THINK BEFORE RETRIEVAL: LATENT POLICY REFINEMENT FOR EFFICIENT AGENTIC RAG

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic retrieval-augmented generation (RAG) improves knowledge-intensive question answering through iterative reasoning and search, but repeatedly decides whether retrieval is warranted. Wrong decisions cause over-search when available knowledge is sufficient and under-search when necessary evidence is missing, harming search efficiency and answer quality respectively. Existing methods typically assess retrieval decisions after execution, leaving no explicit mechanism to revise a pending action beforehand. We introduce **LPR-RAG**, which formulates retrieval control as pre-action **L**atent **P**olicy **R**efinement, selectively updating latent decision states before committing to either SEARCH or NONSEARCH decisions. Training proceeds in two stages. In Stage I, ordinary on-policy trajectories optimize the execution adapter with answer and format rewards, while paired counterfactual preferences supervise the decision pathway. To construct this supervision, selected states are forked into matched SEARCH and NONSEARCH continuations under the same frozen policy snapshot and budget; terminal success determines the preferred action, retrieval and processed-token costs break ties, and unresolved pairs are excluded from decision supervision. The resulting preferences train an initial gate and a shared two-step updater to correct initially wrong decisions while preserving correct ones. Stage II freezes all preceding modules and trains a selector to choose the minimum effective refinement depth of zero, one, or two updates. At inference, LPR-RAG applies the selected depth before acting, without counterfactual rollouts, gold answers, or external judges. Across seven single-hop and multi-hop QA benchmarks, LPR-RAG consistently improves answer quality and retrieval decision quality by reducing both redundant and insufficient search. Code is available HERE.

1 INTRODUCTION

Agentic retrieval-augmented generation (RAG) maps a question and evolving evidence to a sequence of reasoning steps, retrieval actions, and a final answer Lewis et al. (2020); Yao et al. (2023). Unlike single-pass RAG, an agent can reformulate its information need after each observation and search again Trivedi et al. (2023); Jiang et al. (2023). This control matters for single-hop questions answerable from parametric knowledge and multi-hop questions requiring entity bridging, comparison, or verification. Because every tool call adds latency and context, success depends not only on finding relevant documents but also on deciding whether external information is needed at each high-level state Lewis et al. (2020); Mallen et al. (2023); Wu et al. (2025b); Wang et al. (2025a).

Adaptive retrieval moved beyond fixed pipelines by interleaving retrieval with reasoning and triggering search from self-reflection, real-time information needs, or question complexity Trivedi et al. (2023); Asai et al. (2024); Su et al. (2024); Jeong et al. (2024). RL-based agents later learned multi-step tool use from terminal answer rewards, while process-supervised methods assigned denser credit through step judgments or turn-level potentials Jin et al. (2025); Xiong et al. (2025); Wu et al. (2025a); Xie et al. (2026). Rollout-tree methods estimate intermediate advantages from descendant outcomes, while efficiency controllers select a sufficient depth Zhang et al. (2026b); Sun et al. (2026). Long-horizon systems organize state and stopping through structured registers or explicit belief-state and exhaustion signals Wang et al. (2025c); Kausik et al. (2026). Latent approaches compress thoughts and subqueries, anchor latent blocks to teacher actions, or refine hidden representations before decoding Zheng & Worring (2026); Jung et al. (2026); Hsu & Yao (2026).

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.

These routes improve action learning, credit assignment, horizon control, or internal computation, yet none compares budget-matched terminal SEARCH and NONSEARCH continuations from the same pending state to supervise selective correction of its provisional action distribution.

At decision time, the relative value of SEARCH and NONSEARCH is counterfactual because an ordinary trajectory reveals only its selected continuation. Over-search occurs when existing knowledge or evidence supports a correct continuation, so another retrieval consumes calls and tokens and may add distracting context Jin et al. (2025); Wu et al. (2025b); Wang et al. (2025a); Qian et al. (2025). Under-search occurs when evidence is missing but the agent continues internally or answers, possibly propagating an unsupported conclusion Wu et al. (2025b). The outcome is also entangled with execution competence: SEARCH may fail because its query is poor, while NONSEARCH may fail because reasoning or answer generation is weak. Thus, one trajectory cannot distinguish a policy-relative action preference from the current policy's ability to execute either choice. Uniform refinement can also disturb correct states, while jointly training a depth controller and an evolving updater creates moving targets that may collapse computation toward one depth. A viable controller must therefore construct preferences without teacher action traces, isolate decision learning from execution failures, and correct hard states without damaging easy ones while delaying depth learning until the updater stabilizes.

LPR-RAG refines a dedicated decision state before committing each high-level retrieval action. At every decision point, an internal `<decide>` vector reads the question, history, and retrieved evidence from read-only context states through cross-attention, then an initial gate predicts SEARCH or NONSEARCH. SEARCH invokes retrieval, whereas NONSEARCH allows either one reasoning step without retrieval before control returns or a final answer; a shared residual updater modifies only the decision vector for at most two iterations. For supervision, a frozen behavior snapshot copies a selected prefix and cached state, forces one action per branch with matched budgets, and continues both until termination or budget exhaustion. Terminal correctness produces a conservative policy-relative preference; retrieval calls issued during the continuations and processed tokens resolve successful ties, while pairs with no successful or unique winner abstain. Intervened branches train only the decision modules, whereas ordinary on-policy trajectories train execution, preventing forced actions from entering execution PPO. Stage I applies correction losses at both updater depths when the initial gate is wrong and consistency losses when it is correct. Stage II freezes all Stage I modules, labels each resolvable state by the earliest depth whose prediction matches its preference, and trains a selector to choose zero, one, or two updates at inference without paired branches, gold answers, or external judges. Our contributions are summarized as follows.

- We formulate agentic retrieval control as policy-relative pre-action correction and construct conservative state-level supervision by comparing budget-matched SEARCH and NONSEARCH continuations from the same frozen-policy state, without teacher action labels or teacher reasoning traces.
- We develop a selective two-step latent refiner that separates hard-state correction from easy-state preservation, then freeze the execution and decision modules before training a minimum-depth selector to avoid coupling adaptive computation to an evolving updater.
- We evaluate LPR-RAG across seven single-hop and multi-hop QA benchmarks using matched initialization and compute reporting, together with equal-supervision, equal-FLOPs, and equal-compute controls for the corresponding comparisons.

2 Related Work

Agentic RAG Retrieval-augmented generation combines parametric generation with non-parametric memory Lewis et al. (2020), while ReAct established a general pattern of interleaving reasoning traces with external actions Yao et al. (2023). Agentic RAG makes this interaction stateful: IRCoT alternates retrieval and reasoning for multi-step question answering Trivedi et al. (2023), and Active Retrieval Augmented Generation decides when and what to retrieve as generation unfolds Jiang et al. (2023). Multi-hop benchmarks further expose bridge-entity and comparison reasoning over multiple supporting documents Yang et al. (2018). Later systems sharpen the control problem. Self-RAG learns retrieval and critique through reflection tokens Asai et al. (2024); DRAGIN triggers retrieval from a real-time estimate of information need Su et al. (2024); Adaptive-RAG routes questions among no retrieval, single-step retrieval, and iterative retrieval according to complexity

Jeong et al. (2024). Boundary- and uncertainty-based approaches retrieve selectively from model confidence or internal state Mallen et al. (2023); Yao et al. (2025), while CtrlA controls retrieval timing and query formation at the representation level Liu et al. (2025). Training has likewise moved beyond supervised action imitation. Search-R1 and R1-Searcher use outcome-based reinforcement learning for autonomous multi-turn search Jin et al. (2025); Song et al. (2025). RAG-Gym applies step-level process supervision to iterative search trajectories Xiong et al. (2025), TIPS adds turn-level information-potential shaping Xie et al. (2026), and TreePS-RAG estimates intermediate values from descendant rollout outcomes Zhang et al. (2026b). HiPRAG couples answer quality with over-search and under-search efficiency Wu et al. (2025a). Complementary work formalizes both failure modes Wu et al. (2025b), discourages needless tool use Wang et al. (2025a); Qian et al. (2025), or learns the earliest sufficient search depth Sun et al. (2026). Laser provides a structured action protocol and persistent context for long-horizon search Wang et al. (2025c), whereas the Context Gathering Decision Process introduces a predicate belief state and a programmatic exhaustion gate Kausik et al. (2026).

