R3PO: UNIFIED REASONER-RETRIEVER REINFORCEMENT POLICY OPTIMIZATION FOR AGENTIC RAG

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic retrieval-augmented generation (RAG) systems depend on a reasoner and a retriever, yet a schism in their optimization—reinforcement learning (RL) for the reasoner versus indirect signals for the retriever—hinders true end-to-end alignment with task outcomes. We introduce **R**easoner-**R**etriever **R**einforcement **P**olicy **O**ptimization (R3PO), a staged framework that applies a common, outcome-driven policy optimization interface to both reasoner and retriever actions. Our approach is built on three pillars: **principled policy construction**, which defines a stochastic policy that respects the geometry of the pretrained retriever's native scoring function; **principled adaptation**, which uses a unified RL objective to refine both components with trajectory-level rewards; and **principled generalization**, which confronts the critical "candidate-to-corpus chasm." We prove that adapting a retriever on a small candidate pool makes it blind to score shifts that are catastrophic for full-corpus retrieval. R3PO solves this by introducing background documents as anchors during training, grounding the learned policy and ensuring it generalizes beyond the candidate set. This preserves the retriever's efficient inner-product structure, enabling direct, full-corpus deployment. Across seven QA benchmarks and three backbones, R3PO consistently achieves **higher accuracy with greater efficiency**, delivering relative gains of 2.2%-3.5% in average exact match (EM) while even enhancing an already agent-adapted retriever. These results demonstrate that a principled, unified RL approach unlocks significant performance gains.

1 INTRODUCTION

Large language models (LLMs) are evolving into autonomous agents that interleave reasoning with actions like external search (Jin et al., 2025; Chen et al., 2025; Song et al., 2025). This paradigm, agentic RAG, depends on the synergy between a **reasoner** that generates queries and a **retriever** that selects evidence. However, a fundamental schism exists in how these components are optimized: while the reasoner can be directly aligned with task outcomes via RL, the retriever is often adapted using indirect signals or bypassed entirely. This schism prevents the realization of a truly end-to-end optimized, deployable agentic RAG system.

Current adaptation strategies are built on incomplete paradigms that fail to resolve this core issue. The first paradigm relies on **creating proxy supervisory signals**. Methods like Agentic-R (Liu et al., 2026) and LRAT (Zhou et al., 2026) convert downstream outcomes into contrastive pairs or labels, breaking the end-to-end RL loop. This transforms the problem into a supervised task that relies on brittle annotation pipelines (e.g., LLM annotators) and optimizes for an intermediate proxy, not holistic trajectory success. The second paradigm **bypasses the native retriever** with auxiliary rankers (Zeng et al., 2026; Wang et al., 2018). While effective in reranking, this approach abandons improving the first-stage dense retriever itself, sacrificing the key advantage of a single, efficient, deployable full-corpus index. A third, more direct paradigm applies RL to the native retriever's scoring function (Zhang et al., 2026; Wei et al., 2026). While a conceptual step forward, these methods operate on the critical, unexamined assumption that optimization on a small candidate set will generalize to a full-corpus index. This **candidate-to-corpus chasm** is a fundamental barrier; without a principled mechanism for generalization, any RL adaptation remains unreliable and risks catastrophic performance degradation. The field has thus reached an impasse: existing methods are either indirect, non-deployable, or lack a guarantee of generalization.

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.

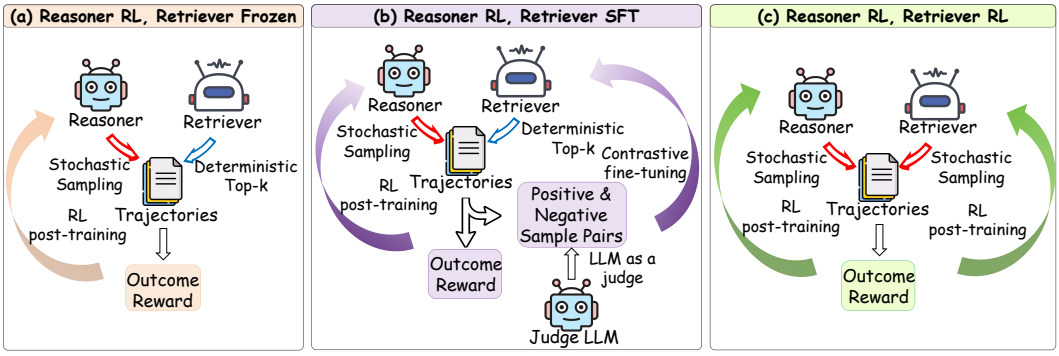

Figure 1: **Unifying LLM and retriever post-training.** (a) RL trains the reasoner while retrieval is frozen. (b) The reasoner uses RL while the retriever uses a separate supervised contrastive procedure. (c) R3PO exposes both pretrained models as stochastic policies: the reasoner selects tokens and the retriever selects documents. Both use the **same rollout–outcome-evaluation–GRPO workflow**, applied successively to the reasoner and retriever.

