
$\mathrm{RHRD}_{\mathrm{sent}}(t) < \tau$	`CONTINUE_THINK` or `ANSWER`	$+\gamma$	Correct non-retrieval under low risk.
$\mathrm{RHRD}_{\mathrm{sent}}(t) < \tau$	`SEARCH`	$-\gamma$	False positive retrieval decision.

false negative retrieval decision under high risk is more harmful than a false positive retrieval decision under low risk.

This reward does not treat $\mathrm{RHRD}_{\mathrm{sent}}(t)$ as a direct post-hoc hallucination label. It treats the thresholded RHRD score as a sentence-end reward-assignment signal. When the signal indicates potentially high local hallucination risk, the policy is encouraged to retrieve. Otherwise, it is encouraged to continue reasoning or answer without unnecessary retrieval.

The grouping of `CONTINUE_THINK` and `ANSWER` applies only to the local retrieval decision, since both actions avoid retrieval at the current sentence-end position. Their overall returns remain distinct: `ANSWER` terminates the episode and fixes the final output, whereas `CONTINUE_THINK` reaches subsequent decision points and can accrue future sentence-level rewards. The episode-level quality-cost reward further evaluates the resulting trajectory and final answer.

B.8 Validation Calibration of the Sentence-end Risk Threshold

The sentence-end risk threshold τ separates higher-risk and lower-risk sentence-end positions for reward assignment. We calibrate τ on a validation set rather than tuning it on test benchmarks.

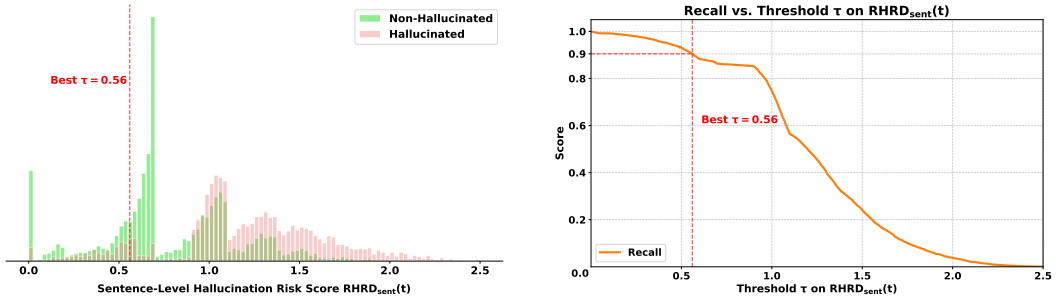
The calibration set contains two subsets. The high-risk subset consists of 5,000 fictional-entity questions sampled from HalluLens (Bang et al., 2025). These questions are designed to induce factuality hallucination under unsupported generation. The low-risk subset contains 5,000 commonsense questions generated by ChatGPT, where the answers are expected to fall within common parametric knowledge. We run the Phase I base policy on these questions and collect generated sentences with their $\mathrm{RHRD}_{\mathrm{sent}}(t)$ scores.

Each sentence is then labeled by an automatic two-stage annotation pipeline. The first stage performs factuality verification with external evidence from multiple channels.[5] A Qwen2.5-72B-Instruct judge determines whether each verifiable sentence is supported, contradicted, or insufficiently supported by the collected evidence. The second stage checks faithfulness against the input instruction, context constraints, retrieved evidence, and local generation context using the same judge model. A sentence is labeled as containing factuality hallucination if it is contradicted by or insufficiently supported by external evidence for a verifiable claim. It is labeled as containing faithfulness hallucination if it is inconsistent with the instruction, retrieved evidence, or local context. A sentence is labeled as non-hallucinated only if neither condition holds.

Figure 5 shows the calibration results. The score distributions are not perfectly separable, which supports using $\mathrm{RHRD}_{\mathrm{sent}}(t)$ as an operational proxy signal rather than a direct hallucination label. We therefore choose τ by scanning thresholds under a recall-oriented objective. As τ increases, recall on hallucinated sentences decreases. We choose $\tau = 0.56$ because it retains about 0.9 recall for hallucinated sentences. This value is fixed for Phase II training, the RHRD hard-threshold control, and the frozen RHRD proxy labels used in the RHRD-alignment diagnostic.

To directly evaluate the discriminative validity of RHRD against the sentence-level hallucination labels used for calibration, we additionally report threshold-independent and operating-point metrics

[5]For factuality verification, we use Google Search, Google News, Google Scholar through SerpAPI, Wikipedia API, and DuckDuckGo Search. All retrieval calls were conducted from November to December 2025.



(a) Score distributions for hallucinated and non-hallucinated sentences.

(b) Recall of hallucinated sentences under different thresholds.

Figure 5: Validation calibration of the sentence-end risk threshold τ. The selected threshold $\tau = 0.56$ prioritizes recall of hallucinated sentences and is fixed before test-time evaluation.

Table 9: RHRD discrimination and operating-point metrics on the threshold-calibration validation set. Higher is better for AUROC, AUPRC, Precision, and Recall; lower is better for FPR.

Metric	**Value**
AUROC	0.760703
AUPRC	0.762547
Precision @ $\tau = 0.56$	0.535500
Recall / TPR @ $\tau = 0.56$	0.904741
FPR @ $\tau = 0.56$	0.784785

in Table 9. RHRD achieves an AUROC of 0.7607 and an AUPRC of 0.7625, indicating meaningful discrimination despite the overlap between hallucinated and non-hallucinated score distributions. At the selected threshold $\tau = 0.56$, recall reaches 0.9047, while precision and FPR are 0.5355 and 0.7848, respectively. The relatively high FPR reflects the deliberately recall-oriented calibration and indicates that $\tau = 0.56$ is not intended as a selective stand-alone retrieval controller. Indeed, directly applying the same threshold at inference increases the average number of retrieval calls to 4.12, whereas Grounded-R1 (full) averages 2.04 calls, only slightly above the 2.00 calls of Grounded-R1 (episode-semantic) (Table 2; Appendix K). This contrast motivates using RHRD as sentence-level reward supervision within the mixed objective and KL-constrained Phase II optimization, which trade the recall-oriented local risk signal against answer quality and retrieval cost rather than mechanically reproducing threshold decisions.

B.9 Mixed Reward, Policy Optimization, and Capability Preservation

Phase II combines the sentence-level reward with the Phase I episode-level reward:

$$R_{\text{total}}(x, y) = \lambda_{\text{episode}} \cdot R_{\text{episode}}(x, y) + \sum_{t \in \mathcal{T}_{\text{end}}} r_{\text{step}}(t). \tag{24}$$

Here, $\mathcal{T}_{\text{end}}$ is the set of sentence-end decision time steps in trajectory y. We set $\lambda_{\text{episode}} = 1$ by default. The episode-level term keeps answer quality and retrieval cost in the objective, while the sentence-level term gives denser feedback for retrieval trigger timing.

The Phase II objective follows Eq. 8:

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}} \Big[\mathbb{E}_{y \sim \pi_\theta(\cdot|x;\mathcal{R})} [R_{\text{total}}(x, y)] - \beta \mathbb{D}_{\text{KL}} \big(\pi_\theta(\cdot \mid x; \mathcal{R}) \| \pi_{\theta_{\text{stage1}}}(\cdot \mid x; \mathcal{R}) \big) \Big]. \tag{25}$$

We set $\beta = 0.05$ in the main Phase II setting. The reference policy is the converged Phase I policy $\pi_{\theta_{\text{stage1}}}$.

Table 10: Benchmark usage map. "Episode" denotes episode-level final-answer evaluation. "RHRD alignment" denotes sentence-level RHRD-alignment diagnostic. "Sent. hallucination" denotes sentence-level hallucination analysis. "Trigger timing" denotes the matched-prefix intervention and equal-budget retrieval benefit evaluation. TruthfulQA is an additional benchmark for the RHRD-alignment diagnostic, while HalluLens is used only for threshold calibration.

Dataset	Split	Raw #	Eval. #	Answer type	Episode	RHRD alignment	Sent. hallucination	Trigger timing	Reported in
HotpotQA (Yang et al., 2018)	fullwiki	105K	5K	single-answer / multi-hop	Yes	Yes	No	No	main
2WikiMultiHopQA (Ho et al., 2020)	dev	12.5K	5K	single-answer / multi-hop	Yes	No	No	No	main
MuSiQue (Trivedi et al., 2022)	default	44.7K	5K	single-answer / multi-hop	Yes	Yes	No	No	main
Bamboogle (Press et al., 2023)	test	125	125	single-answer / two-hop	Yes	Yes	Yes	Yes	main
PopQA (Mallen et al., 2023)	test	14.3K	5K	multi-answer / entity QA	Yes	Yes	Yes	Yes	main
TriviaQA (Joshi et al., 2017)	RC validation	17.9K	5K	multi-answer / open-domain QA	Yes	No	Yes	Yes	main
ASQA (Stelmakh et al., 2022)	dev	948	500	long-form answer	Yes	No	Yes	Yes	main
HaluEval (Li et al., 2023)	test	10K	2K	hallucination label / QA	Yes	Yes	Yes	Yes	main
TruthfulQA (Lin et al., 2022)	benchmark	791	791	diagnostic QA	No	Yes	No	No	appendix only
HalluLens (Bang et al., 2025)	calibration subset	15K	5K	hallucination induction	No	No	No	No	calibration only

Phase II uses three safeguards to preserve the base generation capability. First, the mixed reward retains $R_{\text{episode}}(x, y)$, which keeps answer quality and retrieval cost aligned with Phase I.[6] Second, the KL term constrains the updated policy to remain close to $\pi_{\theta_{\text{stage1}}}$. Third, Phase II is initialized from $\pi_{\theta_{\text{stage1}}}$ and uses conservative updates with longer warmup and one policy-update epoch per batch. Together, these choices allow the policy to learn hallucination-risk-aligned sentence-end retrieval triggering decisions without reducing Phase II to a deterministic imitation of the RHRD threshold, while staying close to the high-quality, low-cost behavior learned in Phase I.

Role of the RHRD proxy. The sentence-level reward provides local risk-alignment supervision rather than an estimate of the counterfactual downstream utility of retrieval at each sentence-end state. Accordingly, Phase II is not optimized to reproduce the binary RHRD threshold policy in isolation. The RHRD-derived step-level reward is jointly optimized with the episode-level quality-cost reward, while the KL constraint anchors the updated policy to the Phase I policy. The learned policy can therefore condition retrieval on the evolving dynamic context while trading local hallucination-risk alignment against answer quality and retrieval cost, rather than mechanically applying the threshold rule. We evaluate the downstream utility of retrieval timing separately through the controlled timing interventions in Section 4.5, whose outcomes do not use RHRD-derived proxy labels.

C Benchmarks, Dataset Statistics, and Preprocessing

This section reports the benchmark usage map, dataset statistics, and preprocessing rules used in the main experiments and appendix diagnostics. We separate episode-level evaluation, the sentence-level RHRD-alignment diagnostic, sentence-level hallucination analysis, causal retrieval trigger timing evaluation, and threshold calibration because they serve different purposes.

C.1 Benchmark Usage Map

Table 10 summarizes the benchmark pool. Eight datasets are used for episode-level ground-truth evaluation. ASQA, Bamboogle, HaluEval, PopQA, and TriviaQA form the shared benchmark set for sentence-level hallucination analysis and both causal retrieval trigger timing experiments. TruthfulQA is used as an additional benchmark for the RHRD-alignment diagnostic and is not included in the eight ground-truth episode-level benchmarks. HalluLens is used only for calibrating the sentence-end risk threshold and is not used for model comparison.

C.2 Dataset Statistics and Preprocessing Protocol

We use the base evaluated sample counts in Table 10 after preprocessing and sampling. For a dataset D, we report the retained ratio as:

$$\text{RetainedRatio}(D) = \frac{N_{\text{eval}}(D)}{N_{\text{raw}}(D)}. \tag{26}$$

[6] This preservation mechanism is used to reduce catastrophic forgetting, where learning a new objective can degrade previously learned capabilities. In this work, the goal is to avoid loss of answer quality or uncontrolled retrieval cost while improving retrieval trigger timing.

Table 11: Preprocessing and failure-handling rules for benchmark evaluation. NHR denotes non-hallucination rate.

Item	Applied	Rule
Deduplication	Yes	Remove duplicate questions, duplicate identifiers, or duplicate-answer records when such fields are available.
Missing-answer filtering	Yes	Remove samples without a reference answer, with an empty answer field, or with missing ground-truth references.
Abnormal-format handling	Yes	For methods using the structured think–search–answer interface, invalid or incomplete actions receive a format-correction observation and generation continues within the configured interaction-turn limit. The final answer is extracted from the last complete `<answer>...</answer>` span; if no such span is produced, the prediction is set to `None`.
Overlong-output handling	Yes	Outputs exceeding the maximum response length are truncated. If the final round still yields no valid extracted answer, the answer is set to `None`.
Retrieval-failure handling	Yes	If the retriever fails, returns an empty result, or returns fewer than top-k documents, the retrieved content is set to `None`.
Automatic annotation failure	Yes	If LLM-as-a-judge scoring or NHR judging fails to parse a valid output, the corresponding score is set to 0.
Maximum retrieval budget	Yes	We allow at most five retrieval calls for one response during evaluation.

Table 12: Dataset statistics used in the experiments and calibration. The retained ratio is computed by Eq. 26. The evaluated counts denote the base benchmark pools, while timing-specific branch-point and candidate-position counts are reported separately. TruthfulQA is used as an additional benchmark for the sentence-level RHRD-alignment diagnostic, and HalluLens is used only for calibration.

Dataset	Raw #	Evaluated #	Retained ratio	Primary use	Notes
HotpotQA	105K	5K	4.76%	episode / RHRD alignment	sampled from the fullwiki split
2WikiMultiHopQA	12.5K	5K	40.00%	episode	sampled from the dev split
MuSiQue	44.7K	5K	11.19%	episode / RHRD alignment	sampled from the default split
Bamboogle	125	125	100.00%	episode / RHRD alignment / sentence-level hallucination / trigger timing	full test set
PopQA	14.3K	5K	34.97%	episode / RHRD alignment / sentence-level hallucination / trigger timing	sampled from the test split
TriviaQA	17.9K	5K	27.93%	episode / sentence-level hallucination / trigger timing	sampled from the RC validation split
ASQA	948	500	52.74%	episode / sentence-level hallucination / trigger timing	sampled from the dev split
HaluEval	10K	2K	20.00%	episode / RHRD alignment / sentence-level hallucination / trigger timing	sampled from the test split
TruthfulQA	791	791	100.00%	RHRD alignment	additional benchmark for the sentence-level RHRD-alignment diagnostic
HalluLens	15K	5K	33.33%	threshold calibration	calibration subset only

Here, $N_{\mathrm{raw}}(D)$ is the raw candidate size and $N_{\mathrm{eval}}(D)$ is the number of examples retained in the base evaluation pool.

Table 11 gives the preprocessing and failure-handling rules. We apply deduplication and missing-answer filtering before evaluation. Generation-time failures are handled without changing the evaluated input set.

Table 12 reports the raw and base evaluated sample counts. The causal retrieval trigger timing experiments operate on sentence-end branch points or candidate positions drawn from the same five-benchmark pool as the sentence-level hallucination analysis. Their timing-specific counts are reported in Tables 29 and 36.

D Evaluation Protocols and Judge Prompts

This appendix gives the evaluation details used in the main experiments and appendix diagnostics. We distinguish episode-level answer evaluation, the sentence-level RHRD-alignment diagnostic, sentence-level hallucination evaluation, and causal retrieval trigger timing evaluation because they correspond to different evaluation targets. Final-answer metrics evaluate episode-level task success. RHRD-alignment metrics measure agreement between sentence-end policy actions and the frozen RHRD proxy. Sentence-level hallucination metrics evaluate local grounded generation. Retrieval trigger timing metrics measure the downstream benefit of timely retrieval and selector-conditioned allocation under equal retrieval budgets.

Both causal retrieval trigger timing experiments use the same five-benchmark set as the sentence-level hallucination evaluation. The matched-prefix intervention evaluates 5,000 branch points, while the equal-budget retrieval benefit evaluation ranks a shared pool of 2,000 candidate positions and assigns each selector an identical budget of 200 positions. The first downstream sentence hallucination rate and suffix NHR reuse the same sentence segmentation, factuality annotation, and faithfulness annotation protocols as the sentence-level NHR evaluation. Final EM reuses the episode-level answer extraction and normalization protocol.

Table 13: Evaluation scope across the main experiments and appendix diagnostics. "RHRD alignment" denotes sentence-level RHRD-alignment diagnostic. "Trigger timing" denotes the matched-prefix intervention and equal-budget retrieval benefit evaluation. NHR denotes non-hallucination rate. TruthfulQA is used only for the sentence-level RHRD-alignment diagnostic.