Latent Reasoning Continuous-computation methods replace or augment verbal chains of thought with internal-state transitions. Coconut recursively feeds the previous hidden state as the next input embedding, allowing continuous thoughts Hao et al. (2024); Compressed Chain of Thought learns variable-length dense contemplation tokens Cheng & Durme (2024); and CODI self-distills explicit rationales into designated continuous states Shen et al. (2025). Related mechanisms allocate extra computation without exposing full textual reasoning: learned pause tokens delay prediction Goyal et al. (2024), Looped Transformers reuse model depth to realize implicit thoughts across iterations Saunshi et al. (2025), and Soft Thinking propagates probability-weighted mixtures of token embeddings as continuous concept tokens Zhang et al. (2025). SynAdapt aligns synthetic continuous chains of thought and adds difficulty-aware rethinking Wang et al. (2025b), while Adaptive Computation Time provides the broader principle of learning input-dependent computation steps Graves (2016). Retrieval-aware latent methods connect internal computation to actions. LatentRAG uses a fixed set of latent thought and subquery slots within one forward pass, with subquery states driving retrieval Zheng & Worring (2026). Adaptive Latent Agentic Reasoning aligns a latent block to teacher actions Jung et al. (2026), while Learning to Refine Hidden States iteratively updates pre-decoding states under learned depth and action controllers Hsu & Yao (2026).

3 METHOD

LPR-RAG separates retrieval control from action execution. At each high-level state, an initial gate predicts whether to invoke retrieval; a shared latent updater may refine only the corresponding decision representation before the action is committed. Training supervision is obtained by comparing matched `SEARCH` and `NONSEARCH` continuations under a frozen behavior-policy snapshot. The execution and decision pathways are optimized from separate supervision streams with isolated gradients, and a second, frozen-model stage learns the minimum refinement depth required by each state.

3.1 PROBLEM FORMULATION AND MODEL INTERFACE

At decision step t, the agent state is

$$s_t = (q, \mathcal{H}_t, \mathcal{D}_t), \qquad a_t \in \{\texttt{SEARCH}, \texttt{NONSEARCH}\}, \tag{1}$$

where q is the question, $\mathcal{H}_t$ contains previous reasoning, queries, conclusions, and actions, and $\mathcal{D}_t$ contains retrieved documents. `SEARCH` invokes the retriever. `NONSEARCH` is a one-step macro-action: the execution policy either produces one non-retrieval reasoning step and then returns control, or terminates with an answer. Thus zero-retrieval answering and later retrieval remain representable without introducing a separate stopping action. The selected action is injected as an internal control ID that routes execution and is omitted from the visible output.

A frozen backbone is equipped with an execution adapter and a decision pathway. The former generates queries, non-retrieval reasoning, document-conditioned conclusions, and answers. The latter contains Stage-I parameters ϕ_{dec} for the context reader, extractor, initial gate, and shared updater, together with Stage-II selector parameters ψ_{depth}; their gradients are isolated from the

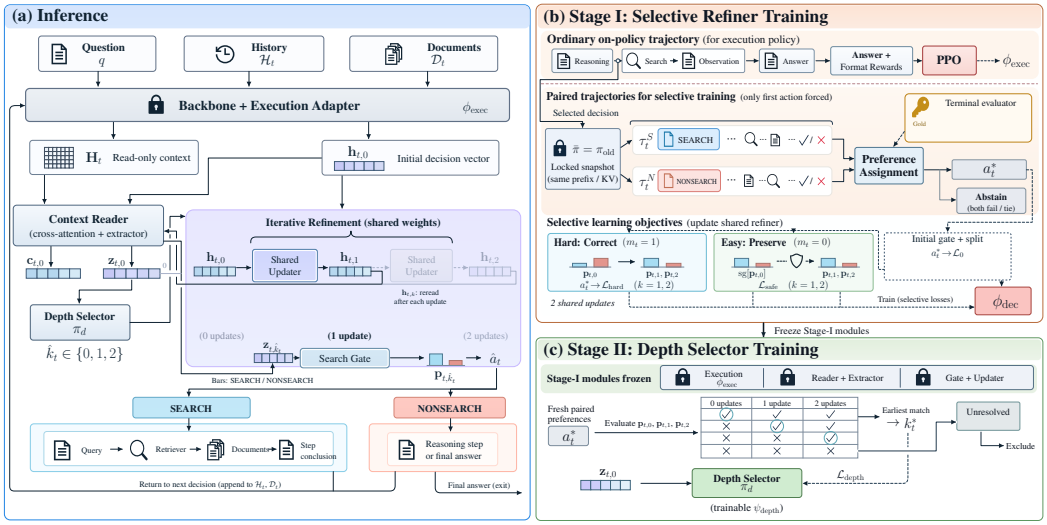


Figure 1: Overview of LPR-RAG. (a) At inference, a depth selector chooses zero, one, or two latent updates before the agent commits to SEARCH or NONSEARCH. (b) In Stage I, ordinary on-policy trajectories optimize the execution adapter with answer and format rewards, while paired counterfactual preferences train a shared refiner to correct initial decisions that disagree with the preference and preserve those that agree. (c) Stage II freezes all preceding modules and trains the depth selector to predict the earliest refinement depth whose action matches the paired preference.

execution pathway. Let $\theta = (\theta_0, \phi_{\text{exec}})$, where θ_0 denotes the frozen backbone parameters and ϕ_{exec} the execution-adapter parameters, and let F_θ denote their composition. Neither ϕ_{dec} nor ψ_{depth} belongs to F_θ. Before each high-level action, an internal `<decide>` position yields read-only context states $\mathbf{H}_t$ and the initial decision vector $\mathbf{h}_{t,0}$:

$$\mathbf{H}_t = F_\theta(q, \mathcal{H}_t, \mathcal{D}_t, \langle\text{decide}\rangle), \qquad \mathbf{h}_{t,0} = \mathbf{H}_t[\langle\text{decide}\rangle]. \tag{2}$$

The decision pathway reads $\mathbf{H}_t$ but does not modify it.

3.2 Paired On-Policy Counterfactual Preference

We firstly construct supervision without teacher actions by comparing both admissible continuations from one frozen-policy state.

Matched branches. At each rollout iteration, the current policy is frozen as a behavior snapshot $\bar{\pi} = \pi_{\text{old}}$ and first generates ordinary on-policy trajectories. Here π_{old} comprises the frozen backbone, the current execution adapter, and the Stage-I decision modules in ϕ_{dec}; because the depth selector is not yet trained, it applies both latent updates before sampling each high-level action. For a selected legal decision state s_t at which both actions are admissible, we clone the exact token prefix, attention state, position indices, and KV cache, then force the first action to be `SEARCH` or `NONSEARCH`. Writing S and N for `SEARCH` and `NONSEARCH`, the resulting continuations τ_t^S and τ_t^N use the same $\bar{\pi}$, retriever, decoding configuration, remaining retrieval budget, generation budget, and timeout rule, and run until legal termination or budget exhaustion. Gold answers are unavailable to either branch and are used only after both continuations finish.