This paper argues that a new framework is not just beneficial, but essential to move forward. We introduce **R**easoner-**R**etriever **R**einforcement **P**olicy **O**ptimization (R3PO), a principled framework conceived from the ground up to resolve this schism and enable true end-to-end adaptation and deployment. R3PO is built on three pillars that directly confront the limitations of prior work:

1. **Principled Policy Construction**: Instead of treating the retriever as a generic model, we start from first principles. We prove that an InfoNCE-pretrained retriever's score encodes a specific **density-ratio**, and derive that the only policy structure preserving its native geometry is a **positive affine transformation** (Theorem 1). This provides a theoretically grounded "native policy" that respects, rather than replaces, the powerful priors of pretraining.
2. **Principled Adaptation**: R3PO rejects a separate passage-labeling pipeline, favoring a direct, end-to-end feedback loop. We introduce a unified **Group Relative Policy Optimization (GRPO) interface** that applies **completed-trajectory rewards** directly to reasoner and retriever actions. The update is framed as a **relative correction** to the pretrained policy (Section 5), ensuring adaptation is a refinement of existing knowledge, not a relearning process.
3. **Principled Generalization**: Critically, and what sets R3PO apart from all prior work, we identify and solve the candidate-to-corpus chasm. Through a **candidate-conditioned identifiability analysis** (Theorem 2), we prove why training on hard negatives alone is insufficient. Our solution—augmenting the candidate pool with **uniform background documents** (Section 6)—acts as a theoretical and practical bridge, creating "cross-pool" comparisons that constrain score drift and ensure the learned policy generalizes to the full corpus. This is validated by experiments showing performance scales to a **rebuilt, full-corpus index** (Appendix D.3).

Thus, R3PO offers a complete and theoretically-grounded framework for the end-to-end adaptation of dense retrievers in agentic systems. It moves beyond the limitations of proxy supervision and auxiliary modules, and critically, provides the first principled solution to the candidate-to-corpus generalization problem. Our results demonstrate **higher accuracy and greater efficiency**, showing consistent relative gains of 2.2%-3.5% in average EM across diverse backbones—including an already agent-adapted one. This demonstrates that by resolving these fundamental roadblocks, we can unlock reliable performance gains in real-world agentic RAG.

2 Related Work

Paradigms and Pitfalls in Agentic Retriever Adaptation. The evidence-selection bottleneck in agentic RAG motivates direct retriever adaptation, but current paradigms are incomplete. As reviewed thoroughly in Appendix A, existing methods typically rely on either indirect supervision via proxy signals (Liu et al., 2026), auxiliary rerankers that sacrifice deployability (Zeng et al., 2026), or direct RL applications that lack a principled generalization mechanism (Zhang et al., 2026). R3PO builds on this third, more direct paradigm but addresses its critical limitations head-on.

Respecting Pretrained Retriever Structure. The powerful priors in a pretrained retriever should be refined, not discarded. InfoNCE pretraining (Oord et al., 2018) gives the retriever's scores

meaning related to density-ratios (Ma & Collins, 2018; Durkan et al., 2020). Several works, including Retrieval-GRPO (Liu et al., 2025), SSR-GRPO (Song et al., 2026), and PAO (Wei et al., 2026), implicitly respect these priors, framing adaptation as a KL-regularized correction to a reference policy. R3PO makes this connection explicit and foundational. We ask a structural question: which policy maps preserve the essential geometry (ranking and offset invariance) of a contrastively trained scorer? Our analysis (Theorem 1) shows this is the class of **positive affine transformations**, elevating the "native policy" choice from an implementation detail to a theoretically-grounded principle.

The Candidate-to-Corpus Generalization Chasm. Training a retriever on a restricted candidate set—common to nearly all prior adaptation work (e.g., HARR, PAO, Agentic-R (Zhang et al., 2026; Wei et al., 2026; Liu et al., 2026))—presents a fundamental, unaddressed risk. A policy can learn score adjustments effective *within* the candidate pool but drift catastrophically relative to the vast number of documents outside it. We formally analyze this problem through **candidate-conditioned identifiability** (Theorem 2), proving such "cross-pool" score shifts are invisible to a standard KL-regularization objective. R3PO is the first to identify this chasm and propose a principled solution: augmenting the hard-negative pool with **uniform background documents**. These documents act as anchors, creating "cross-pool" comparisons necessary to constrain score drift and ensure the learned policy generalizes from the training pool to a rebuilt, full-corpus index at deployment.