Evaluation	Benchmarks	Metrics
Episode-level answer and cost evaluation	HotpotQA, 2WikiMultiHopQA, MuSiQue, Bamboogle, PopQA, TriviaQA, ASQA, HaluEval	EM, F1, LLM-as-a-judge score, HaluEval NHR, and the average number of retrieval calls
Sentence-level RHRD-alignment diagnostic	Bamboogle, HaluEval, HotpotQA, MuSiQue, PopQA, TruthfulQA	Precision, Recall, FPR, and F2 score
Sentence-level hallucination evaluation	ASQA, Bamboogle, HaluEval, PopQA, TriviaQA	Factuality NHR, faithfulness NHR, overall NHR
Causal retrieval trigger timing evaluation	ASQA, Bamboogle, HaluEval, PopQA, TriviaQA	First downstream sentence hallucination rate, suffix NHR, final EM, and ASQA F1 for the matched-prefix intervention; Immediate − Delayed suffix NHR for the equal-budget retrieval benefit evaluation

For all final-answer metrics, each method is reduced to its final response before scoring, and the same evaluation pipeline is applied across methods. Missing or `None` responses receive zero credit. Exact match (EM) and F1 are computed after standard answer normalization, including lowercasing, punctuation removal, article removal, and whitespace normalization. Cost is measured by the average number of retrieval calls when reported as an auxiliary cost metric. Duplicate retrieval is used in the training reward but is not reported as a main final-answer metric.

D.1 Episode-level and Response-level Metric Definitions

Let $\hat{a}$ denote the extracted prediction and let $\mathcal{A}$ denote the set of reference answers. For HotpotQA, 2WikiMultiHopQA, MuSiQue, Bamboogle, and HaluEval, each example has a single reference answer a. We define:

$$\mathrm{EM}(\hat{a}, a) = \mathbf{1}\{\mathrm{norm}(\hat{a}) = \mathrm{norm}(a)\}. \tag{27}$$

For PopQA, TriviaQA, and ASQA, multiple references may be available. We compute EM by matching any reference:

$$\mathrm{EM}(\hat{a}, \mathcal{A}) = \max_{a \in \mathcal{A}} \mathbf{1}\{\mathrm{norm}(\hat{a}) = \mathrm{norm}(a)\}. \tag{28}$$

Macro-average EM is the unweighted mean of the eight dataset-level EM values. Grounded-R1 (full) achieves 0.41075 and Search-R1 achieves 0.34800, corresponding to an 18.0% relative improvement and an absolute gain of 0.06275.

For datasets where F1 is reported, we compute token-level precision and recall after normalization. For one reference answer a, let $P(\hat{a}, a)$ and $R(\hat{a}, a)$ denote token-level precision and recall. The token-level F1 score is:

$$\mathrm{F1}(\hat{a}, a) = \frac{2P(\hat{a}, a)R(\hat{a}, a)}{P(\hat{a}, a) + R(\hat{a}, a)}. \tag{29}$$

If both precision and recall are zero, the F1 score is set to 0. For multiple-reference datasets, we take the maximum F1 over references:

$$\mathrm{F1}(\hat{a}, \mathcal{A}) = \max_{a \in \mathcal{A}} \mathrm{F1}(\hat{a}, a). \tag{30}$$

For semantic answer evaluation, we use Qwen2.5-72B-Instruct as the judge model. The judge compares the candidate answer with the reference answer and outputs a score from 0 to 100. We normalize the LLM-as-a-judge score to $[0, 1]$:

$$s_{\mathrm{judge}}(\hat{a}, a) = \frac{\mathrm{Score}_{\text{0-100}}(\hat{a}, a)}{100}. \tag{31}$$

For multiple-reference datasets, we use the reference answer selected by the evaluation script for judge scoring.

For HaluEval, we also report episode-level NHR. Let $u_i \in \{0, 1\}$ denote the binary judgment for sample i, where $u_i = 1$ means the answer is judged as non-hallucinated and $u_i = 0$ means the answer

is judged as hallucinated. For a HaluEval evaluation set with N examples, episode-level NHR is:

$$\mathrm{NHR}_{\mathrm{episode}} = \frac{1}{N}\sum_{i=1}^{N} u_i. \tag{32}$$

This episode-level NHR is reported on the $[0, 1]$ scale.

Retrieval cost is averaged over samples when reported:

$$\mathrm{RetrievalCalls} = \frac{1}{N}\sum_{i=1}^{N} N_{\mathrm{search}}(y_i), \tag{33}$$

where $N_{\mathrm{search}}(y_i)$ is the number of effective retrieval calls in output y_i.

D.2 Final-answer Judge and Episode-level NHR Judge Prompts

Final-answer judge prompt. We use Qwen2.5-72B-Instruct to compute the LLM-as-a-judge score. The judge receives one candidate answer and one reference answer, and returns only an integer score from 0 to 100. Invalid or unparsable judge outputs are assigned a score of 0.

```
# Role
You are an objective evaluator comparing a candidate response to a
golden answer.

# Instructions
1. Rate on five dimensions (0-5 integers):
   - 70% Semantic Accuracy
   - 7.5% Completeness
   - 7.5% Logical Coherence
   - 7.5% Clarity
   - 7.5% Fluency
2. Compute overall similarity (0-100):
   overall = (0.70*SA + 0.075*(CMP+LC+CLR+FL)) * 20
3. Output only the overall score. No other fields or text.

# Few-Shot
Golden: Apple founded by Jobs/Wozniak (1976)
Candidate: Jobs and Wozniak co-founded Apple (1976), launching the
PC era
-> 100

Golden: Atmospheric layers:
troposphere/stratosphere/mesosphere/thermosphere/exosphere
Candidate: Atmosphere: troposphere and stratosphere
-> 20

Golden: Deep learning stages: data preprocessing/model
design/training/evaluation
Candidate: DL workflow: data cleaning/network
design/training/testing
-> 100

Golden: EV benefits: zero emissions/high efficiency/low
maintenance
Candidate: EVs produce no pollutants but require charging networks
-> 67

# Evaluation
Candidate: {model_response}
Golden: {best_answer}
```

Episode-level NHR judge prompt. For HaluEval, Qwen2.5-72B-Instruct judges whether the final answer contains hallucination using the question, world knowledge, right answer, and model response. The output must be either `Yes` or `No`. Here, `Yes` means that the answer contains hallucination, and `No` means that it does not.

```
System:
You are a hallucination evaluator. You must determine whether the
provided
answer contains hallucination for the question based on the world
knowledge
and the right answer. Your output must be exactly "Yes" or "No".

User:
#Question#: {question}
#World knowledge#: {knowledge}
#Right answer#: {right_answer}
#Answer#: {model_response}
#Your Judgement#:
```

Sentence-level factuality and faithfulness annotation prompts are reported in Appendix H.

D.3 RHRD-Alignment Metric Definitions

We measure alignment between sentence-end policy actions and the frozen RHRD proxy at each sentence-end time step. For sample i and sentence-end time step t, the model decision is:

$$\hat{z}_{i,t} = \mathbf{1}\{a_{i,t} = \texttt{SEARCH}\}. \tag{34}$$

Both `CONTINUE_THINK` and `ANSWER` are treated as non-retrieval decisions.

We construct a proxy reference label with a frozen sentence-end risk threshold $\tau = 0.56$:

$$z_{i,t} = \mathbf{1}\{\mathrm{RHRD}_{\mathrm{sent}}(i,t) \geq \tau\}. \tag{35}$$

The label $z_{i,t}$ is a proxy reference label for the RHRD-alignment diagnostic. It is not a ground-truth hallucination label or a direct observation of whether retrieval is necessary. Threshold calibration is described in Appendix B.8.

For each sample i, we compute confusion counts over its evaluated sentence-end decision points:

$$TP_i = \sum_t \mathbf{1}\{z_{i,t} = 1 \wedge \hat{z}_{i,t} = 1\}, \tag{36}$$

$$FP_i = \sum_t \mathbf{1}\{z_{i,t} = 0 \wedge \hat{z}_{i,t} = 1\}, \tag{37}$$

$$FN_i = \sum_t \mathbf{1}\{z_{i,t} = 1 \wedge \hat{z}_{i,t} = 0\}, \tag{38}$$

$$TN_i = \sum_t \mathbf{1}\{z_{i,t} = 0 \wedge \hat{z}_{i,t} = 0\}. \tag{39}$$

At the sample level, precision, recall, false positive rate (FPR), false negative rate (FNR), and F2 score are:

$$\mathrm{Precision}_i = \frac{TP_i}{TP_i + FP_i}, \tag{40}$$

$$\mathrm{Recall}_i = \frac{TP_i}{TP_i + FN_i}, \tag{41}$$

$$\mathrm{FPR}_i = \frac{FP_i}{FP_i + TN_i}, \tag{42}$$

$$\mathrm{FNR}_i = \frac{FN_i}{TP_i + FN_i}, \tag{43}$$

$$\mathrm{F2}_i = \frac{5 \cdot \mathrm{Precision}_i \cdot \mathrm{Recall}_i}{4 \cdot \mathrm{Precision}_i + \mathrm{Recall}_i}. \tag{44}$$

F2 is the primary RHRD-alignment summary because the Phase II proxy objective places greater weight on missed proxy-positive sentence-end states.

D.4 Macro-averaging Protocol for Sentence-level Metrics

Sentence-level RHRD-alignment metrics are first computed at the sample level, then averaged within each dataset, and finally averaged across datasets. We use macro averages rather than micro averages because different benchmarks have different output lengths and different numbers of sentence-end decision points. This prevents long outputs or a single large benchmark from dominating the reported proxy alignment.

Suppose the RHRD-alignment diagnostic contains M_{align} datasets. Dataset j contains N_j samples. For a sample-level metric $m_{j,i}$, where $m \in \{\text{Precision}, \text{Recall}, \text{FPR}, \text{FNR}, \text{F2}\}$, the dataset-level macro metric is:

$$\text{Macro-}m_j = \frac{1}{N_j} \sum_{i=1}^{N_j} m_{j,i}. \tag{45}$$

The global macro metric is:

$$\text{Global-Macro-}m = \frac{1}{M_{\text{align}}} \sum_{j=1}^{M_{\text{align}}} \text{Macro-}m_j. \tag{46}$$

Table 22 reports dataset-level sample-macro metrics, and Table 23 reports their unweighted averages across datasets.

For sentence-level hallucination evaluation, each generated answer is split into $N_{j,i}$ sentences. Sentence k in sample i from dataset j has factuality hallucination label $h_{\text{fact},j,i,k} \in \{0, 1\}$ and faithfulness hallucination label $h_{\text{faith},j,i,k} \in \{0, 1\}$. We define:

$$h_{\text{overall},j,i,k} = \max(h_{\text{fact},j,i,k}, h_{\text{faith},j,i,k}). \tag{47}$$

The sample-level NHR metrics are:

$$\text{NHR}_{\text{fact},j,i} = \left(1 - \frac{1}{N_{j,i}} \sum_{k=1}^{N_{j,i}} h_{\text{fact},j,i,k}\right) \times 100\%, \tag{48}$$

$$\text{NHR}_{\text{faith},j,i} = \left(1 - \frac{1}{N_{j,i}} \sum_{k=1}^{N_{j,i}} h_{\text{faith},j,i,k}\right) \times 100\%, \tag{49}$$

$$\text{NHR}_{\text{overall},j,i} = \left(1 - \frac{1}{N_{j,i}} \sum_{k=1}^{N_{j,i}} h_{\text{overall},j,i,k}\right) \times 100\%. \tag{50}$$

A sentence is counted as overall non-hallucinated only when it contains neither factuality hallucination nor faithfulness hallucination.

For dataset j, the dataset-level sentence-level NHR is:

$$\text{NHR}_{q,j} = \frac{1}{N_j} \sum_{i=1}^{N_j} \text{NHR}_{q,j,i}, \quad q \in \{\text{fact}, \text{faith}, \text{overall}\}. \tag{51}$$

For the M_{hall} sentence-level hallucination benchmarks, the global macro sentence-level NHR is:

$$\text{Macro-NHR}_q = \frac{1}{M_{\text{hall}}} \sum_{j=1}^{M_{\text{hall}}} \text{NHR}_{q,j}, \quad q \in \{\text{fact}, \text{faith}, \text{overall}\}. \tag{52}$$

The main paper reports overall NHR. Factuality and faithfulness breakdowns are reported in Appendix H.

Table 14: Baseline implementations and evaluation scope. "Episode" denotes episode-level final-answer evaluation. "RHRD alignment" denotes sentence-level RHRD-alignment diagnostic. "Sent. hallucination" denotes sentence-level hallucination analysis. "Matched-prefix" denotes the matched-prefix intervention. "Equal-budget" denotes the equal-budget retrieval benefit evaluation.

Method	Category	Implementation	Episode	RHRD alignment	Sent. hallucination	Matched-prefix	Equal-budget
Direct Inference	no retrieval	Qwen2.5-7B-Instruct directly generates the answer under a zero-shot setting without external retrieval.	Yes	No	No	No	No
Static RAG (Lewis et al., 2020)	static RAG	Qwen2.5-7B-Instruct receives the top-3 Wikipedia documents retrieved by e5-base-v2 as reference context, follows the same structured reasoning format, and extracts the content inside `<answer>` as the final output.	Yes	No	No	No	No
FLARE (Jiang et al., 2023)	training-free dynamic RAG	FLARE uses confidence-based active retrieval: it predicts the next sentence, triggers retrieval when token probabilities fall below a predefined threshold, and regenerates the sentence using the retrieved evidence. We evaluate it with the same base model and top-3 retrieval setting.	Yes	No	No	No	No
DRAGIN (Su et al., 2024)	training-free dynamic RAG	DRAGIN uses RIND to estimate information need and QFS to construct retrieval queries from self-attention. We evaluate it with the same base model and top-3 retrieval setting.	Yes	No	No	No	No
SeaKR (Yao et al., 2025)	training-free dynamic RAG	SeaKR uses self-aware uncertainty to guide retrieval and evidence integration through self-aware re-ranking and self-aware reasoning.	Yes	No	No	No	No
ETC (Li et al., 2026)	training-free dynamic RAG	ETC selects retrieval timing from token-level entropy trends using first- and second-order entropy differences, without additional training.	Yes	No	Yes	No	No
IRCoT (Trivedi et al., 2023)	training-free dynamic RAG	IRCoT interleaves chain-of-thought reasoning with retrieval. Each generated reasoning sentence can guide the next retrieval step.	Yes	No	Yes	No	No
Search-o1 (Li et al., 2025)	training-free dynamic RAG	Search-o1 integrates an agentic retrieval workflow into reasoning and uses a Reason-in-Documents module to refine retrieved documents before injecting them into the reasoning chain.	No	No	Yes	No	No
Self-RAG (Asai et al., 2024)	learning-based dynamic RAG	Self-RAG learns to retrieve and critique through reflection tokens, enabling retrieval and self-assessment during generation.	No	No	Yes	No	No
Prompted Reasoning-Retrieval Qwen	prompted dynamic RAG	Qwen2.5-7B-Instruct uses the same reasoning-retrieval interleaving scaffold as Search-R1, but relies only on prompting rather than RL.	No	Yes	Yes	No	No
Search-R1 (Jin et al., 2025)	learning-based dynamic RAG	Search-R1 trains the reasoning-retrieval interleaving scaffold with RL. We use the public v0.3 setting for comparison.	Yes	Yes	Yes	No	Yes
R1-Searcher (Song et al., 2025)	learning-based dynamic RAG	R1-Searcher uses a two-stage outcome-based RL design. The first stage encourages retrieval invocation format, and the second stage adds answer reward for task solving.	Yes	Yes	Yes	No	No
Grounded-R1 (episode-semantic + RHRD hard-threshold control)	our inference-time control	This control reuses the selected 7B GRPO Grounded-R1 (episode-semantic) Phase I checkpoint without additional policy optimization. At each sentence-end position, it forces `SEARCH` when $\mathrm{RHRD}_{\mathrm{sent}}(t) \geq \tau$ and masks `SEARCH` otherwise, subject to the same maximum retrieval budget. It uses the same retrieval environment, top-3 evidence, decoding configuration, maximum retrieval budget, and answer extraction protocol as Grounded-R1 (full).	Yes	No	No	No	No
Grounded-R1 (episode-EM)	our ablation	This ablation uses terminal EM, format reward, and retrieval-cost penalties without the Phase II sentence-level reward.	Yes	Yes	Yes	No	No
Grounded-R1 (episode-semantic)	our episode-level variant	This ablation uses the LLM-as-a-judge score, format reward, and retrieval-cost penalties without the Phase II sentence-level reward.	Yes	Yes	Yes	No	Yes
Grounded-R1 (full)	our full model	This model adds the Phase II sentence-level reward to optimize retrieval trigger timing while retaining the episode-level quality-cost objective.	Yes	Yes	Yes	Yes	Yes

E Baseline Implementation and Evaluation Scope

This section summarizes the baseline implementations and evaluation scope across the main experiments and appendix diagnostics. The episode-level comparison uses the full baseline set reported in the main experiments. The sentence-level RHRD-alignment diagnostic includes methods with recoverable sentence-end retrieval decisions. The sentence-level hallucination analysis includes methods whose outputs can be segmented and annotated under the same protocol. The causal retrieval trigger timing evaluation includes the matched-prefix intervention and equal-budget retrieval benefit evaluation.

Throughout the paper, Grounded-R1 (episode) denotes Grounded-R1 (episode-semantic). We use the shortened name to distinguish the episode-level policy from Grounded-R1 (full), which further adds the Phase II sentence-level reward. In ablation tables, we keep Grounded-R1 (episode-EM) and Grounded-R1 (episode-semantic) separate.

All methods use the same final-response scoring protocol described in Appendix D.