Conservative preference assignment. Let $u(\tau) \in \{0, 1\}$ indicate legal termination, valid format, and correctness under normalized cover exact match. Let $n_{\text{ret}}(\tau)$ count the retrieval calls issued during continuation τ, including the forced first call in τ_t^S, and let $n_{\text{tok}}(\tau)$ count processed tokens.

Define the cost key $\kappa(\tau) = (n_{\text{ret}}(\tau), n_{\text{tok}}(\tau))$. The policy-relative action label is

$$a_t^* = \begin{cases} S, & u(\tau_t^S) > u(\tau_t^N), \\ N, & u(\tau_t^N) > u(\tau_t^S), \\ S, & u(\tau_t^S) = u(\tau_t^N) = 1,\ \kappa(\tau_t^S) \prec_{\text{lex}} \kappa(\tau_t^N), \\ N, & u(\tau_t^S) = u(\tau_t^N) = 1,\ \kappa(\tau_t^N) \prec_{\text{lex}} \kappa(\tau_t^S), \\ \text{abstention}, & \text{otherwise}, \end{cases} \tag{3}$$

where $\prec_{\text{lex}}$ denotes strict lexicographic order. Hence two failed, non-terminating, or exactly tied branches abstain rather than producing a soft label. Equation 3 estimates a finite-sample preference under $\bar{\pi}$; it is neither a teacher action nor a globally optimal action-value estimate.

Information and gradient boundary. Counterfactual branches supervise only the decision pathway. Their forced first actions and intervened continuations never enter execution PPO/GRPO. The execution adapter is updated only from ordinary on-policy trajectories. Within an optimization iteration, ordinary rollouts, branch labels, prefixes, and read-only context features are all produced and cached under the same π_{old} before any parameter update, preventing label–state mismatch.

3.3 Selective Latent Decision Refinement

The refiner rereads the fixed context and updates only the dedicated decision vector; Stage I below defines how these updates are trained selectively. For $k \in \{0, 1\}$, the current decision vector rereads the frozen context and is updated by a shared, bounded residual operator:

$$\begin{aligned} \mathbf{c}_{t,k} &= \text{Attn}(\mathbf{h}_{t,k}, \mathbf{H}_t, \mathbf{H}_t), & \mathbf{z}_{t,k} &= f_{\text{ext}}([\mathbf{h}_{t,k}; \mathbf{c}_{t,k}]), \\ \mathbf{d}_{t,k} &= \mathbf{W}_2 \sigma(\mathbf{W}_1 \mathbf{z}_{t,k}), & \mathbf{v}_{t,k} &= \frac{\mathbf{d}_{t,k}}{\|\mathbf{d}_{t,k}\|_2 + \epsilon}, \\ \alpha_{t,k} &= \alpha_{\max} \tanh(\mathbf{w}_\alpha^\top \mathbf{z}_{t,k}), & \mathbf{h}_{t,k+1} &= \text{LayerNorm}(\mathbf{h}_{t,k} + \alpha_{t,k} \mathbf{v}_{t,k}). \end{aligned} \tag{4}$$

After each update, $\mathbf{c}_{t,k+1}$ and $\mathbf{z}_{t,k+1}$ are recomputed. A shared gate produces the action distribution at depths $k \in \{0, 1, 2\}$,

$$\mathbf{p}_{t,k} = \text{Softmax}(\mathbf{W}_g \mathbf{z}_{t,k}). \tag{5}$$

Only the dedicated decision vector changes; question, history, and evidence representations remain read-only. The maps f_{ext}, $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{w}_\alpha$, and $\mathbf{W}_g$ are learned decision parameters, σ is a pointwise nonlinearity, ϵ stabilizes normalization, and $\alpha_{\max}$ bounds each update.

3.4 Sequential Optimization and Inference

LPR-RAG is trained in two mandatory stages. Stage I learns the execution policy, initial gate, and two-step refiner. Stage II freezes those modules and learns the minimum effective depth.

3.4.1 Stage I: execution and selective refinement

Stage I trains the execution and decision modules from separate supervision streams: ordinary on-policy continuations update execution, whereas branch outcomes supervise the decision pathway.

Execution Learning. Ordinary trajectories update only the execution adapter with PPO using

$$R_{\text{exec}}(\tau) = R_{\text{answer}}(\tau) + \lambda_f R_{\text{format}}(\tau), \tag{6}$$

where R_{answer} scores terminal answer correctness, R_{format} scores structured-output validity, and λ_f weights the latter. We update ϕ_{exec} with the standard clipped PPO objective computed from these ordinary trajectories. PPO gradients are applied only to execution-token generation and the execution adapter; they do not update the binary gate, latent updater, or depth selector.

Selective decision learning. For each non-abstaining paired preference, the initial gate is supervised and a stop-gradient indicator identifies whether its prediction is wrong:

$$\mathcal{L}_0 = \text{CE}(\mathbf{p}_{t,0}, a_t^*), \qquad m_t = \mathbb{I}\left[\arg\max_a \mathbf{p}_{t,0}(a) \neq a_t^*\right]. \tag{7}$$

Thus $\mathcal{L}_0$ learns the provisional action, while m_t partitions states by whether its top-1 prediction already matches the paired preference. For hard states, the updater is trained to correct the initial decision at both refinement depths:

$$\mathcal{L}_{\text{hard}} = m_t[\beta_1 \,\text{CE}(\mathbf{p}_{t,1}, a_t^*) + \beta_2 \,\text{CE}(\mathbf{p}_{t,2}, a_t^*)]\,. \tag{8}$$

Here $\beta_1, \beta_2 \geq 0$ weight one- and two-step correction, respectively, with $\beta_2 \geq \beta_1$. Only hard states contribute to this term, and both refinement depths are supervised to train the full correction operator. For easy states, deeper predictions preserve the correct initial distribution rather than sharpen it further:

$$\mathcal{L}_{\text{safe}} = (1 - m_t) \sum_{k=1}^{2} D_{\text{KL}}(\text{sg}[\mathbf{p}_{t,0}] \,\|\, \mathbf{p}_{t,k})\,, \tag{9}$$

where $\text{sg}[\cdot]$ stops gradients through the initial distribution. This term preserves rather than sharpens the initial prediction, limiting right-to-wrong flips caused by unnecessary refinement. The Stage-I decision objective is

$$\mathcal{L}_{\text{decision}} = \mathcal{L}_0 + \lambda_h \mathcal{L}_{\text{hard}} + \lambda_s \mathcal{L}_{\text{safe}}, \tag{10}$$

where $\lambda_h, \lambda_s \geq 0$ balance hard-state correction and easy-state preservation. This objective updates only ϕ_{dec}; counterfactual continuations never update ϕ_{exec}. At the end of the prescribed Stage-I budget, ϕ_{exec} and ϕ_{dec} are frozen.