3 THE R3PO FRAMEWORK

The central challenge in agentic RAG is adapting a pretrained dense retriever to task outcomes without sacrificing the efficient, inner-product structure that enables full-corpus deployment. We formalize this within an agentic system where a **reasoner** (LLM) and a **retriever** interact to produce a trajectory ξ for a given question x:

$$\xi = (x, a_1^{\text{res}}, \ldots, a_{t_1}^{\text{res}}, q_1, D_1, a_{t_1+1}^{\text{res}}, \ldots, q_2, D_2, \ldots, \hat{a}), \tag{1}$$

This trajectory exposes a fundamental dichotomy. The reasoner's token actions (a^{res}) are governed by a standard probabilistic policy $\pi_\theta(a_t^{\text{res}}|a_{<t})$, making them directly amenable to reinforcement learning. In contrast, the retriever typically uses its native score function $s_\phi(q, d)$ to return a deterministic set of documents, $\text{TopK}_{d\in\mathcal{D}} s_\phi(q, d)$, which provides no action probabilities for optimization.

The core task is therefore to construct, adapt, and deploy a **stochastic retriever policy** that resolves this issue. Reasoner-Retriever Reinforcement Policy Optimization (R3PO) addresses this systematically through three interconnected pillars. First, **Principled Policy Construction** (Section 4) answers the foundational question of defining a stochastic policy that respects the geometry of pretrained scores. With a native policy, **Principled Adaptation** (Section 5) then addresses refining this policy using outcome-driven rewards from completed trajectories. Finally, and critically, **Principled Generalization** (Section 6) solves the candidate-to-corpus chasm by ensuring these policy updates, learned on a small candidate set, generalize reliably to the full corpus at deployment. This structured approach systematically bridges the gap from a deterministic, black-box retriever to a fully integrated and deployable component of an end-to-end optimized agent.

4 PILLAR 1: PRINCIPLED POLICY CONSTRUCTION

Rather than attaching a new head to the retriever's representations, we ask what its scores already encode and let that answer determine the policy.

What Contrastive Similarity Encodes. To construct a principled policy, we first analyze what a retriever's native scores, learned via contrastive pretraining, actually represent. InfoNCE (Oord et al., 2018) frames learning as a classification task: it trains a scorer to identify the single positive document, drawn from the data distribution $p_{\text{data}}(d|q)$, from a set of $M-1$ negatives drawn from a sampling distribution $\nu(d|q)$. The optimal scorer learns to compute the log-odds of a document being the true positive. As derived in Appendix B.1, this leads to a score function s^* that is proportional to the log-ratio of these two densities:

$$\frac{s^*(q, d)}{\tau} = \log \frac{p_{\text{data}}(d \mid q)}{\nu(d \mid q)} + b(q), \tag{2}$$

where τ is the contrastive temperature and $b(q)$ is an arbitrary offset that is constant for all documents d given a query q (Ma & Collins, 2018; Durkan et al., 2020).

This identity reveals two properties fundamental to our policy design. The presence of the arbitrary, query-wise offset $b(q)$ implies that absolute score values are not identified by the pretraining objective; in other words, **pretraining only identifies score differences**. Consequently, any policy built upon

Table 1: Correspondence between LLM and dense-retriever pretraining. The distribution-recovery row describes idealized population optima; its retriever equality additionally requires uniform negatives. It does not assert calibration of released checkpoints.

	Autoregressive LLM	Dense retriever
Prediction task	Next token a_t given $a_{<t}$	Document d given query q
Idealized distribution recovery	$p_{\mathrm{data}}(a_t \mid a_{<t}) = \mathrm{softmax}_{a\in\mathcal{V}}\, z^*(a_{<t}, a)$	$p_{\mathrm{data}}(d \mid q) = \mathrm{softmax}_{d\in\mathcal{D}}(s^*(q,d)/\tau)$
Logit passed to softmax	$z_\theta(a_{<t}, a)$	$s_\phi(q,d)/\tau$
Score function	$z_\theta(a_{<t}, a) = H_\theta(a_{<t})^\top W_a$	$s_\phi(q,d) = E_\phi(q)^\top E_\phi(d)$
Pretraining objective	Next-token cross-entropy	InfoNCE

these scores must exhibit **offset invariance**: its behavior cannot change if a constant is added to all document scores for a given query. This principle directly shapes the class of valid policy maps we consider. Furthermore, the core term $\log(p_{\mathrm{data}}/\nu)$ establishes a meaningful ranking that our policy should respect. This leads to the insight that **scores are log-density-ratios, not log-probabilities**. The distinction is critical: only in the special case of uniform negative sampling does the denominator $\nu(d|q)$ become a constant that can be absorbed into the offset $b(q)$. In that scenario, the score $s^*(q,d)$ becomes proportional to $\log p_{\mathrm{data}}(d|q)$, and applying a softmax function would recover the true data distribution. However, since modern retrievers often use non-uniform "hard" negatives, the score must correctly down-weight documents that are common negatives to remain discriminative. This motivates the principle of **order preservation**: our policy should, by default, respect the preference ordering established by the pretrained model, as it reflects these powerful, learned priors. R3PO is therefore designed around a policy structure that is both offset-invariant and order-preserving, directly respecting the geometry encoded by contrastive pretraining.