F Reproducibility, Training Data, Retrieval Environment, and Compute

This section reports the training data, retrieval environment, hyperparameters, compute, and checkpoint selection protocol used in Section 4. We follow the Search-R1 training and retrieval setup whenever possible, while adding the sentence-level reward used by Grounded-R1 (full).

F.1 Training Data Sources and Stage-wise Supervision

Both Phase I and Phase II use the merged Natural Questions (NQ) and HotpotQA training data released in `PeterJinGo/nq_hotpotqa_train`, following the Search-R1 setup (Jin et al., 2025; Kwiatkowski et al., 2019; Yang et al., 2018). The data include approximately 170k training

Table 15: Training data and supervision protocol. The two phases use the same NQ and HotpotQA data distribution.

Item	Setting
Training dataset	`PeterJinGo/nq_hotpotqa_train`
Source tasks	Natural Questions and HotpotQA
Training split	170k examples in `train.parquet`
Held-out split from training source	51.7k examples in `test.parquet`
Phase I supervision	Episode-level quality-cost reward
Phase II supervision	Mixed reward with sentence-level reward at sentence-end decision points
Training batch size	512
Validation batch size	256

Table 16: Unified retrieval environment used for Grounded-R1 and retrieval-based baselines whenever possible.

Item	Setting
Knowledge base	2018 Wikipedia dump
Corpus	`https://huggingface.co/datasets/PeterJinGo/wiki-18-corpus`
Index	`https://huggingface.co/datasets/PeterJinGo/wiki-18-e5-index`
Retriever backbone	`e5-base-v2`
Retriever URL	`https://huggingface.co/intfloat/e5-base-v2`
Retrieval backend	Dense retrieval with FAISS-GPU
Similarity	Cosine similarity
Retrieved unit	Wikipedia document
Top-k	3
Retriever update	Fixed during training and evaluation

examples in `train.parquet` and 51.7k held-out examples from the same training source in `test.parquet`. We use the same data distribution across the two phases.

Phase I uses episode-level supervision. Its reward is the episode-level quality-cost reward in Eq. 4, which combines final-answer quality, format compliance, the number of effective retrieval calls, and duplicate-retrieval penalties. Phase II starts from the Phase I policy and adds the sentence-level reward in Eq. 6. The additional reward is computed at sentence-end decision points where the RHRD sentence-level score is available. This keeps the task distribution fixed while adding a sentence-level learning signal to the episode-level objective.

F.2 Unified Retrieval Environment

All retrieval-based methods use the same retrieval environment whenever possible. We use the 2018 Wikipedia dump as the external knowledge base. The retrieval corpus and index follow the released `wiki-18-corpus` and `wiki-18-e5-index` resources. We encode questions and text with `e5-base-v2` and perform dense retrieval with FAISS-GPU (Wang et al., 2022; Johnson et al., 2021). The retriever returns the top-3 Wikipedia documents by cosine similarity. The retriever and index are fixed during training and evaluation.

F.3 Training Hyperparameters and Compute Details

We train Qwen2.5 models of different sizes, including 3B-Instruct and 7B-Instruct, with proximal policy optimization (PPO) and group relative policy optimization (GRPO) (Schulman et al., 2017; Shao et al., 2024). PPO samples one candidate trajectory for each input and uses a critic network for value estimation. GRPO samples five candidate trajectories for each input and scores each trajectory with the reward function. Rollouts are served with vLLM. The sampling temperature is 1.0 and top-p is 1.0. In the phase-specific objectives, β is set to 0.001 in Phase I and 0.05 in Phase II, corresponding to β_{I} and β_{II} in Table 17, respectively.

The 7B training runs use 8 NVIDIA H200 140GB GPUs. The retrieval service occupies about 4% of GPU memory in all runs. Grounded-R1 (episode-semantic) and Grounded-R1 (full) additionally

Table 17: Training hyperparameters for PPO and GRPO runs. Critic settings apply only to PPO.

Component	Setting
Actor optimizer	AdamW
Actor learning rate	Phase I: $1e{-6}$ (PPO), $5e{-7}$ (GRPO); Phase II: $1e{-6}$
Actor warmup ratio	Phase I: 0.285; Phase II: 0.9
Critic optimizer	AdamW
Critic learning rate	Phase I: $1e{-5}$; Phase II: $5e{-6}$
Critic warmup ratio	Phase I: 0.015; Phase II: 0.05
Phase I KL coefficient β_{I}	0.001
Phase II KL coefficient β_{II}	0.05
Training batch size	512
Validation batch size	256
Actor micro batch size	64
Rollout log-probability micro batch size	128
Policy mini-batch size	256
Phase II policy-update epochs per batch	1
Rollout engine	vLLM
Sampling temperature	1.0
Top-p	1.0
Maximum training interaction turns	5
Maximum training retrieval calls	5
Maximum prompt length	4096
Maximum response length	500
Maximum start length	2048
Maximum observation length	500
Retrieval top-k	3

Table 18: Training schedule and checkpoint saving intervals for Grounded-R1 variants.

Variant	Algorithm	Epochs / steps	Checkpoint interval
Grounded-R1 (episode-EM)	PPO	15 / 500	100
Grounded-R1 (episode-EM)	GRPO	15 / 300	50
Grounded-R1 (episode-semantic)	PPO	15 / 500	100
Grounded-R1 (episode-semantic)	GRPO	15 / 300	50
Grounded-R1 (full)	PPO	15 / 300	50
Grounded-R1 (full)	GRPO	15 / 200	25

launch the Qwen2.5-72B-Instruct judge with vLLM. Table 20 summarizes approximate peak GPU memory usage and average step time observed in our runs.

End-to-end policy optimization cost. Table 21 complements the per-step measurements above by reporting the total wall-clock time and H200 GPU-hours for the complete training paths. Our Search-R1 reproduction follows a single episode-level RL stage, whereas Grounded-R1 (full) sequentially optimizes Phase I and Phase II. Grounded-R1 therefore incurs higher training cost, but the increase is quantified and bounded: the full two-phase path requires $1.64\times$ the compute of Search-R1 under PPO and $1.88\times$ under GRPO, remaining below twice the Search-R1 cost on the same hardware in both cases. This increase is a one-time policy optimization cost; the trained policy is reused across evaluation and deployment queries, and neither training phase is rerun at inference. In return, the main evaluation reports an 18.0% relative macro-EM improvement over the strongest external baseline with comparable retrieval calls, indicating a favorable trade-off between additional training compute and downstream final-answer performance.

The retrieval service is started before training. The Qwen2.5-72B-Instruct judge service is used by Grounded-R1 (episode-semantic) and Grounded-R1 (full).

```
bash retrieval_launch.sh
bash vllm_72b.sh
```

Table 19: Reward weights for Grounded-R1 variants. PPO and GRPO use the same reward weights for each variant. The penalty notation follows Eq. 4, and τ follows Eq. 6.

Weight	Grounded-R1 (episode-EM)	Grounded-R1 (episode-semantic)	Grounded-R1 (full)
Format-compliance weight λ_{format}	0.4	0.4	0.4
Answer-quality weight λ_{quality}	4	4	4
Effective retrieval-call penalty ρ_{search}	0.1	0.1	0.1
Duplicate-retrieval penalty ρ_{repeat}	0.2	0.2	0.2
Episode reward weight λ_{episode}	–	–	1
Sentence-end risk threshold τ	–	–	0.56

Table 20: Approximate compute summary for the main 7B training runs on 8 NVIDIA H200 140GB GPUs.

Run	GPU memory composition	Peak usage	Average step time
Grounded-R1 (episode-EM) PPO	Retrieval 4%, training model 66%	70%	10 min
Grounded-R1 (episode-EM) GRPO	Retrieval 4%, training model 66%	70%	11 min
Grounded-R1 (episode-semantic) PPO	Judge 19%, retrieval 4%, training model 52%	75%	8 min
Grounded-R1 (episode-semantic) GRPO	Judge 25%, retrieval 4%, training model 43%	72%	10 min
Grounded-R1 (full) PPO	Judge 20%, retrieval 4%, training model 21%	62%	14 min
Grounded-R1 (full) GRPO	Judge 25%, retrieval 4%, training model 23%	64%	16 min

F.4 Validation, Checkpoint Selection, and Leakage Control

We select checkpoints using a held-out HaluEval validation subset of 500 examples. The selection criterion uses EM and NHR, matching the final-answer correctness and hallucination-related goals in Section 4. The RHRD hard-threshold control reuses the selected Grounded-R1 (episode-semantic) Phase I checkpoint without separate controller-specific checkpoint selection. The sentence-end risk threshold is fixed to $\tau = 0.56$ before test evaluation for Phase II reward construction, the hard-threshold control, and the RHRD-alignment diagnostic protocol.

Training uses the NQ and HotpotQA mixture described in Appendix F.1. Test benchmark labels are not used to construct training rewards. The retrieval corpus is a fixed 2018 Wikipedia snapshot and is accessed only through the retrieval environment. Gold answers from final test subsets are used only for reporting evaluation metrics. Checkpoint selection uses only the stated held-out HaluEval validation subset, and the final reported HaluEval subset is kept separate from this selection subset.

G RHRD-Alignment Diagnostics for Retrieval Triggering

This section evaluates whether sentence-end policy actions align with the frozen Real-time Hallucination Risk Detection (RHRD) proxy used to construct the Phase II sentence-level learning signal. At each sentence-end decision point, a state is assigned the label `should-retrieve` when $\text{RHRD}_{\text{sent}}(t) \geq 0.56$ and `should-not-retrieve` otherwise. The model decision is read from the generated action tag. `SEARCH` is counted as retrieval, while `CONTINUE_THINK` and `ANSWER` are grouped as non-retrieval actions.

Because the same frozen proxy supplies Phase II supervision, the results below diagnose agreement with the intended RHRD-aligned behavior. They do not treat the proxy labels as ground-truth retrieval necessity and do not independently establish downstream utility or optimal retrieval trigger timing. Those questions are evaluated separately through episode-level outcomes, sentence-level NHR, and the controlled timing interventions in Section 4.5.

G.1 Dataset-level and Macro RHRD Alignment

Table 22 reports sample-macro Precision, Recall, false positive rate (FPR), and F2 under the frozen proxy criterion. As defined in Appendix D.4, each metric is first computed within a sample and then averaged within each dataset. TruthfulQA is included only in this diagnostic. Because the six datasets differ in question type, trajectory length, and number of sentence-end decisions, Table 23 further averages the dataset values without weighting so that no dataset dominates the aggregate result.

Table 21: End-to-end training cost on 8 NVIDIA H200 140GB GPUs. GPU-hours are wall-clock hours multiplied by eight. The relative rows compare the complete two-phase Grounded-R1 path with the single-stage Search-R1 reproduction under the same optimizer and hardware.

Training path	PPO wall-clock / H200 GPU-hours	GRPO wall-clock / H200 GPU-hours
Search-R1 reproduction	83.3 h / 666.7	55.0 h / 440.0
Grounded-R1 Phase I	66.7 h / 533.3	50.0 h / 400.0
Grounded-R1 Phase II	70.0 h / 560.0	53.3 h / 426.7
Grounded-R1 total	136.7 h / 1,093.3	103.3 h / 826.7
Relative to Search-R1	1.64×	1.88×

Table 22: Dataset-wise sample-macro alignment between sentence-end retrieval actions and frozen RHRD proxy labels. Each metric is computed per sample and then averaged within the dataset. P denotes Precision, R denotes Recall, and FPR denotes false positive rate.

Method	Bamboogle				HaluEval				HotpotQA			
	FPR	P	R	F2	FPR	P	R	F2	FPR	P	R	F2
Prompted Reasoning-Retrieval Qwen	0.6680	0.1160	0.1680	0.1489	0.7380	0.2047	0.3113	0.2752	0.7305	0.2194	0.3387	0.2985
Search-R1	0.2407	0.1250	0.1296	0.1260	0.2211	0.1823	0.2013	0.1936	0.2127	0.2194	0.2383	0.2300
R1-Searcher	0.5280	0.0840	0.1000	0.0938	0.5639	0.1285	0.1733	0.1576	0.5703	0.1466	0.1998	0.1809
Grounded-R1 (episode-EM)	0.1950	0.2700	0.2900	0.2833	0.3415	0.2031	0.2386	0.2262	0.3642	0.2245	0.2634	0.2504
Grounded-R1 (episode-semantic)	0.1610	0.0424	0.0424	0.0424	0.1581	0.0599	0.0656	0.0637	0.1358	0.0442	0.0494	0.0477
Grounded-R1 (full)	0.0565	0.3427	0.3024	0.3060	0.1554	0.3414	0.3661	0.3576	0.0599	0.4122	0.3729	0.3743

Method	MuSiQue				PopQA				TruthfulQA			
	FPR	P	R	F2	FPR	P	R	F2	FPR	P	R	F2
Prompted Reasoning-Retrieval Qwen	0.7305	0.1373	0.2044	0.1802	0.7489	0.2335	0.3594	0.3172	0.8585	0.0830	0.1513	0.1283
Search-R1	0.2945	0.1984	0.2219	0.2106	0.1385	0.1592	0.1728	0.1681	0.4094	0.2047	0.2621	0.2419
R1-Searcher	0.5386	0.1321	0.1799	0.1633	0.5534	0.1742	0.2219	0.2047	0.7199	0.0913	0.1398	0.1232
Grounded-R1 (episode-EM)	0.4444	0.2188	0.2743	0.2550	0.3538	0.1697	0.1968	0.1873	0.5744	0.2397	0.3354	0.3035
Grounded-R1 (episode-semantic)	0.2859	0.0686	0.0769	0.0742	0.7979	0.0366	0.0418	0.0401	0.2661	0.0769	0.0918	0.0868
Grounded-R1 (full)	0.0954	0.4459	0.3946	0.3968	0.1969	0.3500	0.3688	0.3625	0.0882	0.5585	0.5067	0.5068

Under the frozen proxy criterion, Grounded-R1 (full) attains the highest F2 on all six datasets and the best macro Precision, Recall, and F2, together with the lowest macro FPR. Relative to Search-R1, its macro F2 increases from 0.1950 to 0.3840, while macro FPR decreases from 0.2528 to 0.1087. Relative to Grounded-R1 (episode-EM), macro F2 increases from 0.2510 to 0.3840 and macro FPR decreases from 0.3789 to 0.1087.

The gains reflect both greater coverage of proxy-labeled high-risk states and fewer retrieval actions at proxy-labeled low-risk states. They are therefore not explained by uniformly increasing the propensity to retrieve. This interpretation is also consistent with Table 2, where Phase II changes the macro average number of retrieval calls only from 2.00 to 2.04. The diagnostic supports the narrower conclusion that Phase II learns the intended RHRD-aligned behavior while preserving retrieval restraint.

G.2 Confusion-Matrix Diagnostics

Figure 6 provides a complementary decision-weighted view by pooling all valid sentence-end decision points within each model-dataset pair before normalization. Unlike the sample-macro metrics in Tables 22 and 23, every decision point receives equal weight, so samples with more sentence-end decisions contribute more to the plotted proportions. TP, FN, FP, and TN measure agreement between policy actions and sentence-end proxy labels rather than final-answer correctness. Since each policy generates its own trajectories, the normalized areas describe the decision states encountered by that policy and should not be interpreted as a shared underlying risk distribution.

The matrices expose distinct proxy-alignment error modes. Search-R1 assigns non-retrieval actions to a larger share of proxy-positive states on several datasets, whereas R1-Searcher places substantial mass in the FP region. Grounded-R1 (full) generally shifts mass toward TP while controlling FP. On HaluEval, its TP mass is 26.9% compared with 13.3% for Search-R1, while its FP mass is 9.3% compared with 15.1%. On PopQA, its TP mass is 30.6% compared with 10.4%, and its FN mass is 15.6% compared with 24.4%. For example, the latter pooled proportions imply a decision-micro Recall of $30.6/(30.6 + 15.6) \approx 66.2\%$, which is not expected to equal the PopQA sample-macro

Table 23: Global sample-macro alignment between sentence-end retrieval actions and frozen RHRD proxy labels. The dataset-level sample-macro values in Table 22 are averaged without weighting over datasets.

Method	Macro Precision	Macro Recall	Macro F2	Macro FPR
Prompted Reasoning-Retrieval Qwen	0.1656	0.2555	0.2247	0.7457
Search-R1	0.1815	0.2043	0.1950	0.2528
R1-Searcher	0.1261	0.1691	0.1539	0.5790
Grounded-R1 (episode-EM)	0.2210	0.2664	0.2510	0.3789
Grounded-R1 (episode-semantic)	0.0548	0.0613	0.0592	0.3008
Grounded-R1 (full)	**0.4084**	**0.3853**	**0.3840**	**0.1087**

Recall of 0.3688 in Table 22. The figure therefore complements rather than summarizes the sample-macro tables.

G.3 Class-conditional Proxy Errors

Within the Phase II objective, a false negative occurs when the frozen RHRD proxy labels a sentence-end state as `should-retrieve` but the policy continues generation or answers without retrieval. A false positive occurs when the policy retrieves at a proxy-labeled `should-not-retrieve` state. The reward uses $\alpha > \gamma$ in Eq. 23, placing greater weight on reducing false negatives while retaining a penalty for unnecessary retrieval.