3.4.2 Stage II: minimum-effective-depth selector

Using the frozen Stage-I model, paired preferences are collected again and all three distributions $\mathbf{p}_{t,0}, \mathbf{p}_{t,1}, \mathbf{p}_{t,2}$ are evaluated. For every non-abstaining state, the minimum effective depth is

$$k_t^* = \min \left\{ k \in \{0, 1, 2\} : \arg\max_a \mathbf{p}_{t,k}(a) = a_t^* \right\}. \tag{11}$$

States unresolved at every depth are discarded from selector training and recorded separately. The depth selector defines the categorical distribution $\pi_d(k \mid \mathbf{z}_{t,0})$ over $k \in \{0, 1, 2\}$. With all Stage-I modules frozen, the selector parameterized by ψ_{depth} alone minimizes

$$\mathcal{L}_{\text{depth}} = \text{CE}(\pi_d(\cdot \mid \mathbf{z}_{t,0}; \psi_{\text{depth}}), k_t^*)\,. \tag{12}$$

The two stages are sequential and both are part of the final method; fixed-depth variants are ablations.

3.4.3 Inference

At each high-level decision state, the execution model first encodes the question, interaction history, and retrieved evidence into the read-only context states $\mathbf{H}_t$ and the initial decision state $\mathbf{h}_{t,0}$. The frozen depth selector chooses $\hat{k}_t = \arg\max_k \pi_d(k \mid \mathbf{z}_{t,0}; \psi_{\text{depth}})$. The shared refiner then applies exactly $\hat{k}_t$ latent updates, after which the gate commits $\hat{a}_t = \arg\max_a \mathbf{p}_{t,\hat{k}_t}(a)$. The execution policy realizes the selected action. A `SEARCH` step generates a query, retrieves documents, and produces a step conclusion; a `NONSEARCH` step either advances the reasoning without retrieval or terminates with the final answer. If the trajectory does not terminate, the resulting history defines the next decision state and the procedure repeats. Inference uses neither paired counterfactual rollouts nor gold answers or external judges.

4 Experiments

4.1 Experimental Setup

Benchmarks. Following the evaluation protocol of HiPRAG and Search-R1, training uses the official training sets of NQ (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018). Evaluation covers seven QA benchmarks: NQ, TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023) as single-hop datasets, and HotpotQA, 2WikiMultihopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023) as multi-hop datasets. NQ and HotpotQA are in-domain; the remaining five splits measure out-of-domain generalization. All trainable methods share the same initialization, retriever, corpus, and search budget.

Table 1: Main results of CEM (higher is better) in percentage on seven QA benchmarks. The upper block contains prompting-based methods; the lower block contains RL-trained methods. Best and second-best results in each column are marked in **bold** and underlined, respectively.

Method	General QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	
Direct Inference	27.4	26.2	41.0	57.9	16.5	7.5	16.5	27.6
Standard RAG	50.7	55.4	65.3	57.8	21.0	19.0	17.6	41.0
IRCoT	28.1	35.5	43.3	50.7	38.0	18.8	21.6	33.7
Search-o1	39.5	42.7	57.6	56.7	45.2	15.9	18.7	39.5
β-GRPO	64.4	75.4	59.5	53.8	65.3	24.5	**51.7**	56.4
Search-R1	63.5	72.9	61.3	54.1	63.4	24.8	42.4	54.6
HiPRAG	67.3	75.6	64.8	58.8	67.9	**25.6**	45.9	58.0
TreePS-RAG	66.8	75.4	64.3	59.3	67.3	24.4	47.2	57.8
AutoSearch	65.0	74.4	64.2	56.4	65.4	24.1	44.8	56.3
ALAR	66.3	75.0	63.8	57.1	66.7	24.4	48.3	57.4
LPR-RAG (ours)	**69.5**	**77.2**	**66.8**	**60.4**	**69.4**	25.4	46.4	**59.3**

Evaluation metrics. We adopt the HiPRAG evaluation suite for comparability. Cover Exact Match (CEM) counts an answer as correct if any normalized gold alias appears as a substring of the normalized prediction. Over-search rate (OSR) and under-search rate (USR) measure the fraction of identifiable search steps judged redundant and of identifiable non-search steps judged factually or logically faulty, using the same judge configuration as HiPRAG. We additionally report the average number of search calls per question (AvgSearch). All methods are evaluated under three seeds; tables report seed means, and paired-seed bootstrap confidence intervals are provided in Appendix B.

Baselines. We evaluate two groups of baselines in our environment. (i) Prompting-based methods: Direct Inference, Standard RAG (Lewis et al., 2020) with a single retrieval turn, IRCoT (Trivedi et al., 2023) interleaving retrieval with chain-of-thought reasoning, and Search-o1 (Li et al., 2025) with agentic search-augmented reasoning. (ii) RL-trained methods: Search-R1 (Jin et al., 2025) with outcome and format rewards only, HiPRAG (Wu et al., 2025a) with hierarchical process rewards, TreePS-RAG (Zhang et al., 2026b) with tree-based process credit, AutoSearch (Sun et al., 2026), which learns a sufficient search depth, ALAR (Jung et al., 2026), the nearest latent-agent baseline, and β-GRPO (Wu et al., 2025b). More baselines are additionally reported in Appendix A.

Implementation details. All methods use Qwen2.5-3B-Instruct with a unified structured tool prompt, an E5-base retriever over the 2018 Wikipedia dump with top-3 passages, and at most four search turns per question. Stage I runs 400 PPO steps with answer and format rewards, a branch ratio of ρ=0.25, one paired rollout per action (M=1), a maximum refinement depth of $K_{\max}$=2, and loss weights β_1=0.5, β_2=1.0, λ_h=1.0, λ_s=0.5, and λ_f=0.2. Stage II collects 20,000 non-abstaining states with the frozen Stage-I model, splits them 90/10, and trains only the depth selector for three epochs. Equal-compute controls and 7B/GRPO extensions are reported in Appendices C and D.

4.2 Main Results

As shown in Table 1, LPR-RAG attains the best macro average among all methods. Since all RL methods share the same backbone and retrieval environment, the comparison isolates the effect of training design rather than model or retriever strength. In the paired per-seed comparison against HiPRAG, LPR-RAG holds a positive margin on six of seven datasets. The exception is MuSiQue, the only dataset where the paired margin turns slightly negative; we attribute this to HiPRAG being particularly strong on this split. We report this exception explicitly rather than averaging it away.

Table 2 summarizes search behavior, and Figure 2 places LPR-RAG and representative agentic baselines in the accuracy–efficiency plane. LPR-RAG achieves the lowest over-search and under-search rates in both the single-hop and multi-hop groups while also issuing the fewest search calls, so the gains in Table 1 stem from better-calibrated retrieval decisions rather than from trading one failure mode for another. Among the methods shown in Figure 2, LPR-RAG combines the highest answer quality with the lowest average search cost.

Table 2: Search behavior of agentic methods: over-search and under-search rates (%) on single-hop and multi-hop groups, and average search calls per question.

Method	Single-hop		Multi-hop		AvgSearch↓
	OSR↓	USR↓	OSR↓	USR↓	
β-GRPO	8.9	45.2	7.7	27.1	2.0
Search-R1	10.0	46.6	8.3	27.3	2.3
HiPRAG	6.5	36.4	5.0	21.2	1.8
TreePS-RAG	8.7	45.0	7.4	26.3	2.1
AutoSearch	5.6	50.8	5.1	29.8	1.9
ALAR	7.8	43.1	6.6	25.2	2.0
LPR-RAG (ours)	4.1	24.2	3.2	14.0	1.6

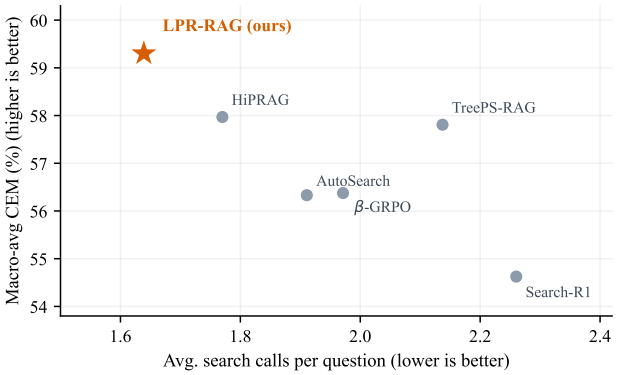


Figure 2: Answer quality versus search cost for LPR-RAG and representative agentic baselines.