From Scores to a Native Document Policy. To transform the retriever's deterministic scores into a stochastic policy, we first require a candidate set of documents $C \subseteq \mathcal{D}$ for a given query q. This set could be the entire corpus or a smaller training pool. Rather than adding an auxiliary ranking head, we construct our policy by applying a shared scalar map $g : \mathbb{R} \to \mathbb{R}$ directly to the native scores $s_\phi(q,d)$. This map converts scores into logits, which are then scaled by a temperature $T > 0$ and normalized via softmax:

$$p^g_{\phi,T}(d|q,C) = \frac{\exp(g(s_\phi(q,d))/T)}{\sum_{u\in C}\exp(g(s_\phi(q,u))/T)}, \quad d \in C. \tag{3}$$

The central question is: what class of functions g preserves the geometric structure encoded by contrastive pretraining? To answer this, we formalize these principles as two requirements on g. First, **offset invariance**: the policy must be invariant to a constant score shift for a given query. Second, **order preservation**: g must strictly increase to respect the pretrained model's preference ranking.

Theorem 1 (**Characterization within shared scalar maps**). *Fix any $T > 0$. A shared pointwise map g satisfies offset invariance on all finite score vectors and strict order preservation if and only if*

$$g(u) = \kappa u + a, \quad \kappa > 0, a \in \mathbb{R}. \tag{4}$$

This result is intuitive: a score gap must carry the same probabilistic meaning regardless of the absolute score values. Shifting two scores by the same constant cannot change their relative probability. Formally, offset invariance implies additivity ($g(u+c) - g(v+c) = g(u) - g(v)$), which, combined with strict monotonicity, restricts g to be an affine function (see Appendix B.2 for a full proof). The theorem thus elevates an affine map from a convenient default to a structural consequence, fixing the function's shape but leaving its scale κ free.

Corollary 1 (**Pretraining-calibrated scale**). *Under Eq.* (3) *with uniform negatives, require full-corpus normalization at $T = 1$ to recover every strictly positive, nonuniform pretraining conditional. Among the maps of Theorem 1, this condition fixes $\kappa = 1/\tau$, giving the R3PO policy*

$$p_\phi(d|q,C) = \frac{\exp(s_\phi(q,d)/(\tau T))}{\sum_{u\in C}\exp(s_\phi(q,u)/(\tau T))}, \quad d \in C. \tag{5}$$

This corollary, proven in Appendix B.2, provides a principled initialization for the policy and illuminates the deep analogy between dense retrievers and autoregressive LLMs, summarized in Table 1. For the retriever, the softmax logit is $s_\phi(q,d)/\tau$, directly analogous to the token logits z_θ in an LLM. The native score function $s_\phi(q,d)$, an inner product, plays the role of the LLM's final linear layer producing logits from hidden states. Under idealized pretraining conditions, the retriever's policy recovers the data distribution $p_{\mathrm{data}}(d|q)$ via a softmax, just as an LLM recovers the next-token distribution. The corollary establishes this calibration as a principled **initialization convention** for the

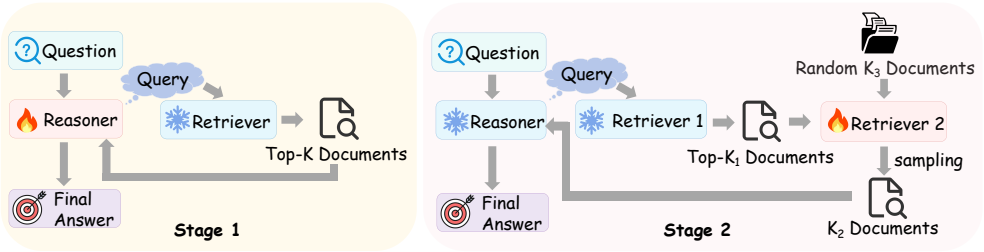

Figure 2: **Two-stage R3PO training.** Stage I trains the reasoner with a frozen retriever. Stage II freezes the reasoner; Retriever 1 combines Top-K_1 retrieval results with K_3 randomly sampled corpus documents to form C_1, and the trainable Retriever 2 re-encodes the query and candidates and samples $K_2 = 1$ document per search for training. After Stage II, the adapted encoder re-encodes the full corpus and its index is rebuilt; evaluation returns Top-3 passages.

score scale τ, not a claim that $T = 1$ is optimal for the downstream task. When a model's pretraining temperature τ is undocumented, we treat it as a hyperparameter representing the base scale.

Together, Theorem 1 and Corollary 1 define a "native" policy for a contrastively trained retriever. Its log-odds are simply the pretrained score differences scaled by the effective temperature τT, and it is implemented using the original retriever encoder without modification. While our characterization does not rule out more complex policy structures (Appendix B.2), we adopt this native policy because it directly reuses the powerful priors from pretraining and, crucially, preserves the standard inner-product scoring interface required for efficient, full-corpus deployment.

5 Pillar 2: Principled Adaptation

With a native policy in place, R3PO improves it using task success while preserving pretrained priors. We treat the reasoner and retriever as policies under a unified interface, driven by completed-trajectory rewards and applied in two sequential stages (Figure 2). Stage I post-trains the reasoner with a frozen retriever. Stage II then freezes the trained reasoner to post-train the retriever. This shared interface (Figure 1(c)) aligns both components with the same end-to-end objective.