Because Figure 6 pools decision points before normalization, we derive the corresponding decision-micro false negative rate from its pooled counts:

$$\mathrm{FNR}_{\text{dec-micro}} = \frac{\mathrm{FN}_{\text{pool}}}{\mathrm{TP}_{\text{pool}} + \mathrm{FN}_{\text{pool}}}. \tag{53}$$

On PopQA, Grounded-R1 (full) reaches 30.6% TP, compared with 10.4% for Search-R1 and 12.1% for R1-Searcher. Its FN mass is 15.6%, lower than the 24.4% of Search-R1. This yields a decision-micro FNR of $15.6/(30.6 + 15.6) \approx 33.8\%$ for Grounded-R1 (full), compared with $24.4/(10.4 + 24.4) \approx 70.1\%$ for Search-R1. Grounded-R1 (full) therefore assigns retrieval to a larger fraction of pooled PopQA decision points labeled `should-retrieve` by the frozen proxy.

HaluEval requires the same pooled class-conditional comparison because Grounded-R1 (full) has a slightly higher absolute FN mass than Search-R1, 39.6% versus 38.5%, but also a larger TP mass, 26.9% versus 13.3%. The corresponding decision-micro FNR is $39.6/(26.9 + 39.6) \approx 59.5\%$ for Grounded-R1 (full) and $38.5/(13.3 + 38.5) \approx 74.3\%$ for Search-R1. Thus, among the pooled states labeled `should-retrieve` on each policy's trajectories, Grounded-R1 (full) assigns a non-retrieval action to a smaller fraction. These decision-micro FNR values are not expected to equal one minus the sample-macro Recall values reported in Table 22.

Together, the sample-macro and pooled decision-micro diagnostics show that Phase II learns the intended association between RHRD-identified risk and sentence-end actions without collapsing into indiscriminate retrieval. This is evidence of successful proxy alignment rather than independent evidence that every proxy-positive state requires retrieval. The effect on generated content is evaluated separately by the sentence-level hallucination analysis in Appendix H, and the timing value of retrieval is tested through the proxy-label-independent interventions in Appendices I and J.

H Sentence-level Hallucination Evaluation

H.1 Sentence-level Hallucination Annotation Protocol

We evaluate local hallucination at sentence-level granularity. Each generated output is first segmented into sentences with spaCy's `en_core_web_sm` pipeline, and each sentence is treated as an annotation unit. The factuality prompt further asks the judge to separate factual assertions from abstention or uncertainty phrases. This keeps the evaluation close to claim-level factual consistency while avoiding an additional manual decomposition stage.



Figure 6: Pooled decision-level confusion-state distributions relative to frozen RHRD proxy labels across six datasets. Each panel aggregates all valid sentence-end decision points for one model-dataset pair and normalizes TP, FN, FP, and TN to sum to 100%. The x-axis denotes policy actions, and the y-axis denotes proxy reference labels. `Search` corresponds to `SEARCH`, while `Continue` groups `CONTINUE_THINK` and `ANSWER`.

For each sentence, we obtain two binary labels. The factuality hallucination label records whether the sentence contains a factual assertion that is contradicted by, or remains unsupported under, the external-evidence annotation protocol. The faithfulness hallucination label records whether the sentence conflicts with the original instruction, retrieved evidence, given context, conversation history, or structural constraints. These two labels are intentionally separated. Factuality hallucination

concerns consistency with verifiable real-world facts, while faithfulness hallucination concerns consistency with the source reference set available to the model.

For a single generated output, let $h_i^{\text{fac}} \in \{0, 1\}$ and $h_i^{\text{faith}} \in \{0, 1\}$ denote factuality and faithfulness hallucination indicators for sentence i, where 1 indicates the presence of the corresponding hallucination type. A sentence is counted as non-hallucinated only when both indicators are zero:

$$n_i = (1 - h_i^{\text{fac}})(1 - h_i^{\text{faith}}), \qquad \text{NHR} = \frac{1}{N} \sum_{i=1}^{N} n_i. \tag{54}$$

Here, n_i is the sentence-level non-hallucination indicator, N is the number of evaluated sentences in that output, and NHR denotes the output-level non-hallucination rate. Eq. 54 therefore defines NHR within a single output rather than by pooling sentences across outputs. Dataset-level and global results average these output-level values first across samples within each dataset and then across datasets, following Appendix D.4. Eq. 54 defines NHR on the $[0, 1]$ scale. In Tables 24 and 25, we report $100 \times \text{NHR}$ as a percentage.

For factuality annotation, a sentence is labeled as factuality hallucination if any factual assertion is contradicted by external evidence or remains unsupported under the decision rules in Appendix H.3. Pure abstention is not treated as factuality hallucination when no factual assertion is present. If external evidence is unavailable, the judge may use internal and general knowledge, but unverifiable factual assertions are labeled as not supported when high-confidence verification is not possible. For partially supported sentences, unsupported or contradicted factual assertions determine the label.

For faithfulness annotation, the judge uses the original instruction, available source context, retrieved evidence, local generation context, the sentence list, and the original full output. A sentence is labeled as faithfulness hallucination if it contradicts the instruction or source context, adds unsupported content relative to the available evidence, violates explicit constraints, or creates a logic or structural inconsistency in the same output. The original full output is used to validate global tag structure and is not treated as an additional evidence source.

For the causal retrieval trigger timing experiments, faithfulness is evaluated using sentence-specific temporally available context. For each target sentence, all retrieval blocks and generated content occurring after that sentence are redacted from every judge input field, including `Original SAME output`. The source context, retrieved evidence, and local generation context therefore contain only information available before that sentence was generated. To retain global tag validation without exposing future content as evidence, the evaluator receives a structure-preserving masked version of the full output in which post-target content is replaced by placeholders while tag boundaries are retained. The unredacted branch is used only by a separate deterministic tag-structure parser and is never exposed as evidentiary context to the faithfulness judge.

When automatic annotation fails because evidence cannot be parsed, the context is missing, the sentence contains no clear claim, or the judge output violates the required format, the item is flagged as an automatic annotation failure rather than silently used as a normal label. These failure cases are inspected in the human sanity check when they appear in the sampled subset described in Appendix H.7.

H.2 Sentence-level Factuality and Faithfulness Judge Prompts

We use Qwen2.5-72B-Instruct as the LLM-as-a-judge model for both factuality and faithfulness annotation. Final-answer judge prompts are reported in Appendix D; here we only report the sentence-level hallucination prompts.

Factuality annotation prompt.

```
Instructions:
You are a fact-consistency judge. You receive a STATEMENT and some
KNOWLEDGE.

0) Determine KNOWLEDGE_AVAILABILITY:
    - If the KNOWLEDGE text is empty or equals, case-insensitively, any
    of
```

```
      "no result", "n/a", "none", "not found", or similar, treat it as
      NO_EXTERNAL_KNOWLEDGE.

1) Reason step-by-step. First, briefly summarize the KNOWLEDGE, or state
   "No external knowledge" if NO_EXTERNAL_KNOWLEDGE.

   Extract claims:
    - Split the STATEMENT into factual assertions vs. abstention or
      uncertainty phrases.
    - PURE_ABSTENTION = the STATEMENT contains only uncertainty or
      abstention and no factual assertions.

2) Decision rules:
    A) If NO_EXTERNAL_KNOWLEDGE:
        - If PURE_ABSTENTION, label as "{SUPPORTED_LABEL}".
        - If there exists any factual assertion, judge that assertion
        using
          internal or general knowledge.
        - If you cannot verify it with high confidence, label
          "{NOT_SUPPORTED_LABEL}".
        - Hedging or uncertainty does not convert an unsupported
        assertion
          into supported.
    B) If external KNOWLEDGE is present:
        - Label "{SUPPORTED_LABEL}" if the factual assertions are
        directly
          supported or strongly implied.
        - Label "{NOT_SUPPORTED_LABEL}" if any factual assertion
        contradicts
          the KNOWLEDGE or is not supported by it.
        - Hedging or uncertainty phrases do not override unsupported or
          contradicted assertions.

3) Evidence:
    - If external KNOWLEDGE exists, list key supporting snippets as
    bullets.
    - If NO_EXTERNAL_KNOWLEDGE, write:
      "No external evidence (knowledge retrieval returned no result)."

4) After the reasoning, restate the STATEMENT and then give the final
decision.

5) The final decision must be exactly "{SUPPORTED_LABEL}" or
   "{NOT_SUPPORTED_LABEL}", wrapped in square brackets.

KNOWLEDGE:
{_KNOWLEDGE_PLACEHOLDER}

STATEMENT:
{_STATEMENT_PLACEHOLDER}
```

Faithfulness annotation prompt.

```
You are a strict faithfulness judge. Using ONLY the given Original
Instruction,
Available Source Context, Retrieved Evidence, Local Generation Context,
and
the list of output sentences, label EACH sentence as:
- 0 = faithful (no faithfulness hallucination)
- 1 = faithfulness hallucination

Definitions to apply. Do not use outside knowledge.

Temporal evidence rule:
- For each target sentence, use only source context, retrieved evidence,
and
```

```
  local generation context available before that sentence was generated.
- Never use a later retrieval block or later generated content to
support,
  contradict, or otherwise relabel an earlier sentence.
- Original SAME output is provided only for global tag-structure
validation.
  Content appearing only in Original SAME output is not an evidence
  source.

1) Source/Context inconsistency:
The sentence contradicts the instruction or question, adds unsupported
claims
not grounded in the available source context or retrieved evidence, or
contradicts other provided context.

2) Instruction/Constraint inconsistency:
The sentence violates explicit constraints in the instruction, including
topic,
role, required format, required tags, count, JSON/XML format, language,
or
prohibited content.

3) Logic/Reasoning inconsistency:
The sentence contradicts other sentences in the same output or the
required
structure. This includes conflicts between <think>...</think> and
<answer>...</answer>, missing or duplicated tags, and final answers that
contradict prior reasoning.

Crucial global tag-pairing rule:
- Always analyze tag structure on the full Original SAME output first,
not
  per sentence.
- If the full output contains a correctly paired tag block, treat that
block
  as well-formed even if the opener and closer are split across adjacent
  sentences.
- Tag-only sentences that participate in a correct pair must be labeled
0.
- If a tag error truly exists in the full output, assign 1 only to the
  sentence containing the actual error token.

Tag vocabulary restriction:
The only valid lowercase tag pairs are:
<search>...</search>
<information>...</information>
<think>...</think>
<answer>...</answer>

No other tag names are allowed. Bracketed variants such as [search],
[/search], [answer], or [/answer] are errors when format is required.

Judgment order:
A) Use Original SAME output to determine whether tags are required and
whether
   the global tag structure is well-formed.
B) If global structure is well-formed, label tag-only sentences in
correct
   pairs as 0 and evaluate content-bearing sentences for faithfulness.
C) If a true structural error exists, assign 1 only to the sentence
containing
   the error token.

Output format:
Return ONLY a sequence of bracketed digits, one per sentence, with no
spaces
```

```
or extra text.
Correct example: [0][1][0]

Original Instruction:
<<<
{instruction}
>>>

Available Source Context:
<<<
{source_context}
>>>

Retrieved Evidence:
<<<
{retrieved_evidence}
>>>

Local Generation Context:
<<<
{local_context}
>>>

Sentences:
{sent_block}

Original SAME output:
{original_output}

Now output the labels for sentences 1..{n_sent} as bracketed digits ONLY.
```

H.3 External Evidence Channels for Factuality Annotation

For factuality annotation, each sentence is checked against five external evidence channels: Google Search[7], Google News[8], Google Scholar[9], Wikipedia[10], and DuckDuckGo[11]. Each channel returns an evidence set for the target sentence. The LLM-as-a-judge model then decides whether the sentence is supported or not supported under that channel.

The channel-level decisions are aggregated by majority voting. If not-supported decisions form the majority, the sentence is labeled as factuality hallucination. Otherwise, it passes factuality verification. These external evidence channels are used for sentence-level annotation, including the validation annotations used to calibrate τ; they are not exposed to the policy or used as direct reward inputs during policy optimization.

If a channel returns no usable result, the factuality prompt applies the no-external-knowledge branch. Pure abstention is treated as supported, while factual assertions must still be verified using internal or general knowledge. If high-confidence verification is not possible, the assertion is labeled as not supported. This rule makes the factuality annotation conservative for unsupported confident claims.

H.4 Evidence and Context Sources for Faithfulness Annotation

Faithfulness annotation uses source-side information rather than open-world evidence. The judge receives the original instruction, available source context, retrieved evidence, local generation context, the output sentences, and the original full output. Retrieved evidence, given context, prior generated content, and explicit task constraints are treated as the source reference set when available.

The faithfulness label captures evidence faithfulness rather than factual consistency. A sentence may be factually true in the world but still unfaithful if it is not supported by the provided context

[7] https://serpapi.com/
[8] https://serpapi.com/
[9] https://serpapi.com/
[10] https://pypi.org/project/wikipedia/
[11] https://pypi.org/project/duckduckgo-search/

Table 24: Full sentence-level non-hallucination results. Fac., Faith., and Overall denote factuality NHR, faithfulness NHR, and overall NHR, respectively. NHR denotes non-hallucination rate.

Method	ASQA			Bamboogle			HaluEval		
	Fac.	Faith.	Overall	Fac.	Faith.	Overall	Fac.	Faith.	Overall
Prompted Reasoning-Retrieval Qwen	61.18	81.13	55.62	68.29	88.46	65.96	50.73	87.31	47.41
Search-R1	56.14	77.50	48.40	55.65	72.39	40.94	39.50	71.04	29.46
R1-Searcher	51.27	65.53	41.34	60.69	69.49	48.41	36.22	71.24	31.50
Self-RAG	66.83	80.49	60.10	49.04	78.77	44.65	50.69	65.48	41.12
Search-o1	62.18	86.05	58.22	65.49	**90.18**	62.68	51.43	87.99	47.00
IRCoT	58.06	86.34	54.17	69.79	86.61	65.44	53.62	85.10	49.34
ETC	68.50	75.47	54.84	**72.41**	78.96	59.92	54.33	73.87	42.48
Grounded-R1 (full)	**81.79**	**93.86**	**79.63**	70.19	88.16	**67.02**	**56.11**	**91.42**	**52.96**

Method	PopQA			TriviaQA			Macro Avg.		
	Fac.	Faith.	Overall	Fac.	Faith.	Overall	Fac.	Faith.	Overall
Prompted Reasoning-Retrieval Qwen	60.53	83.97	52.76	72.02	82.10	65.05	62.55	84.59	57.36
Search-R1	46.58	75.72	37.38	56.93	72.09	46.32	50.96	73.75	40.50
R1-Searcher	41.74	72.34	33.99	59.16	69.10	48.67	49.82	69.54	40.78
Self-RAG	67.92	82.31	60.83	72.16	77.27	64.95	61.33	76.86	54.33
Search-o1	58.59	90.12	53.68	67.73	81.90	61.33	61.08	87.25	56.58
IRCoT	50.03	88.86	46.51	67.78	85.31	61.02	59.86	86.44	55.30
ETC	60.56	69.64	46.05	73.38	74.47	59.43	65.84	74.48	52.54
Grounded-R1 (full)	**75.45**	**95.38**	**73.52**	**77.23**	**90.57**	**73.44**	**72.15**	**91.88**	**69.31**

or retrieved evidence. Conversely, a sentence may be faithful to the given evidence while still requiring factuality verification against external evidence. This separation motivates the factuality and faithfulness NHR breakdowns reported in this appendix, while Section 4.4 reports overall NHR.

H.5 Full Sentence-level Non-hallucination Results

Table 24 reports the full sentence-level hallucination results on five open-ended benchmarks. All values are percentages. The macro average is computed as an unweighted average over ASQA, Bamboogle, HaluEval, PopQA, and TriviaQA.

Grounded-R1 (full) achieves the highest overall NHR on all five benchmarks and the best macro factuality, faithfulness, and overall NHR. Its macro overall NHR is 69.31%, exceeding the strongest external baseline, Search-o1, by 12.73 percentage points. These results establish the sentence-level groundedness improvement under open-ended generation. The downstream value of timely retrieval and its allocation across sentence-end positions are evaluated separately by the controlled timing interventions in Appendices I and J.

The gains are largest on ASQA and PopQA. On ASQA, Grounded-R1 (full) reaches 79.63% overall NHR, improving over Self-RAG by 19.53 points. This indicates stronger reliability in long-form answer generation, where local factual drift is more likely to accumulate. On PopQA, Grounded-R1 (full) reaches 73.52% overall NHR. This result is consistent with the intended role of timely retrieval in reducing unsupported factual claims in long-tail entity questions.

HaluEval provides a more adversarial setting. Grounded-R1 (full) obtains 52.96% overall NHR, higher than IRCoT and Search-o1. It also achieves the highest factuality and faithfulness NHR on this benchmark, indicating that the model reduces both unsupported factual content and source-side inconsistency.

Bamboogle shows a more balanced pattern. Grounded-R1 (full) does not obtain the best factuality or faithfulness NHR separately, but it achieves the best overall NHR. Since overall NHR requires both hallucination types to be absent, this result suggests that Grounded-R1 (full) improves the joint reliability of factual consistency and evidence faithfulness under difficult compositional reasoning.

H.6 Sentence-level Hallucination Ablation Results

Table 25 reports the sentence-level hallucination ablation over Grounded-R1 variants. The three variants follow the same naming convention as the main paper. Grounded-R1 (episode-EM) uses terminal EM, format reward, and retrieval-cost penalties. Grounded-R1 (episode-semantic) replaces terminal EM with a continuous LLM-as-a-judge semantic reward. Grounded-R1 (full) further adds the Phase II sentence-level reward for retrieval trigger timing.