4.3 Ablation Study

Table 3 isolates each design choice under the full evaluation protocol. Training with outcome rewards alone causes the largest degradation, so the gains stem from policy-relative preferences rather than longer training; alternative supervision schemes—short-horizon branches, F1 soft preferences, branch replay into execution RL, and $M{=}3$ rollouts at tripled branch cost—either underperform or add little, justifying the terminal-continuation, $M{=}1$ default. Under identical supervision, a direct gate and equal-parameter or equal-FLOPs MLPs fall short of the shared residual updater, and removing the easy-state consistency term or adopting uniform two-step supervision further degrades results. Fixed-depth variants all trail the adaptive system, and jointly training the depth selector with the evolving updater adds no gain over the weakest fixed configuration—evidence for the two-stage schedule of Section 3.4—while the offline depth oracle upper-bounds the learned selector with a small gap. Finally, zeroing the updater output, swapping update vectors, or shuffling depth labels all degrade performance, confirming reliance on refined decision states rather than the initial gate; training only the decision pathway retains most of the gain over outcome-only training, whereas the execution adapter alone recovers far less, locating the main benefit in retrieval-decision supervision.

4.4 Analysis

Selective computation. Figure 3 examines where refinement is spent. When the initial gate is already correct, the selector overwhelmingly chooses zero updates; when it is wrong, deeper refinement dominates. Correction rates and the share of deep computation both decrease as the initial-gate margin grows, and right-to-wrong flips remain rare. The selector therefore allocates internal computation to uncertain states rather than refining uniformly.

Label quality and relation to ALAR. Repeating branch rollouts three times on a fixed audit subset shows high agreement with single-rollout labels (86%); only a small fraction of states remains unresolved at every depth, and such states are discarded rather than mislabeled. Comparing per-state computation allocation, LPR-RAG's deep states overlap only moderately with ALAR's latent-mode states, and deep refinement is considerably more concentrated on initial-gate errors: ALAR

Table 3: Ablation study. The reference row is the main configuration ($M=1$); the offline depth oracle replaces the learned selector with the best fixed depth per state and serves as an upper bound.

Group	Variant	CEM↑	AvgSearch↓
	LPR-RAG (full, $M=1$)	**59.3**	1.6
Counterfactual supervision	outcome rewards only (no branches)	54.4	2.3
	short-horizon branches	57.7	1.5
	F1 soft preference (no abstention)	58.4	1.6
	branch replay into execution RL	58.7	1.6
	$M=3$ rollouts per action	59.5	1.6
Refiner design	direct gate (no refinement)	59.0	1.7
	equal-parameter MLP	59.0	1.7
	equal-FLOPs MLP	59.1	1.7
	uniform two-step CE	59.1	1.7
	hard-only (no easy consistency)	59.2	1.6
Depth training	fixed depth 0	59.0	1.7
	fixed depth 2	59.1	1.6
	joint dynamic depth	59.0	1.6
	offline depth oracle	*59.5*	1.6
Causal checks	zeroed updater output	58.9	1.7
	swapped update vectors	58.6	1.8
	shuffled depth labels	59.0	1.7
Attribution	decision pathway only	58.2	1.6
	execution pathway only	54.2	2.2

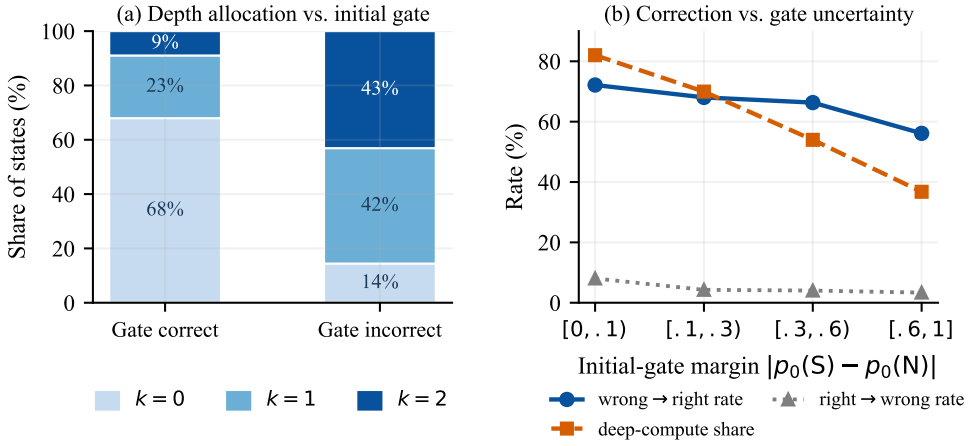


Figure 3: Where refinement is spent. (a) Refinement-depth allocation conditioned on initial-gate correctness. (b) Wrong-to-right correction, right-to-wrong flip, and deep-compute share versus the initial-gate margin: correction and deep compute concentrate at low margins.

adapts the reasoning mode, whereas LPR-RAG targets retrieval decisions whose provisional action distribution disagrees with the paired preference.

5 Conclusion

In this work, we identify and address a structural gap in agentic RAG: retrieval decisions are committed as soon as they are formed, with no mechanism to revise a pending action before its cost is incurred. To bridge this gap, we propose LPR-RAG, which unifies paired on-policy counterfactual preferences, a selective two-step refiner, and a two-stage depth schedule, turning each SEARCH or NONSEARCH decision into a revisable computational object. Extensive evaluations across seven QA benchmarks confirm that LPR-RAG consistently improves answer quality while reducing both redundant and insufficient search. Our findings suggest that comparing budget-matched continuations from the same pending state, and refining only the dedicated decision representation before acting, is a promising recipe for search-efficient agentic RAG; extending it to open web settings with query safety constraints and to grounded, citation-required settings remains future work.

AI USE STATEMENT

In this work, we used LLM for language editing and proofreading of the paper. We have not used LLM for generating research ideas, designing or conducting experiments, producing experimental results, or analyzing data; all research content was conceived, executed, and verified by the authors. We have reviewed all AI-assisted text and take full responsibility for the final content of this work, including all text, claims, and artifacts.

ETHICS STATEMENT

This work uses only publicly available question-answering benchmarks (NQ, TriviaQA, PopQA, HotpotQA, 2WikiMultihopQA, MuSiQue, and Bamboogle) and a public Wikipedia corpus, and involves no human subjects or personal data. Like all retrieval-augmented systems, agentic RAG can produce incorrect or unsupported answers, and its search decisions carry latency and computational cost; we therefore report both accuracy and search-efficiency metrics, including negative results, to give a balanced picture of the trade-offs. The trained agents are intended for research on retrieval-control efficiency and are not designed for safety-critical or high-stakes deployment. This research received no specific external funding, and the authors have no competing interests to disclose.