Adaptation as a Relative Correction. We frame adaptation as a principled **correction** to the initial policy $p_0 = p_{\phi_0}$, from a frozen retriever checkpoint. Given a query and candidate set, the trained policy p_ϕ is an explicit reweighting of the initial policy. Let $\Delta_\phi(d) = [s_\phi(q, d) - s_{\phi_0}(q, d)]/(\tau T)$ be the change in a document's normalized score. The policy update has the exact form:

$$p_\phi(d|q, C) = \frac{p_0(d|q, C)e^{\Delta_\phi(d)}}{\mathbb{E}_{u\sim p_0(\cdot|q,C)}e^{\Delta_\phi(u)}}. \tag{6}$$

The numerator applies a learned multiplicative correction $e^{\Delta_\phi(d)}$ to the initial probability, while the denominator normalizes. This makes adaptation purely relative: only score changes differing from the average affect the final distribution. This identity uses the actual checkpoint as reference, not an idealized distribution, making the KL penalty below a tool for stable refinement not relearning.

This perspective directly connects to standard KL-regularized policy improvement. The policy p^* that maximizes $\mathbb{E}_p Q - \beta_Q D_{\mathrm{KL}}(p||p_0)$ for a utility function $Q(d)$ is $p^*(d) \propto p_0(d)e^{Q(d)/\beta_Q}$ (Peters et al., 2010). This reveals that our score change $\Delta_\phi(d)$ serves as a learned proxy for the scaled utility $Q(d)/\beta_Q$. In log-odds space, this update is additive:

$$\log \frac{p^*(d)}{p^*(u)} = \frac{s_{\phi_0}(q, d) - s_{\phi_0}(q, u)}{\tau T} + \frac{Q(d) - Q(u)}{\beta_Q}. \tag{7}$$

Equation (7) shows how outcome-based utility corrects the initial preferences encoded in the pretrained scores. Documents with equal utility preserve their original log-odds, while a utility advantage can strengthen or reverse an initial preference, thus refining the model's priors with task-specific rewards. Appendix B.3 provides a formal derivation.

The KL penalty is applied within the sampled candidate sets, but it constrains the sampler's overall behavior. This scope is formalized by considering a distribution μ over candidate sets, which reflects the training process where different sets may be drawn for the same query.

Proposition 1 (Scope of the reference constraint). *Fix q and a common candidate-set law μ. Define the expected conditional divergence $\mathcal{K}_\mu = \mathbb{E}_{C\sim\mu} D_{\mathrm{KL}}(p_\phi^C||p_0^C)$ and the document marginal*

$\overline{p}_\phi(d) = \mathbb{E}_{C\sim\mu}[1\{d \in C\}p_\phi^C(d)]$. *Then, using natural logarithms,*

$$D_{\mathrm{KL}}(\overline{p}_\phi||\overline{p}_0) \le \mathcal{K}_\mu \quad \textit{and} \quad ||\overline{p}_\phi - \overline{p}_0||_{\mathrm{TV}} \le \sqrt{\mathcal{K}_\mu/2}, \tag{8}$$

where total variation is half the sum of absolute probability differences.

This proposition (proven in Appendix B.4) shows that controlling the average KL divergence within candidate sets provides an upper bound on the change to the marginal document probabilities. In essence, it ensures that local refinements do not cause uncontrolled shifts in the sampler's global preferences. However, it is crucial to note that this sampling marginal $\overline{p}_\phi$ is not the same as a policy derived from normalizing scores over the entire corpus. This distinction highlights the candidate-to-corpus chasm, the very problem that Pillar 3 addresses.

A Unified GRPO Interface for Token and Document Actions. We now derive a unified Group Relative Policy Optimization (GRPO) interface to adapt both the reasoner and the retriever. This framework optimizes both token and document selection policies using a shared, trajectory-level reward signal. For each question, we sample G trajectories from the behavior policy. All optimized actions in trajectory i share $\hat{A}_i$: its reward minus the group mean, divided by the group standard deviation (zero for constant-reward groups), following GRPO (Shao et al., 2024). The objectives for both policies use a clipped surrogate update, $l(\rho, A) = \min\{\rho A, \mathrm{clip}(\rho, 1-\epsilon, 1+\epsilon)A\}$, where A denotes the trajectory advantage $\hat{A}_i$ and ρ is the likelihood ratio of an action's probability under the current policy versus the behavior policy (π_{old}). Ratios for LLM token actions ($a_{i,t}$) and retriever document actions ($d_{i,j}$) are:

$$\rho_{i,t}^{\mathrm{res}}(\theta) = \frac{\pi_\theta(a_{i,t}|a_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(a_{i,t}|a_{i,<t})}, \quad \rho_{i,j}^{\mathrm{ret}}(\phi) = \frac{\pi_\phi^{\mathrm{ret}}(d_{i,j}|q_{i,j}, C_{i,j})}{\pi_{\phi_{\mathrm{old}}}^{\mathrm{ret}}(d_{i,j}|q_{i,j}, C_{i,j})} \tag{9}$$

Here, $\pi_\phi^{\mathrm{ret}} \equiv p_\phi$ is the native policy from Eq. (6) and $C_{i,j}$ is the realization of $C_1(q_{i,j})$ combining hard-pool and uniform background documents.