Table 25: Sentence-level hallucination ablation results for Grounded-R1 variants. Fac., Faith., and Overall denote factuality NHR, faithfulness NHR, and overall NHR, respectively.

Variant	ASQA			Bamboogle			HaluEval		
	Fac.	Faith.	Overall	Fac.	Faith.	Overall	Fac.	Faith.	Overall
Grounded-R1 (episode-EM)	61.40	73.08	53.11	67.28	87.25	63.43	**56.27**	90.67	52.18
Grounded-R1 (episode-semantic)	70.53	**95.04**	68.81	64.18	82.19	57.59	47.74	91.21	46.27
Grounded-R1 (full)	**81.79**	93.86	**79.63**	**70.19**	**88.16**	**67.02**	56.11	**91.42**	**52.96**

Variant	PopQA			TriviaQA			Macro Avg.		
	Fac.	Faith.	Overall	Fac.	Faith.	Overall	Fac.	Faith.	Overall
Grounded-R1 (episode-EM)	44.25	72.00	38.47	73.07	75.71	62.85	60.45	79.74	54.01
Grounded-R1 (episode-semantic)	62.93	89.76	58.80	72.80	90.04	69.53	63.64	89.65	60.20
Grounded-R1 (full)	**75.45**	**95.38**	**73.52**	**77.23**	**90.57**	**73.44**	**72.15**	**91.88**	**69.31**

The overall NHR increases from 54.01% for Grounded-R1 (episode-EM) to 60.20% for Grounded-R1 (episode-semantic), and further to 69.31% for Grounded-R1 (full). This progression shows that terminal EM alone is too coarse to constrain local generation. The semantic reward improves sentence-level reliability, and Grounded-R1 (full), which further adds Phase II, achieves the strongest sentence-level groundedness. The controlled timing experiments separately show that earlier delivery of the same evidence cuts next-sentence hallucination by 11.1 points and that, under equal retrieval budgets, Grounded-R1 targets positions with an observed 8.3-point downstream hallucination reduction.

The main gain from Grounded-R1 (episode-semantic) appears in faithfulness NHR. Its macro faithfulness NHR rises from 79.74% to 89.65%, suggesting that continuous semantic reward helps reduce inconsistency with given context and retrieved evidence. Grounded-R1 (full) brings larger gains in factuality NHR and overall NHR. Compared with Grounded-R1 (episode-semantic), it improves macro factuality NHR from 63.64% to 72.15% and macro overall NHR from 60.20% to 69.31%.

The non-monotonic patterns on Bamboogle and HaluEval are also informative. Grounded-R1 (episode-EM) can outperform Grounded-R1 (episode-semantic) on some individual metrics, but Grounded-R1 (full) still obtains the best overall NHR. This suggests that terminal semantic reward does not uniformly improve every local hallucination dimension, especially in difficult compositional or hallucination-heavy settings. Phase II reduces this mismatch by directly optimizing sentence-end retrieval trigger timing with a sentence-level reward.

H.7 Human Sanity Check

We conduct a small human sanity check to validate the automatic sentence-level hallucination annotation protocol. This check is not intended to replace the automatic evaluation or construct a full human benchmark. It asks two narrower questions: first, whether human annotators can reach stable judgments on factuality hallucination and faithfulness hallucination; and second, whether automatic labels are broadly aligned with final human labels after adjudication.

Sampling protocol. We sample 100 claim-bearing sentences from the open-ended generation results used in Section 4.4. The sample covers ASQA, Bamboogle, HaluEval, PopQA, and TriviaQA. It also covers major compared model families, including Search-R1, R1-Searcher, and Grounded-R1 variants. The sample includes sentences from different RHRD score ranges, automatic hallucination-negative and hallucination-positive cases, and sentences before and after retrieved evidence is inserted. Each item retains the original input, available context or retrieved evidence, target sentence, and model source. The model source is hidden from annotators.

Annotation procedure. Two annotators independently label every sentence along two binary dimensions. The factuality label asks whether the target sentence contains a verifiable claim that is contradicted by or insufficiently supported by real-world evidence. The faithfulness label asks whether the target sentence is faithful to the given input, retrieved evidence, local context, prior generation, and explicit task constraints. The two dimensions are independent, and a sentence may be positive on both.

Before formal annotation, the two annotators label 10 pilot samples to align the decision standard. Pilot samples are not included in the reported statistics. After independent annotation, disagreements

Table 26: Human sanity check results. Human-human and tool-human columns report agreement in percentages. FNR and FPR are also percentages.

Dimension	Human-human	Tool-human	Precision	Recall	F1	FNR	FPR
Factuality	96.0	88.0	0.733	1.000	0.846	0.0	17.9
Faithfulness	95.0	91.0	0.650	0.867	0.743	13.3	8.2

Table 27: Tool-human confusion matrices for the human sanity check. The positive class is the presence of the corresponding hallucination type.

Dimension	TP	FP	FN	TN
Factuality	33	12	0	55
Faithfulness	13	7	2	78

are reviewed by a third adjudicator. If the two annotators agree, their shared label is used as the final human label. Otherwise, the adjudicator label is used.

Tool-human comparison metrics. We treat the hallucination-positive label as the positive class. For each dimension, TP means that both the automatic tool and final human label identify a hallucination. FP means that the tool identifies a hallucination but the final human label does not. FN means that the final human label identifies a hallucination but the tool misses it. TN means that both labels identify no hallucination. We compute:

$$\text{Agreement} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}, \quad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{55}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad \text{FNR} = \frac{\text{FN}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{56}$$

Tables 26 and 27 show high human-human agreement, with 96.0% on factuality and 95.0% on faithfulness. This suggests that the two annotation dimensions are operationally separable and can be judged with stable criteria on the sampled sentences.

The automatic tool also aligns reasonably well with final human labels. On factuality, tool-human agreement is 88.0%, precision is 0.733, recall is 1.000, and F1 is 0.846. The factuality FN count is zero, meaning that all human-confirmed factuality hallucinations in this sample are captured by the automatic annotation. The main factuality error type is false positive. This conservative behavior may slightly underestimate NHR, but it reduces the risk of systematically missing factuality hallucination.

On faithfulness, tool-human agreement is 91.0%, precision is 0.650, recall is 0.867, and F1 is 0.743. Faithfulness is harder than factuality because it depends more strongly on evidence coverage, omitted context, implicit reasoning, and constraint interpretation. The tool produces seven false positives and two false negatives. False positives usually reflect indirect support or semantically equivalent wording, while false negatives may arise when a sentence appears locally compatible but introduces unsupported information or violates a broader contextual constraint.

Overall, the human sanity check supports the use of the automatic sentence-level hallucination annotation protocol for relative model comparison. It also supports the factuality and faithfulness NHR breakdowns reported in this appendix. The two dimensions capture different failure modes, while Section 4.4 reports overall NHR as the stricter measure of sentence-level grounded generation.

I Matched-Prefix Intervention on Retrieval Trigger Timing

This appendix provides the full protocol and results for the matched-prefix intervention in Section 4.5. The experiment asks whether retrieval is useful at the sentence-end positions selected by Grounded-R1 and whether the same fixed retrieval block is more useful before rather than after the next sentence is generated.

I.1 Experimental Design and Estimands

For branch point i, let P_i denote the common generation prefix immediately before the original SEARCH action, q_i the original retrieval query, and E_i the fixed, ordered top-3 retrieved evidence. Let

$$R_i = \text{serialize}(q_i, E_i) \tag{57}$$

denote the fixed retrieval block produced by the original serialization template, and let s_i denote the first complete downstream sentence generated without access to R_i. Table 28 defines the three paired conditions.

Table 28: Matched-prefix retrieval-block insertion schedules. Immediate and Delayed receive the same fixed retrieval block exactly once. Delayed and No retrieval share the same first downstream sentence.

Condition	Before the first downstream sentence	First downstream sentence	After the sentence	Remaining generation	Prescribed retrievals
Immediate retrieval	Insert R_i	Generate from $P_i + R_i$	No insertion	Disable additional SEARCH	1
Delayed retrieval	No insertion	Reuse the shared s_i	Insert the same R_i	Disable additional SEARCH	1
No retrieval	No insertion	Reuse the shared s_i	No insertion	Disable additional SEARCH	0

For an outcome Y, the primary timing estimand is

$$\Delta_{\text{timing}} = \mathbb{E}[Y_i(\text{Immediate}) - Y_i(\text{Delayed})]\,. \tag{58}$$

Immediate and Delayed use exactly the same R_i and one prescribed retrieval, so this contrast isolates retrieval-block insertion timing relative to s_i. Because q_i and E_i move together across conditions, this contrast should not be interpreted as the isolated causal effect of evidence tokens alone; any prompting or priming effect of q_i is part of the block-level treatment. We also report

$$\Delta_{\text{retrieval}} = \mathbb{E}[Y_i(\text{Immediate}) - Y_i(\text{No retrieval})]\,, \tag{59}$$

which measures the value of executing the original retrieval interaction rather than skipping it. Both estimands are conditional on sentence-end positions at which Grounded-R1 actually selects SEARCH; they do not describe the average effect over all possible sentence-end positions.

I.2 Datasets and Branch-Point Selection

We use the same 7B GRPO Grounded-R1 (full) checkpoint reported in the main experiments. The intervention covers ASQA, Bamboogle, HaluEval, PopQA, and TriviaQA, spanning long-form question answering, long-tail knowledge, and hallucination-intensive settings. We first run the policy in the unmodified retrieval environment and record its complete trajectories, including the generated token sequence, retrieval history, SEARCH positions, queries, passage identifiers, retrieved text, passage ordering, and final answer.

Eligible branch points are positions at which the policy emits SEARCH after a complete sentence. We retain at most one point per question and then randomly sample 5,000 points without replacement from the resulting pool. Sampling uses only the pre-intervention trajectory and never uses outcomes from any of the three continuations. Table 29 reports the sample composition and retrieval history.

Table 29: Branch-point sampling audit. Prior retrieval calls are counted before the selected sentence-end position. Each question contributes at most one branch point.

Dataset	Eligible questions	Branch points	Sample share (%)	Prior retrieval calls 0 / 1 / 2+	Median prefix sentences
ASQA	492	477	9.5	192 / 171 / 114	6
Bamboogle	123	123	2.5	76 / 34 / 13	4
HaluEval	1,640	802	16.0	465 / 242 / 95	4
PopQA	4,300	1,799	36.0	1,052 / 538 / 209	3
TriviaQA	4,250	1,799	36.0	1,048 / 542 / 209	3
Total	**10,805**	**5,000**	**100.0**	**2,833 / 1,527 / 640**	**3**

I.3 Matched Prefix, Fixed Retrieval Block, and Branch Construction

The common prefix P_i is stored as the exact input-token sequence received by the model immediately before the original SEARCH action. It contains the system instruction, question, all pre-fork reasoning, earlier retrieval calls and evidence, and the sentence that has just ended. Every condition continues from the same token IDs, without decoding the prefix to text and encoding it again.

The query q_i is the original query emitted at the selected SEARCH action, and E_i contains the original top-3 passage identifiers, passage text, and ordering. The fixed block R_i serializes q_i and E_i in the original retrieval format. Immediate and Delayed reuse this cached block instead of regenerating the query or rerunning the retriever. We verify that R_i is byte-identical across the two conditions, thereby holding the query, evidence relevance and content, passage ordering, serialization, and retrieval randomness fixed.

Immediate appends R_i to P_i and generates a new continuation. To construct the other two branches, we generate s_i once from P_i while masking SEARCH at the action layer. The same token sequence s_i is copied into Delayed and No retrieval. Delayed then appends R_i, whereas No retrieval does not. Generated content is never deleted, rewritten, or rolled back. Accordingly, throughout this work, preemptive intervention refers to preventing downstream propagation after a risky sentence-end state, rather than preventing or revising an error already emitted within the current sentence. From the fork until the final answer, all three conditions mask every additional SEARCH action. Immediate and Delayed therefore differ only in whether the same fixed retrieval block is inserted before or after s_i.

All branches use the same checkpoint, decoding parameters, stopping rule, generated-token budget, context window, and branch-point random seed. Retrieval-block tokens inserted by the intervention do not count toward the generated-token budget. The same context-handling rule is applied to both retrieval conditions, and the integrity checks reject any branch point that would produce differential prefix truncation.

I.4 Controls and Integrity Checks

Table 30 summarizes the factors held fixed. Retrieval is masked through decoding constraints rather than a natural-language instruction, so the intervention does not add condition-specific text to the prompt.

Table 30: Controls used in the matched-prefix intervention. The only manipulated factor between Immediate and Delayed is retrieval-block insertion timing.

Factor	Control
Model and state	One Grounded-R1 (full, 7B GRPO) checkpoint and identical pre-fork token IDs, including prior retrieval history
Branch point	The same sentence-end `SEARCH` position within each question
Retrieval block	Byte-identical R_i, including q_i, top-3 passages E_i, ordering, and serialization, for Immediate and Delayed
Retrieval schedule	One prescribed retrieval for Immediate and Delayed, none for No retrieval, and no additional post-fork retrieval
First downstream sentence	Token-identical s_i for Delayed and No retrieval
Decoding and budget	Identical decoding configuration, stopping rule, generated-token limit, context handling, and branch-point random seed
Evaluation	Identical sentence segmentation, answer extraction, judge checkpoint, judge prompts, and condition-blind scoring

Each retained branch point must satisfy automated checks for identical common-prefix token IDs; byte-identical Immediate and Delayed retrieval-block checksums; prescribed retrieval counts of one, one, and zero; zero additional post-fork retrievals; token-identical first downstream sentences for Delayed and No retrieval; identical hallucination labels for that shared sentence; exclusion of the sentence at the fork from suffix metrics; successful application of the common stopping rule; and no deletion based on observed counterfactual outcomes. These checks make the equality of the Delayed and No retrieval first-sentence results a design implication rather than a coincidence between independent rollouts.

All 5,000 sampled branch points passed these integrity checks and produced at least one complete downstream sentence in every condition; no sampled point was replaced or excluded after continuation outcomes were observed.

I.5 Metrics and Statistical Inference

For branch point i and condition c, let $m_{i,c}$ be the number of complete sentences generated after the fork, and let $n_{i,c,j} = 1$ when downstream sentence j contains neither factuality hallucination nor faithfulness hallucination under the protocol in Appendix H.1. We define

$$\mathrm{SuffixNHR}_{i,c} = \frac{1}{m_{i,c}} \sum_{j=1}^{m_{i,c}} n_{i,c,j}. \tag{60}$$

The sentence that ends at the fork and all earlier text are excluded. We first average within branch points, then within each dataset, and finally compute an unweighted mean over the five datasets. This prevents longer continuations or larger datasets from receiving greater weight.

The first downstream hallucination rate evaluates only $1 - n_{i,c,1}$. Delayed and No retrieval share the same first downstream sentence, so this metric must be identical for those two conditions. Final EM uses the answer extraction and scoring procedure from the main benchmark evaluation. ASQA F1 is reported separately because F1 is part of the paper's existing ASQA evaluation protocol.

Factuality and faithfulness judgments follow Appendix H. Factuality is evaluated using the same external fact-verification protocol as in the main sentence-level evaluation. Faithfulness is evaluated using a sentence-specific temporally censored record. For each target sentence, the source context, retrieved evidence, and local generation context contain only information available before that sentence was generated. Post-target retrieval blocks and generated content are masked in the structure-only copy of the full output while tag boundaries are retained. Consequently, when the shared first downstream sentence s_i is scored in Delayed, R_i, including E_i, is absent from every evidentiary field and cannot retroactively support s_i. Model identity and condition names are hidden from the evaluator.

All contrasts are computed within the same branch point. We use a dataset-stratified paired bootstrap that resamples branch points with replacement inside each dataset, recomputes the dataset means and five-dataset unweighted macro average, and takes the 2.5th and 97.5th percentiles of the paired

difference distribution as the 95% confidence interval. ASQA F1 is resampled within ASQA. Reported contrasts are computed before rounding the displayed absolute values.

I.6 Aggregate Results

Table 31: Aggregate matched-prefix results in percentages. The first three outcomes are unweighted macro averages across five datasets; ASQA F1 is computed on ASQA only.

Condition	Suffix NHR ↑	First downstream hallucination rate ↓	Final EM ↑	ASQA F1 ↑
No retrieval	52.0	54.8	48.0	67.7
Delayed retrieval	52.7	54.8	50.1	69.5
Immediate retrieval	**59.8**	**43.7**	**52.6**	**74.2**

Table 32: Paired improvements in percentage points with 95% confidence intervals. The hallucination column reports a reduction, so positive values indicate improvement. Immediate − Delayed is the primary timing contrast.

Paired contrast	Suffix NHR gain ↑	First-sentence hallucination reduction ↑	Final EM gain ↑	ASQA F1 gain ↑
Immediate − No retrieval	$+7.8$ [$+5.6$, $+9.8$]	$+11.1$ [$+8.2$, $+13.9$]	$+4.6$ [$+2.7$, $+6.8$]	$+6.5$ [$+3.1$, $+10.0$]
Immediate − Delayed	$+7.1$ [$+5.0$, $+9.1$]	$+11.1$ [$+8.2$, $+13.9$]	$+2.6$ [$+0.8$, $+4.4$]	$+4.7$ [$+1.5$, $+8.1$]

Immediate is best on all four outcomes. The primary timing contrast holds R_i and the prescribed retrieval count fixed, yet inserting the same retrieval block before the first downstream sentence improves suffix NHR by 7.1 percentage points, reduces first-sentence hallucination by 11.1 points, and improves final EM and ASQA F1 by 2.6 and 4.7 points. Every paired interval excludes zero. Delayed and No retrieval have the same first-sentence hallucination rate by construction, providing a direct integrity check for the shared-sentence protocol.