REPRODUCIBILITY STATEMENT

We are committed to ensuring the reproducibility of our work. Section 4.1 provides a complete overview of our experimental framework, including the training and evaluation datasets, the evaluation metrics (CEM, OSR, USR, and AvgSearch), all baselines, and implementation details such as the backbone model, retriever, corpus, search budget, and training hyperparameters. Section 3 fully specifies the LPR-RAG method, including the paired counterfactual preference construction, the selective refinement objective, and the two-stage training schedule. The appendix contains full per-seed statistics with confidence intervals for all methods, the complete set of 45 ablation variants, training and inference cost breakdowns, per-dataset latency measurements, and case studies. We release the code, configurations, and data-processing scripts.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *CoRR*, abs/2412.13171, 2024. doi: 10.48550/ARXIV.2412.13171. URL https://doi.org/10.48550/arXiv.2412.13171.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=ph04CRkPdC.

Alex Graves. Adaptive computation time for recurrent neural networks. *CoRR*, abs/1603.08983, 2016. URL http://arxiv.org/abs/1603.08983.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *CoRR*, abs/2412.06769, 2024. doi: 10.48550/ARXIV.2412.06769. URL https://doi.org/10.48550/arXiv.2412.06769.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020*, pp. 6609–6625, 2020. doi: 10.18653/v1/2020.coling-main.580.

Chia-Hsuan Hsu and Jui-Ming Yao. Learning to refine hidden states for reliable LLM reasoning. *CoRR*, abs/2606.17524, 2026. doi: 10.48550/ARXIV.2606.17524. URL https://doi.org/10.48550/arXiv.2606.17524.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 7036–7050. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.389. URL https://doi.org/10.18653/v1/2024.naacl-long.389.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 7969–7992. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.495. URL https://doi.org/10.18653/v1/2023.emnlp-main.495.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *CoRR*, abs/2503.09516, 2025. doi: 10.48550/ARXIV.2503.09516. URL https://doi.org/10.48550/arXiv.2503.09516.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-1147. URL https://doi.org/10.18653/v1/P17-1147.

Dongwon Jung, Peng Shi, Yi Zhang, Junshan Zhang, and Muhao Chen. Adaptive latent agentic reasoning. *CoRR*, abs/2606.02871, 2026. doi: 10.48550/ARXIV.2606.02871. URL https://doi.org/10.48550/arXiv.2606.02871.

Chinmaya Kausik, Adith Swaminathan, and Nathan Kallus. The context gathering decision process: A POMDP framework for agentic search. *CoRR*, abs/2605.07042, 2026. doi: 10.48550/ARXIV.2605.07042. URL https://doi.org/10.48550/arXiv.2605.07042.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466, 2019. doi: 10.1162/TACL_A_00276. URL https://doi.org/10.1162/tacl_a_00276.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *CoRR*, abs/2501.05366, 2025. URL http://arxiv.org/abs/2501.05366.

Huanshuo Liu, Hao Zhang, Zhijiang Guo, Jing Wang, Kuicai Dong, Xiangyang Li, Yi Quan Lee, Cong Zhang, and Yong Liu. Ctrla: Adaptive retrieval-augmented generation via inherent control.

In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pp. 12592–12618. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-ACL.652. URL https://doi.org/10.18653/v1/2025.findings-acl.652.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 9802–9822. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.546. URL https://doi.org/10.18653/v1/2023.acl-long.546.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, volume EMNLP 2023 of *Findings of ACL*, pp. 5687–5711. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.378. URL https://doi.org/10.18653/v1/2023.findings-emnlp.378.

Cheng Qian, Emre Can Acikgoz, Hongru Wang, Xiusi Chen, Avirup Sil, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. SMART: self-aware agent for tool overuse mitigation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pp. 4604–4621. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-ACL.239. URL https://doi.org/10.18653/v1/2025.findings-acl.239.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=din0lGfZFd.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: compressing chain-of-thought into continuous space via self-distillation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 677–693. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.36. URL https://doi.org/10.18653/v1/2025.emnlp-main.36.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *CoRR*, abs/2503.05592, 2025. doi: 10.48550/ARXIV.2503.05592. URL https://doi.org/10.48550/arXiv.2503.05592.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. DRAGIN: dynamic retrieval augmented generation based on the real-time information needs of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 12991–13013. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.702. URL https://doi.org/10.18653/v1/2024.acl-long.702.

Jingbo Sun, Wenyue Chong, Songjun Tu, Qichao Zhang, Yaocheng Zhang, Jiajun Chai, Xiaohan Wang, Wei Lin, Guojun Yin, and Dongbin Zhao. Autosearch: Adaptive search depth for efficient agentic RAG via reinforcement learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 28059–28079. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.FINDINGS-ACL.1399. URL https://doi.org/10.18653/v1/2026.findings-acl.1399.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554, 2022. doi: 10.1162/TACL_A_00475. URL https://doi.org/10.1162/tacl_a_00475.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 10014–10037. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.557. URL https://doi.org/10.18653/v1/2023.acl-long.557.

Hongru Wang, Cheng Qian, Wanjun Zhong, Xiusi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong, and Heng Ji. Acting less is reasoning more! teaching model to act efficiently, 2025a. URL https://arxiv.org/abs/2504.14870.

Jianwei Wang, Ziming Wu, Fuming Lai, Shaobing Lian, and Ziqian Zeng. Synadapt: Learning adaptive reasoning in large language models via synthetic continuous chain-of-thought. *CoRR*, abs/2508.00574, 2025b. doi: 10.48550/ARXIV.2508.00574. URL https://doi.org/10.48550/arXiv.2508.00574.

Shuting Wang, Qiaolin Xia, Vich Wang, Herberttli, Bobsimons, and Zhicheng Dou. Laser: Governing long-horizon agentic search via structured protocol and context register. *CoRR*, abs/2512.20458, 2025c. doi: 10.48550/ARXIV.2512.20458. URL https://doi.org/10.48550/arXiv.2512.20458.

Peilin Wu, Mian Zhang, Kun Wan, Wentian Zhao, Kaiyu He, Xinya Du, and Zhiyu Chen. Hiprag: Hierarchical process rewards for efficient agentic retrieval augmented generation. *CoRR*, abs/2510.07794, 2025a. doi: 10.48550/ARXIV.2510.07794. URL https://doi.org/10.48550/arXiv.2510.07794.

Peilin Wu, Mian Zhang, Xinlu Zhang, Xinya Du, and Zhiyu Chen. Search wisely: Mitigating sub-optimal agentic searches by reducing uncertainty. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 19723–19734. Association for Computational Linguistics, 2025b. doi: 10.18653/V1/2025.EMNLP-MAIN.998. URL https://doi.org/10.18653/v1/2025.emnlp-main.998.

Yutao Xie, Nathaniel Thomas, Nicklas Hansen, Yang Fu, Li Erran Li, and Xiaolong Wang. TIPS: turn-level information-potential reward shaping for search-augmented llms. *CoRR*, abs/2603.22293, 2026. doi: 10.48550/ARXIV.2603.22293. URL https://doi.org/10.48550/arXiv.2603.22293.

Guangzhi Xiong, Qiao Jin, Xiao Wang, Yin Fang, Haolin Liu, Yifan Yang, Fangyuan Chen, Zhixing Song, Dengyu Wang, Minjia Zhang, Zhiyong Lu, and Aidong Zhang. Rag-gym: Optimizing reasoning and search agents with process supervision. *CoRR*, abs/2502.13957, 2025. doi: 10.48550/ARXIV.2502.13957. URL https://doi.org/10.48550/arXiv.2502.13957.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1259. URL https://doi.org/10.18653/v1/d18-1259.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 27022–27043. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.1312. URL https://doi.org/10.18653/v1/2025.acl-long.1312.