Let L_i count tokens in the full trajectory context, N_i the generated tokens ($m_{i,t} = 1$), and M_i document actions; $a_{i,<t}$ includes the prompt and retrieved observations. The objectives for the reasoner ($\mathcal{J}^{\mathrm{res}}$) and retriever ($\mathcal{J}^{\mathrm{ret}}$) combine the clipped advantage with a KL-divergence penalty against a frozen reference policy (π_{ref}) from the stage start ($\pi_{\mathrm{ref}}^{\mathrm{ret}} = p_0$). Expectations average over questions, rollouts, and candidate sampling.

$$\mathcal{J}^{\mathrm{res}}(\theta) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{\max(1,N_i)}\sum_{t=1}^{L_i} m_{i,t}\Big(l(\rho_{i,t}^{\mathrm{res}}, \hat{A}_i) - \beta D_{\mathrm{KL}}[\pi_\theta(\cdot|a_{i,<t})||\pi_{\mathrm{ref}}(\cdot|a_{i,<t})]\Big)\Bigg]$$

$$\mathcal{J}^{\mathrm{ret}}(\phi) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{\max(1,M_i)}\sum_{j=1}^{M_i}\Big(l(\rho_{i,j}^{\mathrm{ret}}, \hat{A}_i) - \beta_{\mathrm{ret}} D_{\mathrm{KL}}[\pi_\phi^{\mathrm{ret}}(\cdot|q_{i,j}, C_{i,j})||\pi_{\mathrm{ref}}^{\mathrm{ret}}(\cdot|q_{i,j}, C_{i,j})]\Big)\Bigg]$$

The two objectives share the clipped update and reference regularization, differing only in the action space they govern. This unified structure directly applies outcome-driven rewards to the retriever's native scoring function, fulfilling a core goal of R3PO.

Completed-Trajectory Reward. The reward signal for retriever adaptation in Stage II must address a key challenge: with a frozen reasoner, final-answer correctness is a noisy signal for retrieval quality. The retriever might provide the correct evidence, yet receive no credit if the reasoner fails to use it. To create a more stable learning signal, we design a composite reward that combines three signals from each completed trajectory ξ_i.

We combine a primary signal for final answer correctness (R_{ans}) with a direct reward for evidence acquisition (R_{doc}), and a penalty for search inefficiency. Let $D(\xi_i)$ be the set of documents retrieved in trajectory i and $N_{\mathrm{search}}(\xi_i)$ be its search count, and a^* the gold answer. The reward components are defined as:

$$R_{\mathrm{ans}}(\xi_i) = 1\{\mathrm{EM}(\hat{a}_i, a^*) = 1\}, \quad R_{\mathrm{doc}}(\xi_i) = 1\{\exists d \in D(\xi_i) : \mathrm{contains}(d, a^*)\} \tag{10}$$

The total trajectory reward is a weighted combination that encourages both correctness and efficiency:

$$R(\xi_i) = R_{\mathrm{ans}}(\xi_i) + \lambda_{\mathrm{doc}} R_{\mathrm{doc}}(\xi_i) - \lambda_{\mathrm{search}} N_{\mathrm{search}}(\xi_i). \tag{11}$$

Hyperparameters λ_{doc} and $\lambda_{\mathrm{search}}$ are given in Appendix C. This reward design has two crucial properties. First, all components are derived from the completed trajectory, maintaining an end-to-end optimization loop that avoids external labels or learned reward models. Second, the reward is applied at the trajectory level. This means every document action within a trajectory receives the same

advantage in the GRPO update, directly linking the retriever's policy to the holistic success of the agent's strategy. The 'contains' function serves as a simple but effective proxy for evidence utility. Appendix D.1 reports the reward-component ablation.

6 PILLAR 3: PRINCIPLED GENERALIZATION FROM CANDIDATES TO CORPUS

Retriever RL is typically trained on a small candidate set of hard negatives, creating a critical generalization risk. A policy can learn a uniform score shift across this pool that, while preserving internal rankings, is invisible to candidate-conditioned objectives. Relative to the full corpus, however, this drift can cause irrelevant background documents to outrank the entire hard-negative set at deployment. For example, shifting hard-pool logits from $(3, 2)$ to $(-1, -2)$ leaves the within-pool softmax unchanged, yet an outside document at 0 now ranks first.