I.7 Dataset-Level Results

Table 33: Dataset-level matched-prefix results in percentages. ASQA F1 is not defined for the other four datasets under the paper's evaluation protocol.

Dataset	Condition	Suffix NHR ↑	First downstream hallucination rate ↓	Final EM ↑	ASQA F1 ↑
ASQA	Immediate retrieval	**66.7**	**35.4**	**65.0**	**74.2**
	No retrieval	56.2	51.8	55.3	67.7
	Delayed retrieval	57.4	51.8	58.9	69.5
Bamboogle	Immediate retrieval	**60.2**	**45.9**	**37.4**	–
	No retrieval	57.0	48.8	35.0	–
	Delayed retrieval	56.1	48.8	35.8	–
HaluEval	Immediate retrieval	**45.1**	**59.9**	**52.2**	–
	No retrieval	36.7	70.1	46.0	–
	Delayed retrieval	39.1	70.1	51.7	–
PopQA	Immediate retrieval	**57.3**	**46.0**	**45.8**	–
	No retrieval	48.3	58.9	42.2	–
	Delayed retrieval	47.6	58.9	41.5	–
TriviaQA	Immediate retrieval	**69.9**	**31.5**	**62.8**	–
	No retrieval	61.9	44.6	61.6	–
	Delayed retrieval	63.3	44.6	62.4	–

Table 34: Dataset-level Immediate − Delayed effects in percentage points. Positive values indicate improvement; the last column is defined only on ASQA.

Dataset	Suffix NHR	First-sentence hallucination reduction	Final EM	ASQA F1
ASQA	+9.3	+16.4	+6.1	+4.7
Bamboogle	+4.1	+2.9	+1.6	–
HaluEval	+6.0	+10.2	+0.5	–
PopQA	+9.7	+12.9	+4.3	–
TriviaQA	+6.6	+13.1	+0.4	–
Macro / ASQA	+7.1	+11.1	+2.6	+4.7

Immediate improves suffix NHR and final EM and reduces first-sentence hallucination relative to Delayed on every dataset. The suffix NHR gains are largest on ASQA and PopQA at 9.3 and 9.7 points. On HaluEval and TriviaQA, final EM changes by only 0.5 and 0.4 points, while first-sentence hallucination falls by 10.2 and 13.1 points. This contrast shows why final-answer metrics alone can miss local errors that timely evidence prevents during generation.

I.8 Factuality and Faithfulness Breakdown

Table 35: Factuality and faithfulness breakdown in percentages, macro-averaged across the five datasets. Suffix NHR requires both criteria to hold and is not their arithmetic mean.

Condition	Factuality NHR ↑	Faithfulness NHR ↑	Suffix NHR ↑
No retrieval	56.7	86.2	52.0
Delayed retrieval	57.7	86.4	52.7
Immediate retrieval	**63.9**	**88.4**	**59.8**

Relative to Delayed, Immediate improves factuality NHR by 6.2 percentage points and faithfulness NHR by 2.0 points. The larger factuality gain is consistent with earlier retrieval-block insertion supplying external evidence before a factual claim is produced. Faithfulness begins from a higher level and also improves, while the joint suffix NHR remains the stricter outcome.

I.9 Interpretation and Scope

The Immediate and Delayed conditions receive exactly the same fixed retrieval block R_i and one prescribed retrieval. Their difference therefore cannot be attributed to retrieval volume, query formulation, evidence quality, passage ordering, or serialization. It measures the timing of the retrieval interaction relative to the first downstream sentence. The results show that the same retrieval block has different value at different insertion times. Early insertion can prevent a local error, whereas delayed insertion can only support recovery after that error has entered the generated trajectory.

Delayed improves final EM over No retrieval by 2.0 percentage points descriptively, but improves suffix NHR by only 0.7 points. We do not make an inferential claim about this unreported contrast. The pattern is nevertheless consistent with delayed insertion of R_i helping the remaining generation while being unable to alter the shared first downstream sentence.

The intervention outcomes do not use RHRD-derived proxy labels. Branch points come from the policy's actual `SEARCH` actions, treatment assignment changes only the insertion time of the fixed retrieval block R_i, and outcomes are computed from generated sentences and final answers. RHRD does not determine the three conditions or enter the definitions of suffix NHR, first-sentence hallucination, final EM, or ASQA F1. This provides outcome evidence that is distinct from the proxy-agreement analysis in Appendix G.

Finally, the estimands are conditional on Grounded-R1-selected `SEARCH` positions. The experiment establishes the temporal utility of these selected positions, but it does not prove globally optimal retrieval timing or compare selector quality between Grounded-R1 and other policies at a common set of candidate positions.

J EQUAL-BUDGET RETRIEVAL BENEFIT EVALUATION

This appendix provides the full design, estimands, results, and integrity audit for Table 5. Rather than asking again whether timely insertion of a retrieval interaction is effective, the analysis builds on the position-level timing effect defined by the matched-prefix intervention and asks which positions different selectors prioritize. Specifically, we give Grounded-R1 (full), Search-R1, and Grounded-R1 (episode) identical retrieval budgets over a source-balanced shared pool of sentence-end states and compare the observed timing benefit of the positions selected by each policy.

Two levels of inference must be distinguished. At a fixed position, the paired Immediate–Delayed difference comes from a controlled intervention on when a fixed retrieval block is inserted and can therefore be interpreted as the timing effect at that position. Comparisons across selector-chosen sets are selector-conditioned diagnostics. They characterize where each policy allocates its budget among positions with measured timing effects, but neither the checkpoint nor Phase II training itself is randomized. We therefore do not interpret these comparisons as unconditional causal effects of the training stage.

J.1 RESEARCH QUESTION AND ESTIMAND

For candidate sentence-end position i, let P_i denote the common token-level prefix, q_i the prespecified retrieval query, and E_i the fixed, ordered top-3 retrieved evidence. Let

$$R_i = \text{serialize}(q_i, E_i) \tag{61}$$

denote the retrieval block produced by a fixed serialization template. The position-level retrieval timing benefit is

$$\Delta_{\text{time}}(i) = \text{SuffixNHR}_{i,\text{Immediate}} - \text{SuffixNHR}_{i,\text{Delayed}}. \tag{62}$$

Immediate and Delayed use exactly the same R_i and one prescribed retrieval. Their only planned difference is whether R_i enters the dynamic context before or after the first downstream sentence is generated. Hence, $\Delta_{\text{time}}(i) > 0$ means that moving the same retrieval block one sentence earlier improves sentence-level groundedness after the candidate position. Equivalently, because the suffix hallucination rate is one minus suffix NHR, a positive $\Delta_{\text{time}}(i)$ denotes an equal-magnitude reduction in downstream hallucination.

This estimand measures the timing value of a fixed retrieval block, not the total value of retrieving relative to no retrieval. Because q_i and E_i move together across conditions, it should not be interpreted as the isolated causal effect of evidence tokens alone; any prompting or priming effect of the query is part of the block-level timing treatment. The selectors determine only which candidate positions enter their respective analysis sets. They do not generate the queries, alter the retrieved evidence, or act as the continuation writer after branching.

J.2 SOURCE-BALANCED SHARED CANDIDATE POOL

We use unmodified natural trajectories from the three frozen models on ASQA, Bamboogle, HaluEval, PopQA, and TriviaQA. A candidate must occur in model-generated reasoning rather than inside an `<information>` block, immediately follow a completed sentence, precede answer termination, retain at least one retrieval call in the budget, and be readable through the common scoring interface of all three selectors. Eligibility does not require the source model to have emitted `SEARCH` at that position, avoiding evaluation of a selector only on its own native retrieval positions.

We retain at most one position per question. Sampling uses a fixed seed and is frozen before running any selector, generating intervention branches, or inspecting hallucination labels. It does not use selector scores, native triggers, Real-time Hallucination Risk Detection (RHRD) scores, post-hoc evidence quality, or timing outcomes. Grounded-R1 (full), Search-R1, and Grounded-R1 (episode) contribute 667, 667, and 666 positions, respectively, so no single model's state distribution dominates the shared pool.

Table 36: Shared candidate pool, prespecified budgets, and timing evaluability. Source counts are ordered as Grounded-R1 (full) / Search-R1 / Grounded-R1 (episode). N_d is the number of frozen shared candidate positions and B_d is the common dataset budget used by all selectors.

Dataset	Candidate questions	Source counts	N_d	B_d	Timing-evaluable	Coverage
ASQA	500	157 / 156 / 156	469	47	469	100.00%
Bamboogle	125	42 / 42 / 41	125	12	125	100.00%
HaluEval	1,996	156 / 157 / 156	469	47	469	100.00%
PopQA	707	156 / 156 / 157	469	47	469	100.00%
TriviaQA	497	156 / 156 / 156	468	47	468	100.00%
Total	**3,825**	**667 / 667 / 666**	**2,000**	**200**	**2,000**	**100.00%**

All 2,000 frozen candidate positions satisfy the timing-evaluation rules. No candidate is excluded on the basis of branch generation, suffix availability, parsing, or context handling.

Each position retains its trajectory-source checkpoint as the fixed continuation writer. If P_i comes from Search-R1, for example, both Immediate and Delayed are continued by the same frozen Search-R1 checkpoint; the same rule applies to the other sources. Generation capability is therefore fixed within position, while the shared pool covers the state distributions of all three models in nearly equal proportions. Every selector scores all 2,000 prefixes, not only prefixes produced by its own checkpoint.

J.3 Deterministic Selector Scoring and Equal-Budget Selection

All three selectors use the same tokenizer, special action tokens, prefix serialization, and scoring interface. For model m and position i, we perform one deterministic, teacher-forced scoring sweep over the serialized `SEARCH` actions. For each predefined action serialization a, we sum its conditional token log-probabilities and combine unique token-ID sequences with `logsumexp`:

$$\ell_i^{(m)} = \log \sum_{a \in \mathcal{A}_{\text{uniq}}} \exp\left(\sum_{k=1}^{|a|} \log p_{\theta_m}(a_k \mid P_i, a_{<k}) \right). \tag{63}$$

The predefined variants are `" </think>\n<search>"`, `"</think>\n<search>"`, and `" </think> <search>"`. After token-ID deduplication, they are mutually exclusive and have no prefix overlap. We refer to $\ell_i^{(m)}$ as `SEARCH` action log-mass, not as a normalized probability over the full action space. Scores are used only to rank positions within a selector, so absolute calibration across checkpoints is unnecessary.

Scoring uses one forward sweep (`num_scoring_passes=1`) and no sampled rollout; rollout count and random seeds are therefore inapplicable. The interface values temperature $= 0$, top-$p = 1.0$, and maximum new tokens $= 1$ are placeholders only. Any generated token is discarded and does not affect ranking.

The total budget is fixed in advance at 10% of the candidate pool, or 200 positions. We allocate it in proportion to N_d by the largest-remainder method:

$$(B_{\text{ASQA}}, B_{\text{Bamboogle}}, B_{\text{HaluEval}}, B_{\text{PopQA}}, B_{\text{TriviaQA}}) = (47, 12, 47, 47, 47). \tag{64}$$

These budgets are frozen before selector scores, native-trigger counts, or timing outcomes are inspected. Within each dataset, selector m chooses the top-B_d positions by descending $\ell_i^{(m)}$. Ties are broken by ascending `position_id`, which is also frozen before scores and outcomes are available. The selectors rank independently but have identical selection counts within every dataset and overall. Every selected position is timing-evaluable, and no position is replaced or reranked after branch generation.

As an independent diagnostic, we also greedily decode from the same prefix with temperature $= 0$, top-$p = 1.0$, and maximum new tokens $= 64$. We record a native trigger when a valid `SEARCH` action appears before the end of the next complete downstream sentence. This diagnostic executes no retrieval and does not affect B_d, formal selection, or tie-breaking.

J.4 Fixed-Retrieval-Block Timing Intervention

For every frozen candidate position, we construct the retrieval query once as the original question concatenated with the current sentence. A common E5/Wikipedia retriever returns a fixed ordering of the top-3 passages, producing cached evidence E_i; q_i and E_i are then serialized into R_i with a fixed template. Immediate and Delayed share the exact query, passage identifiers, passage text, ordering, serialization, and checksum.

Table 37: Insertion schedules for the fixed-retrieval-block intervention. Immediate and Delayed receive the same prescribed retrieval interaction exactly once and differ only in whether it is inserted before or after the first downstream sentence.

Condition	Before the first downstream sentence	First downstream sentence	Thereafter	Prescribed retrievals
Immediate	Insert fixed R_i	Generate conditioned on R_i	Continue generation with all additional `SEARCH` actions permanently masked	1
Delayed	Do not insert R_i	Generate one complete sentence without access to R_i	Insert the same R_i, then continue with all additional `SEARCH` actions permanently masked	1

The conditions hold fixed the source writer checkpoint, token-level prefix, prior retrieval history, prompt template, decoding configuration, branch seed, stopping rule, generated-token budget, context-handling rule, and R_i. Intervention-inserted tokens do not count against the generated-token budget. The common context-handling rule prevents condition-specific prefix truncation at every candidate position. From the branch point to answer termination, action-level constrained decoding permanently blocks every additional `SEARCH`; this is not implemented with a natural-language instruction or an empty retrieval tag. Both conditions therefore receive exactly one prespecified exogenous retrieval, and their only planned difference is the insertion time of the same R_i relative to the first downstream sentence. Because R_i includes both q_i and E_i, the intervention isolates timing of the retrieval interaction rather than arrival of evidence tokens alone.

Faithfulness judgments use the same sentence-specific temporal masking protocol as the matched-prefix intervention. When the first downstream sentence in Delayed is scored, every evidentiary field contains only the source context, prior retrieval evidence, and generation context available before that sentence was generated. The later-inserted R_i, including E_i, is masked from the structure-only full-output representation and cannot retroactively support the sentence. Neither condition deletes, rewrites, or rolls back generated content. Model identity and condition name are hidden from the judge.

J.5 Suffix NHR, Aggregation, and Statistical Inference

Sentence labels follow the factuality and faithfulness protocol in Appendix H.1. For sentence j, let h_j^{fac} and h_j^{faith} denote the two hallucination indicators. The non-hallucination indicator is

$$n_j = (1 - h_j^{\text{fac}})(1 - h_j^{\text{faith}}). \tag{65}$$

A sentence is non-hallucinated only when it contains neither type of hallucination. The suffix begins with the first complete sentence after the branch point and extends to answer termination; the current sentence ending at the branch point and all preceding content are excluded. Suffix NHR is first averaged over complete downstream sentences within each branch, so longer answers do not receive greater weight.

For a selector-induced position set S, we first compute the paired position-level differences, average them within each dataset, and then take an unweighted macro average over the five datasets:

$$\Delta_{\text{macro}}(S) = \frac{1}{5}\sum_{d=1}^{5} \frac{1}{|S_d|} \sum_{i \in S_d} \Delta_{\text{time}}(i). \tag{66}$$

This order prevents larger datasets from receiving greater weight. We construct 95% confidence intervals with 10,000 paired bootstrap resamples stratified by dataset, preserving the Immediate–Delayed pairing within every position. Because each question contributes at most one position,

clustering by question and by position is equivalent here. All intervals are computed from unrounded position-level statistics.

All three selectors contribute exactly B_d positions within each dataset, yielding $n = 200$ per selector. Every selected position has valid Immediate and Delayed suffix outcomes. No position is excluded, assigned a surrogate effect, replaced, or reranked after branch generation, and all reported effects are computed over the complete prespecified selected sets.

J.6 Dataset-Level and Aggregate Results

Table 38: Dataset-level native-trigger diagnostics, effective sample sizes, and timing benefits. Each selector cell reports native-trigger count / timing-evaluable selected n / Immediate − Delayed suffix NHR in percentage points. Native-trigger counts are diagnostic only and do not determine the formal budget or selected sets.

Dataset	N_d	B_d	Grounded-R1 (full)	Search-R1	Grounded-R1 (episode)
ASQA	469	47	**385 / 47 /** $+9.6$	69 / 47 / $+8.1$	6 / 47 / $+4.9$
Bamboogle	125	12	116 / 12 / $+5.8$	**33 / 12 /** $+6.6$	3 / 12 / $+3.8$
HaluEval	469	47	**354 / 47 /** $+6.4$	26 / 47 / $+5.2$	4 / 47 / -0.6
PopQA	469	47	**344 / 47 /** $+11.2$	45 / 47 / $+8.9$	5 / 47 / $+5.7$
TriviaQA	468	47	**395 / 47 /** $+8.5$	99 / 47 / $+6.2$	6 / 47 / $+3.2$
Total / macro	**2,000**	**200**	**1,594 / 200 /** $+8.3$	**272 / 200 /** $+7.0$	**24 / 200 /** $+3.4$

The observed timing benefit of Grounded-R1 (full) is positive on all five datasets and has the largest point estimate among the three selectors on ASQA, HaluEval, PopQA, and TriviaQA. The exception is Bamboogle, whose budget is only 12 positions, so we do not attach inferential weight to this small cell. Grounded-R1 (episode) is near zero and slightly negative on HaluEval, whereas Grounded-R1 (full) reaches $+6.4$ points on the same dataset. The macro averages preserve the ordering $+8.3$, $+7.0$, and $+3.4$, with full leading on four of five datasets, so the aggregate pattern is not driven by a single benchmark.