Chao Zhang, Yuhao Wang, Derong Xu, Haoxin Zhang, Yuanjie Lyu, Yuhao Chen, Shuochen Liu, Tong Xu, Xiangyu Zhao, Yan Gao, Yao Hu, and Enhong Chen. Tearag: A token-efficient agentic retrieval-augmented generation framework. *ACM Trans. Inf. Syst.*, 44(6):126:1–126:35, 2026a. doi: 10.1145/3818621. URL https://doi.org/10.1145/3818621.

Tianhua Zhang, Kun Li, Junan Li, Yunxiang Li, Hongyin Luo, Xixin Wu, Jim Glass, and Helen Meng. Treeps-rag: Tree-based process supervision for reinforcement learning in agentic RAG. *CoRR*, abs/2601.06922, 2026b. doi: 10.48550/ARXIV.2601.06922. URL https://doi.org/10.48550/arXiv.2601.06922.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, and Xin Wang. Soft thinking: Unlocking the reasoning potential of llms in continuous concept space. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/f7396d1c54d51416958d63e285377103-Abstract-Conference.html.

Yijia Zheng and Marcel Worring. Latentrag: Latent reasoning and retrieval for efficient agentic RAG. *CoRR*, abs/2605.06285, 2026. doi: 10.48550/ARXIV.2605.06285. URL https://doi.org/10.48550/arXiv.2605.06285.

A Additional Baselines

Table 4 reports additional baselines under the full evaluation protocol. TIPS applies turn-level information-potential shaping, TeaRAG trains with knowledge-sufficiency preference pairs, and the exhaustion gate is a programmatic baseline that stops searching when queries repeat or observations stop being novel. The counterfactual direct gate (the direct-gate ablation of Table 3 under paired supervision) receives the same supervision as LPR-RAG but maps the read decision state directly to an action without latent refinement; it is the strongest of these baselines. TIPS (Xie et al., 2026), TeaRAG (Zhang et al., 2026a), and the exhaustion gate remain below the RL-trained group of Table 1, whereas the direct gate shows that the paired counterfactual supervision itself accounts for most of the gain over outcome-only training, surpassing all RL baselines even without refinement. The remaining margin of LPR-RAG over the direct gate (+0.3 CEM, 0.1 fewer search calls) is attributable to selective latent refinement, as analyzed in Section 4.3.

Table 4: Additional baselines under the full evaluation: CEM (%) averaged over three seeds.

Method	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
TIPS	64.4	73.4	62.4	55.1	64.8	22.7	43.2	55.1
TeaRAG	64.7	74.1	62.8	55.6	65.2	23.5	44.3	55.7
Exhaustion gate	65.2	74.4	63.1	56.2	66.1	24.3	45.6	56.4
Counterfactual direct gate	68.0	76.7	65.7	59.1	68.8	25.4	49.6	59.0

B Statistical Details

Table 5 reports per-dataset means, sample standard deviations over the three training seeds, and 95% confidence intervals obtained by deterministic paired-seed bootstrap (27 index resamples, type-7

quantiles). Confidence intervals for the macro average are computed by resampling seeds jointly across datasets, preserving the pairing.

Table 5: Full statistics for Table 1: mean ± sample standard deviation over three seeds, with paired-seed bootstrap 95% confidence intervals in brackets.

Method	NQ	TriviaQA
Direct Inference	27.4±0.2 [27.2, 27.5]	26.2±0.2 [26.0, 26.3]
Standard RAG	50.7±0.3 [50.5, 50.9]	55.4±0.3 [55.2, 55.6]
IRCoT	28.1±0.2 [28.0, 28.2]	35.5±0.2 [35.4, 35.6]
Search-o1	39.5±0.2 [39.3, 39.6]	42.7±0.2 [42.5, 42.9]
β-GRPO	64.4±0.2 [64.2, 64.6]	75.4±0.2 [75.2, 75.6]
Search-R1	63.5±0.3 [63.2, 63.7]	72.9±0.3 [72.6, 73.1]
HiPRAG	67.3±0.4 [67.0, 67.6]	75.6±0.3 [75.4, 75.8]
TreePS-RAG	66.8±0.3 [66.6, 67.0]	75.4±0.3 [75.2, 75.6]
AutoSearch	65.0±0.3 [64.8, 65.2]	74.4±0.4 [74.1, 74.6]
ALAR	66.3±0.3 [66.0, 66.5]	75.0±0.2 [74.8, 75.2]
LPR-RAG (ours)	69.5±0.1 [69.4, 69.5]	77.2±0.3 [77.0, 77.4]
	PopQA	HotpotQA
Direct Inference	41.0±0.2 [40.8, 41.1]	57.9±0.2 [57.7, 58.0]
Standard RAG	65.3±0.3 [65.1, 65.5]	57.8±0.3 [57.6, 58.0]
IRCoT	43.3±0.2 [43.2, 43.4]	50.7±0.2 [50.6, 50.8]
Search-o1	57.6±0.2 [57.4, 57.8]	56.7±0.2 [56.5, 56.9]
β-GRPO	59.5±0.2 [59.3, 59.7]	53.8±0.2 [53.6, 54.0]
Search-R1	61.3±0.0 [61.3, 61.3]	54.1±0.2 [53.9, 54.3]
HiPRAG	64.8±0.3 [64.6, 65.1]	58.8±0.1 [58.7, 58.8]
TreePS-RAG	64.3±0.1 [64.1, 64.4]	59.3±0.3 [59.1, 59.5]
AutoSearch	64.2±0.2 [64.0, 64.4]	56.4±0.2 [56.3, 56.5]
ALAR	63.8±0.1 [63.7, 63.8]	57.1±0.4 [56.9, 57.5]
LPR-RAG (ours)	66.8±0.1 [66.7, 66.9]	60.4±0.2 [60.2, 60.5]
	2WikiMultihopQA	MuSiQue
Direct Inference	16.5±0.2 [16.3, 16.6]	7.5±0.2 [7.3, 7.6]
Standard RAG	21.0±0.3 [20.8, 21.2]	19.0±0.3 [18.8, 19.2]
IRCoT	38.0±0.2 [37.9, 38.1]	18.8±0.1 [18.7, 18.9]
Search-o1	45.2±0.2 [45.0, 45.3]	15.9±0.2 [15.7, 16.1]
β-GRPO	65.3±0.2 [65.1, 65.5]	24.5±0.3 [24.3, 24.7]
Search-R1	63.4±0.1 [63.4, 63.5]	24.8±0.1 [24.7, 24.9]
HiPRAG	67.9±0.3 [67.7, 68.1]	25.6±0.1 [25.5, 25.7]
TreePS-RAG	67.3±0.4 [67.0, 67.6]	24.4±0.1 [24.4, 24.5]
AutoSearch	65.4±0.2 [65.3, 65.6]	24.1±0.2 [23.9, 24.2]
ALAR	66.7±0.2 [66.6, 66.9]	24.4±0.3 [24.2, 24.6]
LPR-RAG (ours)	69.4±0.1 [69.4, 69.5]	25.4±0.2 [25.2, 25.6]
	Bamboogle	Macro Avg.
Direct Inference	16.5±0.5 [16.2, 16.8]	27.6±0.1 [27.5, 27.6]
Standard RAG	17.6±0.0 [17.6, 17.6]	41.0±0.0 [41.0, 41.0]
IRCoT	21.6±0.0 [21.6, 21.6]	33.7±0.0 [33.7, 33.7]
Search-o1	18.7±0.5 [18.4, 19.0]	39.5±0.1 [39.4, 39.5]
β-GRPO	51.7±0.5 [51.4, 52.0]	56.4±0.1 [56.3, 56.4]
Search-R1	42.4±0.0 [42.4, 42.4]	54.6±0.0 [54.6, 54.6]
HiPRAG	45.9±0.5 [45.6, 46.2]	58.0±0.1 [57.9, 58.0]
TreePS-RAG	47.2±0.0 [47.2, 47.2]	57.8±0.0 [57.8, 57.8]
AutoSearch	44.8±0.0 [44.8, 44.8]	56.3±0.0 [56.3, 56.4]
ALAR	48.3±0.5 [48.0, 48.6]	57.4±0.2 [57.2, 57.5]
LPR-RAG (ours)	46.4±0.8 [45.8, 47.0]	59.3±0.1 [59.2, 59.4]