What Candidate-conditioned Regularization Can Identify. The core problem with training on a fixed candidate set is that the standard KL-regularization objective is blind to certain catastrophic score changes. The penalty only constrains score adjustments within the candidate set, but it cannot detect a uniform shift of the entire set relative to the vast number of documents outside it. This creates a critical generalization gap. We can formalize what the training objective can and cannot "see" by defining a comparison graph where documents are connected if they ever co-occur in a candidate set. The following theorem proves that score changes are only identifiable within the connected components of this graph. Let f_0 and f be the reference and current logits (scores divided by τT), and let μ be the distribution over candidate sets.

Theorem 2 (Candidate-conditioned identifiability). *Fix q, finite logits f_0, f on a finite corpus, and a fixed law μ over nonempty candidate sets. Connect u, v when $Pr_{C\sim\mu}(u, v \in C) > 0$, and let $p_f^C = softmax_C(f)$, with P_f denoting full-corpus normalization. Then $\mathbb{E}_{C\sim\mu} D_{\mathrm{KL}}(p_f^C || p_{f_0}^C) = 0$ if and only if $f - f_0$ is constant on each connected component of the comparison graph. Training can therefore detect relative score changes along observed comparisons but cannot detect a common shift of an entire disconnected group. In particular, for a proper nonempty hard-only pool B, the shift $f_c = f_0 + c\mathbf{1}_B$ is invisible to its conditional policy and to the reference penalty computed on that pool, even though it changes the pool's standing in the full corpus. Appendix B.5 proves the theorem and shows that the excluded mass $P_{f_c}(\mathcal{D} \setminus B) \to 1$ as $c \to -\infty$.*

In essence, Theorem 2 proves that if the training pool of hard negatives is disconnected from the rest of the corpus, the objective is mathematically incapable of penalizing a score shift applied uniformly to that pool. This makes generalization to the full corpus unreliable and motivates our solution below.

Background Documents as Anchors. The preceding analysis revealed that training on an isolated hard-negative pool makes the policy blind to uniform score shifts, risking catastrophic generalization failure. To solve this candidate-to-corpus chasm, R3PO's training procedure introduces **background documents** as anchors, ensuring the policy generalizes. This is implemented in two steps within each Stage-II search action. First, a candidate set $C_1(q)$ is constructed by combining a **hard pool** of K_1 documents from the frozen retriever with K_3 uniformly sampled **background documents**.

$$C_1(q) = \mathrm{Top}_{K_1}(q) \cup \mathrm{Uniform}_{K_3}(q) \tag{12}$$

Second, the trainable retriever policy performs a stochastic action by sampling K_2 documents from this combined set $C_1(q)$ to pass to the reasoner. This sampling of K_2 documents (where $K_2 = 1$ in our experiments) is the explicit action optimized by the GRPO objective.

The background documents provide a stable reference. A score shift applied only to the hard pool is invisible to a KL penalty computed on that pool alone (Theorem 2). However, by forcing the policy to select from a mixed set containing anchors, any such shift alters the scores of hard documents relative to the background documents. This change becomes visible to the KL objective, which can then penalize it. To second order, the KL divergence's sensitivity to such a shift c is:

$$D_{\mathrm{KL}}(p_{f_c}^C \| p_{f_0}^C) \approx \tfrac{1}{2}\alpha(1-\alpha)c^2, \quad \text{where } \alpha = p_{f_0}^C(\mathrm{Top}_{K_1}) \tag{13}$$

The visibility of the shift, captured by $\alpha(1-\alpha)$, is zero when training is confined to hard negatives ($\alpha = 1$). By introducing background documents, we ensure $\alpha < 1$, making the objective sensitive to cross-pool drift. This anchoring mechanism connects the comparison graph (Appendix B.6), providing the theoretical bridge between candidate-set optimization and full-corpus deployment.

Full-corpus Deployment. After training, we deploy the adapted model by re-encoding the entire corpus and rebuilding the dense index. This direct deployment is enabled by our policy construction

Table 2: EM (%) on single and multi-hop QA. Rows pair a frozen retriever with its policy-post-trained counterpart for the same reasoner. Bold indicates the better result within each pair. Parentheses show relative gains over the paired baseline, computed from dataset scores before rounding their averages.

	Single-hop QA				Multi-hop QA					
Method	NQ	TriviaQA	PopQA	Avg.	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.	Overall Avg.
SEARCH-R1 + E5	**50.1**	65.7	46.7	54.2	47.8	46.0	22.3	47.2	40.8	46.5
R3PO + E5	**50.1**	**65.9**	**49.1**	**55.0** (+1.6%)	**49.3**	**47.5**	**23.6**	**48.8**	**42.3** (+3.6%)	**47.8** (+2.6%)
SEARCH-R1 + Agentic-R	48.8	**67.1**	46.8	54.2	48.9	45.9	**22.8**	50.4	42.0	47.2
R3PO + Agentic-R	**49.1**	**67.1**	**48.5**	**54.9** (+1.2%)	**50.4**	**47.2**	**22.8**	**52.8**	**43.3** (+3.1%)	**48.3** (+2.2%)
SEARCH-R1 + Qwen-0.6B	46.4	64.1	47.3	52.6	46.6	47.4	19.2	41.6	38.7	44.7
R3PO + Qwen-0.6B	**47.5**	**64.5**	**48.3**	**53.4** (+1.6%)	**47.7**	**47.7**	**20.6**	**47.2**	**40.8** (+5.4%)	**46.2** (+3.5%)

(Theorem 1), which preserves the inner-product structure required for a full-corpus retriever, not just a reranker. The success of this step validates our anchoring strategy (Theorem 2). As shown in Appendix D.3, performance scales with the retrieval scope, improving from 46.9% average EM when reranking a top-10 pool to 47.8% on the rebuilt full-corpus index.