The native-trigger diagnostic records a valid `SEARCH` before the next complete sentence at 1,594, 272, and 24 candidate prefixes for Grounded-R1 (full), Search-R1, and Grounded-R1 (episode), respectively. Because the shared pool mixes states written by all three models, it contains off-policy prefixes for every selector; these counts are therefore diagnostic responses on the common pool rather than end-to-end retrieval-call rates on each model's own trajectories. The formal analysis instead gives every selector exactly 200 positions, measuring ranking quality under a fixed budget rather than differences in retrieval frequency.

Table 39: Timing benefit on the complete selected sets and pairwise-exclusive sets. † indicates that the 95% confidence interval for the timing effect within that set excludes zero.

Analysis set / Selector	Immediate − Delayed suffix NHR (pp) ↑
All selected positions ($n = 200$ per selector)	
Grounded-R1 (full)	$+8.3^{\dagger}$ **[**$+2.3$**,** $+14.5$**]**
Search-R1	$+7.0$ [-0.2, $+14.3$]
Grounded-R1 (episode)	$+3.4$ [-2.8, $+9.2$]
Pairwise-exclusive, Grounded-R1 (full) vs. Search-R1	
Grounded-R1 (full) only	$+5.3$ **[**-2.3**,** $+13.1$**]**
Search-R1 only	$+4.4$ [-4.8, $+12.9$]
Pairwise-exclusive, Grounded-R1 (full) vs. Grounded-R1 (episode)	
Grounded-R1 (full) only	$+8.4^{\dagger}$ **[**$+0.8$**,** $+16.2$**]**
Grounded-R1 (episode) only	$+3.8$ [-4.7, $+11.6$]

Among the three equal-budget selected sets, Grounded-R1 (full) is the only selector whose selected positions exhibit a statistically significant reduction in downstream hallucination when the same retrieval block is inserted one sentence earlier. More importantly, after removing positions shared by full and episode, the full-only set retains a significant $+8.4$-point benefit, essentially unchanged from $+8.3$ points on the full selected set, whereas the episode-only set reaches $+3.8$ points. The aggregate

result is therefore not explained solely by positions selected by both policies and is consistent with Phase II changing how the retrieval budget is allocated across sentence-end states.

Relative to Search-R1, the Grounded-R1 (full) point estimate is 1.3 points higher on the complete selected sets and approximately 1.0 point higher on the pairwise-exclusive sets. The direction agrees with the episode comparison, although the gaps are smaller and more uncertain. The Search-R1 comparison thus supplies a directionally consistent external control, while the within-family episode ablation provides the more direct mechanistic evidence.

J.7 Pairwise-Exclusive Estimands and Direct Between-Selector Tests

Pairwise-exclusive denotes the two set differences relative to a specified comparator, not positions globally unique among all three selectors. For the Grounded-R1 (full) set G and comparator set C, the direct contrast is

$$D(G, C) = \Delta_{\text{macro}}(G \setminus C) - \Delta_{\text{macro}}(C \setminus G). \tag{67}$$

Table 40: Direct comparisons between pairwise-exclusive sets. Direct contrasts use unrounded position-level results.

Comparator	Direct contrast (pp)	95% CI
Search-R1	$+1.0$	$[-10.3, +12.8]$
Grounded-R1 (episode)	$+4.6$	$[-6.4, +16.1]$

Both direct point estimates favor Grounded-R1 (full), but both 95% confidence intervals include zero. Accordingly, † in Table 39 indicates a significantly positive timing effect within a position set; it does not indicate a significant difference between Grounded-R1 (full) and its comparator. The results support the more limited conclusion that Grounded-R1 (full) prioritizes positions with higher observed timing benefit, not the stronger claim that a between-selector advantage has been established statistically.

J.8 Writer-Source Balance and Robustness

The main analysis equalizes each selector's dataset-level budget but does not force identical counts within every dataset $\times$ writer-source cell, preserving each selector's natural ranking of states from different sources. The writer-source compositions of the three formal sets are already nearly balanced. To rule out the remaining small differences, we additionally apply two-way controlled rounding to impose identical dataset $\times$ writer-source quotas on all three selectors and recompute the macro timing benefit.

Table 41: Writer-source composition and quota-matched robustness. Source columns count positions generated and continued by the corresponding frozen checkpoint.

Analysis	Selector	Full source	Search-R1 source	Episode source	Timing benefit (pp)
Shared candidate pool	All	667	667	666	–
Dataset-budget main	Grounded-R1 (full)	68	65	67	$+8.3$
Dataset-budget main	Search-R1	66	69	65	$+7.0$
Dataset-budget main	Grounded-R1 (episode)	67	66	67	$+3.4$
Writer-matched robustness	Grounded-R1 (full)	67	67	66	$+8.1$
Writer-matched robustness	Search-R1	67	67	66	$+6.8$
Writer-matched robustness	Grounded-R1 (episode)	67	67	66	$+3.5$

The writer-matched quotas for ASQA, Bamboogle, HaluEval, PopQA, and TriviaQA are 16/16/15, 4/4/4, 16/16/15, 15/16/16, and 16/15/16, respectively, where each triplet is ordered as Full/Search-R1/Episode source. After exact writer-source matching, the selector ordering remains $+8.1$, $+6.8$, and $+3.5$ points, and the largest change from the main analysis is only 0.2 points. Small writer-source composition differences therefore do not explain the aggregate pattern.

Every writer-matched robustness set contains 200 timing-evaluable positions, with no exclusion, replacement, or reranking after branch generation.

J.9 Parsing and Intervention Integrity

All 2,000 frozen candidate positions produce valid Immediate and Delayed suffix outcomes under the common evaluation rules. No position is excluded, assigned a surrogate outcome, replaced, or reranked after branch generation.

Table 42: Implementation and integrity audit. Parsing failures are handled with prespecified deterministic normalization or formatting retries.

Audit item	Result
Immediate–Delayed fixed R_i checksum match for query, top-3 evidence, ordering, content, and serialization	2,000 / 2,000 = 100%
Delayed generates a complete first downstream sentence before insertion of R_i	2,000 / 2,000 = 100%
Additional autonomous retrieval after branching	0 / 4,000 branch executions
Action parse failures in primary propensity scoring	Not applicable
Native-trigger diagnostic parse failures	First pass: 7 / 6,000; final: 0 / 6,000
Final parse failures in structured Immediate/Delayed judge records	0 / 4,000
Integrity-check failures	0 / 2,000

Primary propensity scoring reads teacher-forced token log-probabilities directly and does not depend on action parsing; the native-trigger parser serves only the diagnostic counts. The 4,000 structured judge records correspond to Immediate and Delayed branches for all 2,000 candidate positions. After fixed-format retries, every judge record is valid and no position is excluded because of a parsing failure.

J.10 Conclusion and Scope

The experiment provides four complementary controls. First, the retrieval block R_i, writer, prefix, and retrieval count are fixed within position, so Immediate–Delayed changes only when the same block is inserted relative to the first downstream sentence; the estimand is not interpreted as the timing effect of evidence tokens alone. Second, the three selectors rank the same source-balanced pool without sampling noise and receive identical dataset-level budgets, so the observed pattern cannot be attributed to different underlying candidate pools, unequal retrieval budgets, or selector-side rollout randomness. Third, the Grounded-R1 (full) selected set attains a macro timing benefit of $+8.3$ points and is the only selected set whose confidence interval lies entirely above zero; after positions shared with episode are removed, the full-only set retains a significant $+8.4$-point benefit. Fourth, dataset-level directions, writer-matched quotas, and the integrity audit rule out a single dataset, writer-source imbalance, post-hoc replacement, extra retrieval, retrieval-block mismatch, or outcome missingness as explanations of the pattern.

Under a source-balanced shared pool and identical within-dataset retrieval budgets, Grounded-R1 (full) selects positions where timely evidence yields larger observed reductions in downstream hallucination than at the positions selected by Grounded-R1 (episode) or Search-R1. The within-family comparison in particular indicates that Phase II reallocates the budget toward positions where inserting the same retrieval block before the next claim is especially valuable, rather than expanding the retrieval budget.

This conclusion is restricted to the prespecified equal-budget selected sets drawn from the five-dataset mixed-state pool, to which the three models contribute in nearly equal proportions. The analysis does not establish that all end-to-end gains of Grounded-R1 arise from trigger timing, nor does it justify claiming a statistically significant between-selector advantage while the direct contrast intervals include zero. More precisely, the controlled results provide mechanism evidence, independent of RHRD proxy labels, that Grounded-R1 allocates retrieval in a manner consistent with more effective trigger timing. Together with the causal finding that timely insertion of a fixed retrieval block prevents local error propagation, they close the central empirical chain supporting Grounded-R1.

K RHRD Hard-Threshold Control

The RHRD hard-threshold control tests whether the frozen sentence-level risk threshold is sufficient as an inference-time retrieval policy. It introduces no additional training. We use the exact 7B GRPO

Grounded-R1 (episode-semantic) Phase I checkpoint from which Grounded-R1 (full) is initialized, and fix $\tau = 0.56$ using the validation calibration in Appendix B.8. No threshold, checkpoint, or controller configuration is selected using test results.

At each sentence-end position t, the controller computes $\mathrm{RHRD}_{\mathrm{sent}}(t)$ under the current dynamic context. When $\mathrm{RHRD}_{\mathrm{sent}}(t) \geq \tau$ and retrieval budget remains, the controller overrides the next action with `SEARCH`. The retrieval query is generated with the same query-generation mechanism as the Phase I policy, and the returned top-3 passages are appended to the dynamic context using the same retrieval environment as Grounded-R1 (full). When the threshold is not crossed, the Phase I policy selects a non-retrieval action from $\{\texttt{CONTINUE_THINK}, \texttt{ANSWER}\}$. If the retrieval budget has been exhausted, `SEARCH` is masked and the policy selects between the two non-retrieval actions.

All non-policy evaluation conditions are held fixed, including the retrieval corpus, retriever, top-k, decoding parameters, maximum turns, maximum retrieval budget, answer extraction, and evaluation prompts. We evaluate the controller on the same eight benchmarks and 16 episode-level metrics reported in Table 1. These metrics are independent of the RHRD threshold labels used for Phase II reward construction and the RHRD-alignment diagnostic.

Both variants share the same Phase I initialization, but obtain retrieval behavior differently. The hard-threshold control performs no additional training and directly maps the frozen threshold state to a retrieval or non-retrieval decision, whereas Grounded-R1 (full) applies Phase II policy optimization using the RHRD-derived sentence-level learning signal together with the episode-level quality-cost objective and the Phase I KL anchor.

This control distinguishes direct execution of the binary proxy from policy learning under RHRD-derived supervision. If Phase II collapsed to deterministic threshold imitation, its deployed retrieval behavior would be expected to approach that of the hard-threshold controller. Instead, the learned policy retains near-Phase-I retrieval volume while achieving substantially stronger episode-level performance.

L Additional Ablation Studies and Training Dynamics

This appendix reports additional Grounded-R1 ablations, PPO and GRPO optimizer comparisons, Phase II RHRD-alignment training dynamics, and the training-log fields used for diagnostic figures. These analyses primarily supplement the reward-design ablation and RHRD-alignment diagnostics in Section G; they are complementary to the sentence-level hallucination analysis, which is evaluated separately by sentence-level NHR. Section 4.2 uses a unified 7B GRPO setting for the main reward design ablation, so that the comparison among Grounded-R1 (episode-EM), Grounded-R1 (episode-semantic), and Grounded-R1 (full) isolates reward design rather than optimizer choice. Here we provide the full benchmark-level results and training diagnostics.

PPO is a policy-gradient RL method based on a clipped surrogate objective (Schulman et al., 2017). GRPO is a PPO-style variant used in DeepSeekMath that estimates a group-level baseline and reduces reliance on a separate critic model (Shao et al., 2024). We do not claim that either optimizer is uniformly better. Instead, the results show that optimizer choice affects training dynamics, while the main comparison in Section 4.2 isolates the effect of reward design under a unified GRPO setting.

L.1 Complete Variant Ablation Results Across Benchmarks

This section reports complete benchmark-level results for Grounded-R1 (episode-EM), Grounded-R1 (episode-semantic), and Grounded-R1 (full) under PPO and GRPO. The RHRD hard-threshold control is not an additional training variant and is therefore reported separately in Appendix K. The three variants form a progression from episode-level answer supervision to sentence-level retrieval trigger timing optimization. Grounded-R1 (episode-EM) uses terminal EM, format reward, and retrieval penalties. Grounded-R1 (episode-semantic) replaces the sparse answer-quality signal with a continuous LLM-as-a-judge semantic reward. Grounded-R1 (full) further adds the Phase II sentence-level reward at sentence-end positions, which directly optimizes retrieval trigger timing.

PPO setting. Table 43 provides the corresponding PPO ablation as a complementary optimizer comparison. The PPO results show an overall progression, although not every metric increases

Table 43: Complete PPO ablation results for Grounded-R1 variants. All benchmark-level metrics are reported on the percentage scale. The best result in each metric column is bold; ties are jointly bolded. LLM denotes LLM-as-a-judge score, and NHR denotes non-hallucination rate.

Model	HotpotQA		2Wiki		MuSiQue		Bamboogle	
	EM	LLM	EM	LLM	EM	LLM	EM	LLM
Grounded-R1 (episode-EM, PPO)	27.60	39.45	23.20	29.74	8.63	13.72	22.40	32.02
Grounded-R1 (episode-semantic, PPO)	29.20	**43.30**	24.60	30.60	9.40	15.96	23.20	35.69
Grounded-R1 (full, PPO)	**31.00**	43.03	**25.00**	**30.83**	**9.80**	**17.18**	**27.20**	**37.63**

Model	HaluEval			PopQA			TriviaQA			ASQA	
	EM	LLM	NHR	EM	F1	LLM	EM	F1	LLM	EM	F1
Grounded-R1 (episode-EM, PPO)	37.72	49.98	52.44	37.91	47.42	50.76	58.40	65.84	72.86	60.00	70.35
Grounded-R1 (episode-semantic, PPO)	**38.90**	50.96	53.75	40.95	47.97	52.08	61.60	70.10	77.04	65.00	74.68
Grounded-R1 (full, PPO)	**38.90**	**53.00**	**55.60**	**41.23**	**48.53**	**52.72**	**62.80**	**70.96**	**78.82**	**66.00**	**75.25**

Table 44: Complete GRPO ablation results for Grounded-R1 variants. All benchmark-level metrics are reported on the percentage scale. The best result in each metric column is bold. LLM denotes LLM-as-a-judge score, and NHR denotes non-hallucination rate.

Model	HotpotQA		2Wiki		MuSiQue		Bamboogle	
	EM	LLM	EM	LLM	EM	LLM	EM	LLM
Grounded-R1 (episode-EM, GRPO)	16.25	25.18	21.97	25.32	3.20	8.41	12.00	20.14
Grounded-R1 (episode-semantic, GRPO)	18.00	27.33	22.80	27.37	5.00	10.70	12.80	20.53
Grounded-R1 (full, GRPO)	**34.60**	**48.80**	**29.20**	**38.24**	**16.00**	**23.78**	**36.80**	**51.63**

Model	HaluEval			PopQA			TriviaQA			ASQA	
	EM	LLM	NHR	EM	F1	LLM	EM	F1	LLM	EM	F1
Grounded-R1 (episode-EM, GRPO)	20.56	29.86	30.58	17.53	21.13	22.50	42.80	48.13	52.12	28.00	37.20
Grounded-R1 (episode-semantic, GRPO)	22.13	32.39	33.55	20.20	23.86	26.84	47.40	54.32	59.86	37.00	46.33
Grounded-R1 (full, GRPO)	**42.30**	**56.16**	**59.10**	**41.65**	**48.33**	**52.50**	**62.00**	**70.29**	**77.42**	**66.00**	**75.37**

monotonically. On HotpotQA, EM improves from 27.60 to 29.20 and then to 31.00, while the LLM-as-a-judge score rises from 39.45 to 43.30 and remains comparable at 43.03 after Phase II. On Bamboogle, EM increases from 22.40 to 23.20 and then to 27.20. On HaluEval, EM reaches 38.90 with Grounded-R1 (episode-semantic) and is maintained by Grounded-R1 (full), while NHR improves from 52.44 to 53.75 and then to 55.60. These results support the same conclusion as Section 4.2. Episode-level semantic reward improves answer-quality feedback, while adding Phase II is followed by improvements on most reported metrics. The controlled timing interventions separately provide mechanism evidence about the downstream value and allocation of timely retrieval.

GRPO setting. Table 44 reports the complete GRPO results for the three trained variants underlying the reward-design portion of Table 1. The first two variants are weaker under the current GRPO setting, especially when only episode-level reward is used. Grounded-R1 (full, GRPO) shows a large improvement across all listed benchmarks, suggesting that the RHRD-derived sentence-level reward provides a more actionable learning signal for this optimizer.