C Full Ablation Results

Table 6 lists all 45 ablation variants, together with the full 3B/7B × PPO/GRPO extension grid, under the same training and evaluation protocol as Table 1, including those summarized in Table 3. Initialization-exploration variants probe the shared initialization, action-entropy settings, and branch ratios; audit rows repeat the main configuration with repeated rollouts or with additional human and judge audits.

D Training and Inference Cost

Table 7 reports training and inference cost. LPR-RAG's training total includes the counterfactual branch rollouts and the Stage-II selector collection, so its training-time retrieval calls exceed

Table 6: All ablation variants under the full evaluation protocol: CEM (%) and average search calls.

Family	Variant	CEM↑	AvgSearch↓
Counterfactual supervision	outcome rewards only	54.4	2.3
	$M=1$ paired branches (main)	59.3	1.6
	$M=3$ paired branches	59.5	1.6
	abstain on both-wrong (main)	59.3	1.6
	F1 soft preference	58.4	1.6
	terminal continuation (main)	59.3	1.6
	short horizon	57.7	1.5
	no branch replay (main)	59.3	1.6
	branch replay into execution	58.7	1.6
Selective refinement	direct gate	59.0	1.7
	equal-parameter MLP	59.0	1.7
	equal-FLOPs MLP	59.1	1.7
	uniform two-step CE	59.1	1.7
	hard-only (no easy consistency)	59.2	1.6
	selective hard/easy (main)	59.3	1.6
	no context cross-attention	59.1	1.7
Depth training	fixed depth 0	59.0	1.7
	fixed depth 1	59.0	1.6
	fixed depth 2	59.1	1.6
	joint dynamic depth	59.0	1.6
	static depth imitation (main)	59.3	1.6
	offline depth oracle	59.5	1.6
Initialization & exploration	raw instruct checkpoint	57.4	1.6
	shared outcome+format init	58.8	1.7
	entropy low	58.8	1.7
	entropy main	59.3	1.6
	entropy high	59.2	1.6
	branch ratio $\rho=0.10$	58.6	1.7
	branch ratio $\rho=0.25$ (main)	59.3	1.6
	branch ratio $\rho=0.50$	59.2	1.6
Extensions & audits	Qwen2.5-7B + PPO	62.1	1.6
	Qwen2.5-7B + GRPO	62.7	1.6
	Qwen2.5-3B + GRPO	59.9	1.6
	strong retriever	60.6	1.5
	retriever shift	55.1	1.7
	corpus shift	53.9	1.8
	grounded-answer setting	57.1	1.7
	$M=3$ stability audit	59.4	1.6
	human preference audit	59.4	1.6
	judge audit	59.5	1.6
Causal checks	zeroed updater output	58.9	1.7
	swapped update vectors	58.6	1.8
	shuffled depth labels	59.0	1.7
	non-shared iterations	59.1	1.7
Attribution	decision pathway only	58.2	1.6
	execution pathway only	54.2	2.2

HiPRAG's, while its test-time AvgSearch is lower. Equal-compute variants of HiPRAG and TreePS-RAG are given training budgets that bracket LPR-RAG's total from below and above. ALAR additionally requires teacher trajectories and post-training, which dominate its cost. At inference, latent refinement adds less than one percent to end-to-end latency.

Table 7: Training and inference cost. Token counts are millions; retrieval calls are thousands; inference latency is the macro average over the seven datasets (first 100 examples each, single GPU).

Method	Rollout	Branch	Teacher	Total	Retr. (k)	Infer. (ms)
Search-R1	120	–	–	220	452	810
HiPRAG	124	–	–	216	378	775
HiPRAG (equal-compute)	156	–	–	272	462	775
TreePS-RAG	138	–	–	238	390	796
TreePS-RAG (equal-compute)	180	–	–	310	498	796
ALAR	128	–	25	264	384	737
LPR-RAG (ours)	138	46	–	297	388	757

E PER-DATASET LATENCY AND PARSE RATES

Table 8 reports per-dataset end-to-end latency over the first 100 examples of each dataset on a single GPU, together with LPR-RAG trajectory parse rates under the official parser. Latency is highest on multi-hop datasets for both methods, and LPR-RAG is faster than HiPRAG on every dataset, reflecting its fewer search calls; the latent refinement overhead is far smaller than the retrieval savings.

Table 8: Per-dataset end-to-end latency (ms) and LPR-RAG parse rate (%).

	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle
HiPRAG latency	668	631	615	823	875	952	861
LPR-RAG latency	648	612	596	802	856	933	852
LPR-RAG parse rate	97.6	96.9	96.5	95.2	95.6	94.1	94.4

F ADDITIONAL ANALYSES

Answer dependency. Partitioning audited states by whether the non-search branch alone yields a correct answer, single-hop questions are answered from parametric knowledge noticeably more often than multi-hop questions, consistent with the knowledge-aware setting: for some states `NONSEARCH` is genuinely the better action. This supports treating retrieval necessity as state-dependent rather than dataset-level.

Bridge versus comparison questions. On HotpotQA, LPR-RAG's improvement over HiPRAG concentrates on bridge questions, while comparison questions are essentially tied; on 2WikiMultihopQA both question types improve. Refinement thus helps most where evidence must be gathered sequentially.

Search-branch query success. Among audited search branches, branches whose first query retrieves supporting evidence end correct far more often than branches whose first query misses. A substantial share of first queries retrieves support, and multi-hop questions show more recovery after an initial miss, reflecting multi-turn retrieval.

G CASE STUDIES

Table 9 summarizes the four case-study types. Each row corresponds to one audited state with its initial action distribution, chosen depth, refined distribution, and paired branch outcomes. The over-search case corrects a redundant search toward `NONSEARCH`; the under-search case corrects toward `SEARCH`; the safe no-op case keeps a correct initial decision unrefined; and the abstention case shows a state where both branches fail and no preference is emitted, illustrating the exploration ceiling discussed in Section 4.4.

Table 9: Case studies: initial gate probability of `SEARCH`, chosen depth k, refined probability, and branch outcomes.

Case	Difficulty	$p_0(\texttt{S})$	k	$p_k(\texttt{S})$	S-branch	N-branch
Over-search correction	hard	0.7	1	0.2	correct	correct
Under-search correction	hard	0.3	2	0.8	correct	wrong
Safe no-op	easy	0.2	0	0.2	correct	correct
Abstention (exploration ceiling)	hard	0.6	2	0.5	wrong	wrong