The GRPO results show a larger performance gain from Phase II. Grounded-R1 (full, GRPO) improves sharply over Grounded-R1 (episode-semantic, GRPO), with HotpotQA EM increasing from 18.00 to 34.60 and its LLM-as-a-judge score increasing from 27.33 to 48.80. Bamboogle EM increases from 12.80 to 36.80, ASQA EM from 37.00 to 66.00, and HaluEval NHR from 33.55 to 59.10. These results do not imply that GRPO is always preferable to PPO. They show that when the reward includes a sentence-level learning signal aligned with retrieval trigger timing, GRPO can benefit substantially.

Across PPO and GRPO, Grounded-R1 (full) is the strongest variant within each optimizer setting. This supports the reward design claim in Section 4.2. The improvement is not only due to changing answer-quality scoring from EM to semantic reward. The additional Phase II sentence-level learning signal reduces the credit assignment gap between the final episode-level outcome and the sentence-end retrieval decision that shaped the generation trajectory.

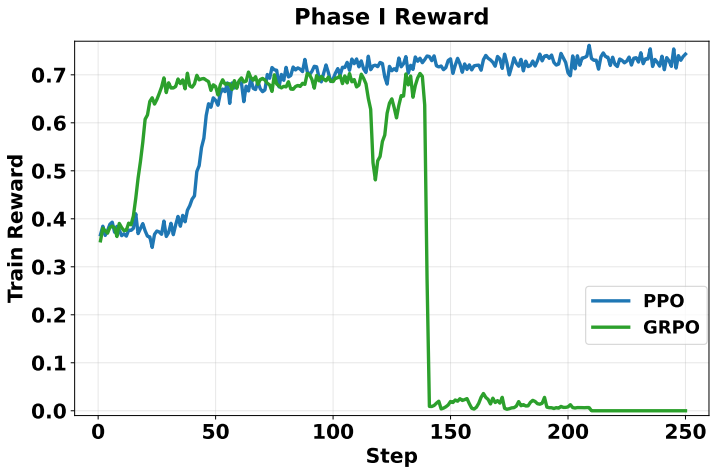


Figure 7: Phase I reward curves under PPO and GRPO for Grounded-R1 (episode-semantic). The curve is used only to compare optimizer behavior under the episode-level quality-cost reward.

Table 45: Representative PPO and GRPO comparison. Metrics are reported on the same scale as Tables 43 and 44.

Variant	Optimizer	HotpotQA EM	Bamboogle EM	HaluEval NHR	PopQA EM	ASQA EM	ASQA F1
Grounded-R1 (episode-semantic)	PPO	29.20	23.20	53.75	40.95	65.00	74.68
Grounded-R1 (episode-semantic)	GRPO	18.00	12.80	33.55	20.20	37.00	46.33
Grounded-R1 (full)	PPO	31.00	27.20	55.60	41.23	66.00	75.25
Grounded-R1 (full)	GRPO	34.60	36.80	59.10	41.65	66.00	75.37

L.2 PPO vs. GRPO: Optimizer Choice and Unified Main Setting

This section compares PPO and GRPO under the training and evaluation settings used in our ablations. The reward curve in Figure 7 corresponds to Phase I episode-level reward learning for Grounded-R1 (episode-semantic). It does not show the Phase II mixed reward for Grounded-R1 (full).

Figure 7 shows that PPO rises more gradually and remains in a higher reward range in the later part of Phase I, whereas GRPO rises earlier but then drops toward a lower reward range under the same episode-level reward setting. This difference confirms that optimizer choice affects training dynamics. We use a unified 7B GRPO setting in Table 1 to keep the reward-design ablation internally consistent, avoid selecting the better-performing optimizer separately for each variant, and ensure that all Grounded-R1 rows refer to the corresponding GRPO checkpoints.

Table 45 connects this training behavior with representative test results. Under Grounded-R1 (episode-semantic), PPO outperforms GRPO on all listed metrics. After adding the Phase II sentence-level reward, Grounded-R1 (full, GRPO) improves substantially and exceeds Grounded-R1 (full, PPO) on several representative metrics.

The unified 7B GRPO setting in Section 4.2 is therefore a consistency and control choice rather than a claim that GRPO is uniformly better or more stable than PPO. Using the same optimizer for all main variants avoids confounding reward design with optimizer selection, while the PPO results provide a complementary optimizer comparison.

L.3 Phase II RHRD-Alignment Training Dynamics Across 3B and 7B

This section studies how alignment between sentence-end policy actions and the frozen RHRD proxy changes during Phase II. Unlike Appendix L.1, which focuses on final benchmark-level outcomes, this analysis is restricted to optimization dynamics under the proxy criterion. Since the Phase II objective places greater weight on missed proxy-positive states, F2 is used as the main diagnostic metric.

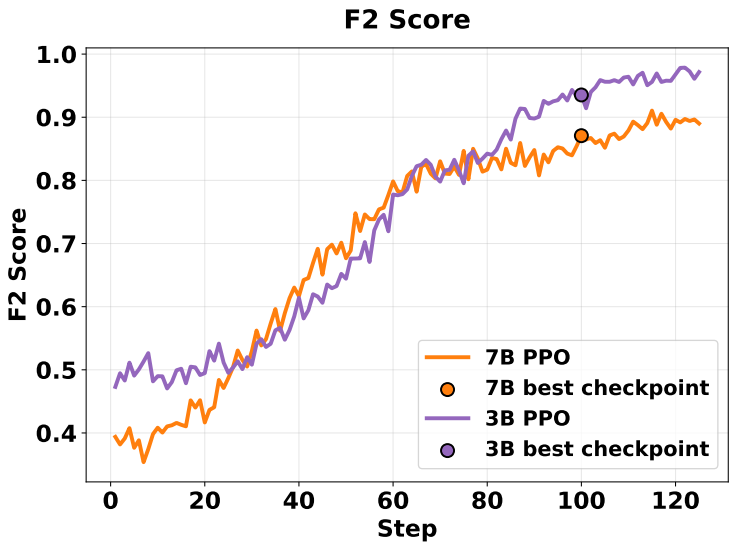
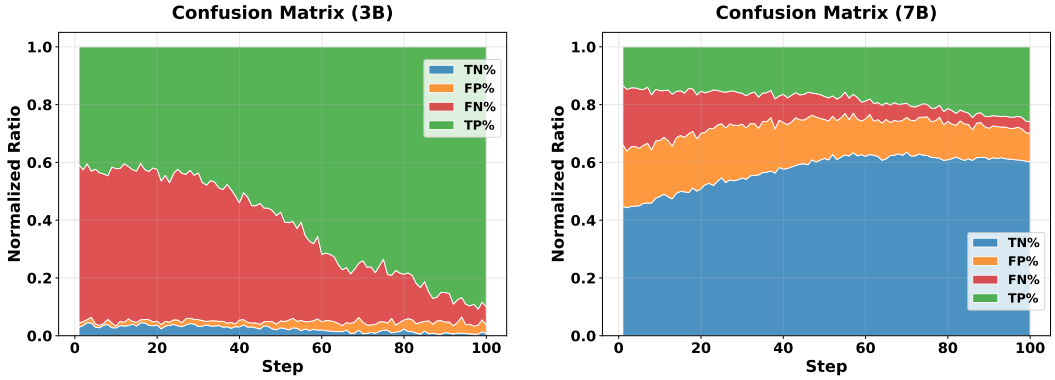


Figure 8: Phase II RHRD-proxy F2 trajectories for 3B and 7B PPO training. F2 is computed from macro precision and recall and measures alignment between sentence-end policy actions and frozen RHRD proxy labels.



(a) 3B (b) 7B

Figure 9: Normalized RHRD-proxy confusion-state dynamics during Phase II. Each panel shows the proportions of TN, FP, FN, and TP under the frozen proxy criterion.

Figure 8 shows increasing proxy F2, computed from macro precision and recall, at both model scales. The 7B curve starts from a lower range, rises during early training, and reaches a higher plateau, while the 3B curve remains in a comparatively high range throughout the displayed steps. These trajectories show that both scales learn the intended RHRD-aligned behavior. They do not establish relative final-answer capability because the plotted F2 values measure agreement with the Phase II proxy rather than downstream task quality.

Figure 9 explains the proxy F2 trends. For 3B, TP expands and FN shrinks rapidly, while FP remains comparatively controlled. For 7B, TN remains large throughout training, while FN decreases and TP increases. The two patterns show increasing agreement with proxy-positive states without an uncontrolled expansion of retrieval actions at proxy-negative states.

The normalized areas should not be read as absolute cross-model ability under an identical risk distribution. They are most informative within each model over training. The key pattern is that FN decreases, TP increases, and FP does not grow without control. Under the frozen proxy criterion, this

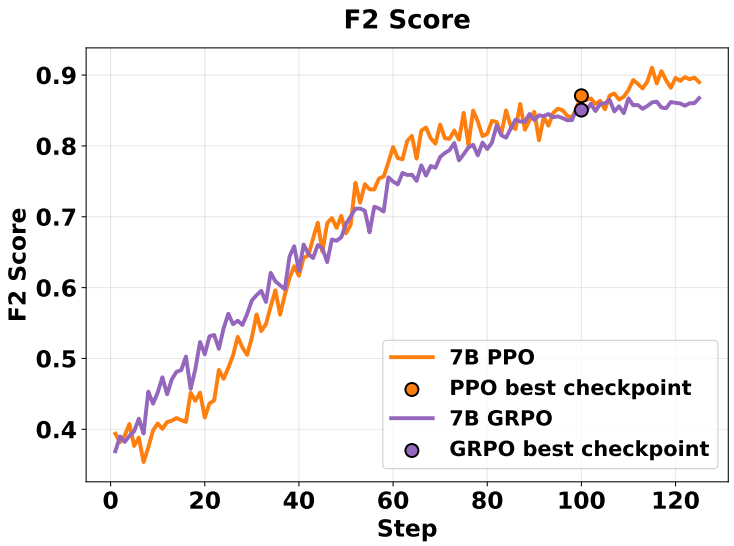


Figure 10: Phase II RHRD-proxy F2 trajectories for 7B PPO and GRPO. F2 is computed from macro precision and recall and measures alignment between sentence-end policy actions and frozen RHRD proxy labels.

supports the narrower conclusion that Phase II learns the intended RHRD-aligned behavior at both model scales.

L.4 Phase II RHRD-Alignment Training Dynamics Under PPO and GRPO

We further compare the 7B Phase II RHRD-alignment trajectories under PPO and GRPO. This comparison complements the scale analysis above by holding model size fixed and varying the optimizer. It is an optimization diagnostic and is not used to rank the optimizers by downstream task performance.

Figure 10 shows that PPO and GRPO follow different early trajectories but both increase substantially before reaching a similar proxy-F2 plateau. The shared trend indicates that the intended RHRD-aligned behavior is learnable under both optimizers rather than being specific to one optimization algorithm. Because the plotted labels are induced by the same frozen RHRD proxy used for Phase II supervision, the curves demonstrate successful objective alignment rather than independent improvement in retrieval utility or optimal retrieval trigger timing.

L.5 Training-log Metrics and Figure Construction Protocol

This section records the training-log fields used to construct the diagnostic figures. It is included for reproducibility and does not introduce new experimental claims. The logs are parsed into JSONL fields and then used to plot reward curves, F2 curves, and normalized confusion-matrix dynamics.

Phase II F2 curves use `step`, `macro_dec_precision`, and `macro_dec_recall`. Accordingly, the plotted F2 is computed from macro precision and recall as:

$$\mathrm{F2} = \frac{5 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{4 \cdot \mathrm{Precision} + \mathrm{Recall}}. \tag{68}$$

Confusion-matrix dynamics use `macro_dec_tp_pct`, `macro_dec_fn_pct`, `macro_dec_fp_pct`, and `macro_dec_tn_pct`. These diagnostics track whether the sentence-level reward increasingly aligns the policy with proxy reference labels induced by the frozen RHRD threshold.

Table 46: Training-log fields used for diagnostic figure construction.

Raw log field	JSONL field	Meaning
`[DEBUG][Step] sample/return_mean: {value}`	`reward_mean`	Mean sample reward
`[DEBUG][Step] sample/return_mean: /std: {value}`	`reward_std`	Reward standard deviation
`[DEBUG][Step] sample/return_mean: /len_mean: {value}`	`len_mean`	Average number of generated sentences
`[DEBUG][EM+Format] Batch reward stats - mean: {value}`	`em_reward_mean`	Mean episode-level EM and format reward
`[DEBUG][Confusion][Macro-Dec] TP%: {value}`	`macro_dec_tp_pct`	TP percentage for retrieval decisions
`[DEBUG][Confusion][Macro-Dec] FN%: {value}`	`macro_dec_fn_pct`	FN percentage for retrieval decisions
`[DEBUG][Confusion][Macro-Dec] FP%: {value}`	`macro_dec_fp_pct`	FP percentage for retrieval decisions
`[DEBUG][Confusion][Macro-Dec] TN%: {value}`	`macro_dec_tn_pct`	TN percentage for retrieval decisions
`[DEBUG][Confusion][Macro-Dec] Precision: {value}`	`macro_dec_precision`	RHRD-alignment precision
`[DEBUG][Confusion][Macro-Dec] Recall: {value}`	`macro_dec_recall`	RHRD-alignment recall
`[DEBUG][Confusion][Macro-Dec] F1: {value}`	`macro_dec_f1`	RHRD-alignment F1 score
`[DEBUG][Confusion][Macro-All] TP%: {value}`	`macro_all_tp_pct`	Global TP percentage
`[DEBUG][Confusion][Macro-All] FN%: {value}`	`macro_all_fn_pct`	Global FN percentage
`[DEBUG][Confusion][Macro-All] FP%: {value}`	`macro_all_fp_pct`	Global FP percentage
`[DEBUG][Confusion][Macro-All] TN%: {value}`	`macro_all_tn_pct`	Global TN percentage
`[DEBUG][Confusion][Macro-All] Precision: {value}`	`macro_all_precision`	Global precision
`[DEBUG][Confusion][Macro-All] Recall: {value}`	`macro_all_recall`	Global recall
`[DEBUG][Confusion][Macro-All] F1: {value}`	`macro_all_f1`	Global F1 score
`[DEBUG][Confusion][Macro-All] DecisionCoverage: {value}`	`macro_all_decision_coverage`	Decision coverage
`[DEBUG][Sentence][Decision] pos_rate: {value}`	`sentence_decision_pos_rate`	Positive reward rate on decision positions
`[DEBUG][Sentence][Decision] neg_rate: {value}`	`sentence_decision_neg_rate`	Negative reward rate on decision positions
`[DEBUG][Sentence][All] pos_rate: {value}`	`sentence_all_pos_rate`	Global positive reward rate
`[DEBUG][Sentence][All] neg_rate: {value}`	`sentence_all_neg_rate`	Global negative reward rate

M LIMITATIONS AND SCOPE

Grounded-R1 is designed to improve retrieval trigger timing by aligning retrieval decisions with local hallucination risk during generation. We discuss several limitations that clarify the intended scope of the method without changing the main conclusions.

First, the RHRD score is an operational proxy for sentence-level hallucination risk rather than a direct observation of hallucination. It uses internal uncertainty, average attention, content-word filtering, and effective subword token count to produce a training-time hallucination risk signal for Phase II reward supervision. This design makes the signal usable for sentence-level reward supervision, but the signal should not be interpreted as a direct verifier of factual correctness or evidence support. Accordingly, Grounded-R1 treats RHRD as a learning signal for retrieval trigger timing, while final answer quality and groundedness are evaluated separately.

Second, RHRD relies on internal model signals such as token distributions and attention patterns. These signals are available for the open-source LLMs used in our experiments, but may be unavailable or implementation-dependent for closed API-only LLMs. Training Grounded-R1 on black-box models would therefore require alternative uncertainty or confidence proxies that do not require access to internal activations. Standard Grounded-R1 inference does not require RHRD.

Third, the effectiveness of dynamic RAG depends on the quality and coverage of the external evidence source. If the retrieval corpus does not contain the required information, or if the retriever returns misleading or weakly relevant evidence, retrieval alone cannot guarantee grounded generation.

Grounded-R1 mitigates poorly timed and redundant retrieval, but it does not replace the need for reliable retrieval infrastructure and well-curated external evidence.

Fourth, our sentence-level formulation reflects a practical trade-off. Sentence-end decisions provide earlier intervention than episode-level rewards while avoiding the fluency and formatting issues caused by token-level interruption. However, some tasks may require finer-grained or coarser-grained intervention points. We view sentence-level optimization as a stable and interpretable design choice for the studied tasks, rather than a claim that sentence-level granularity is universally optimal.

Finally, our empirical study focuses on knowledge-intensive question answering and hallucination-sensitive generation benchmarks with Qwen2.5 backbones and a fixed retrieval setup. The results support the effectiveness of Grounded-R1 under these settings, but broader deployment to other model families, domain-specific corpora, multilingual scenarios, and production retrieval systems may require additional calibration of the training-time risk threshold, reward weights, and retrieval backend. Qwen2.5-72B-Instruct is used both for the Phase I semantic reward and for judge-based evaluation, which can introduce evaluator coupling. We therefore interpret the judge-independent EM and F1 metrics, the matched-prefix final-EM effect, and the human sanity check of sentence-level annotations as complementary evidence. Evaluation with an independent judge or a larger human study remains an important extension.