7 EXPERIMENTS

We evaluate R3PO across retriever backbones and QA datasets, compare against HARR, and examine search usage and policy temperature. The experimental settings are as follows:

Datasets and evaluation. We evaluate on seven open-domain QA benchmarks: NQ, TriviaQA, and PopQA for single-hop questions (Kwiatkowski et al., 2019; Joshi et al., 2017; Mallen et al., 2023), and HotpotQA, 2WikiMultiHopQA, MuSiQue, and Bamboogle for multi-hop reasoning (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022; Press et al., 2023). We report Exact Match (EM) as the primary metric.

Backbones and comparisons. We instantiate R3PO with three diverse retriever backbones: E5 (general-purpose encoder) (Wang et al., 2022), Qwen-0.6B (decoder-based) (Zhang et al., 2025), and Agentic-R (an agent-adapted retriever) (Liu et al., 2026). Our primary baseline for each is the same Stage-I trained reasoner paired with a frozen version of the corresponding retriever (denoted SEARCH-R1 + [Retriever]). This isolates the gains from R3PO's Stage-II retriever optimization.

Training and evaluation configurations are provided in Appendix C.

7.1 MAIN RESULTS

Table 2 shows that R3PO delivers consistent and significant performance improvements across all three backbones. Average EM improves with relative gains of 2.6% for E5, 2.2% for Agentic-R, and 3.5% for Qwen-0.6B. The method improves or matches the baseline in all 21 dataset-backbone comparisons, demonstrating the robustness of our unified adaptation framework (Pillar 2).

R3PO complements and surpasses supervised adaptation. Notably, R3PO with the general-purpose E5 retriever (47.8% EM) outperforms the baseline that uses Agentic-R (47.2% EM), a state-of-the-art retriever already fine-tuned with supervised signals. Applying R3PO to Agentic-R further boosts its performance to 48.3% EM. This shows that direct RL optimization on trajectory outcomes is a powerful and complementary approach to supervised pre-adaptation.

Gains are more pronounced on multi-hop tasks. The relative gains on multi-hop datasets (avg. +4.0%) are consistently larger than on single-hop ones (avg. +1.5%). This supports our motivation: by aligning the retriever with final task success, R3PO learns to fetch crucial "bridge" documents needed for complex reasoning chains, but a retriever trained on simple relevance might miss.

Improved evidence quality leads to robust reasoning. To verify that these end-to-end gains originate from a fundamentally better retriever, we also evaluated its performance in isolation. As shown in Appendix D.4, R3PO achieves significant improvements in retriever-only recall (Pass@3), with relative gains of up to 4.6%. This confirms that post-training increases the probability of surfacing answer-bearing evidence before the reasoner acts, demonstrating a genuine improvement in evidence quality. The qualitative case studies in Appendix F provide concrete examples of this in action, showing how the adapted retriever corrects specific failures like broken multi-hop reasoning chains (Table 14) and distractions from topically related but incorrect entities (Table 15), leading to more robust and accurate agent performance.

Table 3: End-to-end EM (%) with the Qwen-0.6B retriever. Bold marks the best result in each column across all five configurations. Parentheses in all averages show relative gains over the frozen retriever with the same RL state, computed before rounding the dataset averages.

		Single-hop QA				Multi-hop QA					
Method	RL State	NQ	TriviaQA	PopQA	Avg.	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.	Overall Avg.
Frozen	Query + history	46.7	64.9	48.1	53.2	44.9	45.0	16.4	38.4	36.2	43.5
HARR	Query + history	46.5	**65.0**	**48.5**	53.3 (+0.2%)	46.5	47.0	18.3	37.6	37.4 (+3.2%)	44.2 (+1.6%)
Frozen	Query only	46.4	64.1	47.3	52.6	46.6	47.4	19.2	41.6	38.7	44.7
HARR	Query only	47.0	64.9	47.3	53.1 (+0.9%)	47.0	47.1	19.8	45.6	39.9 (+3.0%)	45.5 (+2.0%)
R3PO	Query only	**47.5**	64.5	48.3	**53.4** (+1.6%)	**47.7**	**47.7**	**20.6**	**47.2**	**40.8** (+5.4%)	**46.2** (+3.5%)

7.2 Validating Principled Generalization to Full-Corpus Retrieval

A core claim of R3PO is that it bridges the "candidate-to-corpus chasm" using background documents as anchors (Pillar 3). To validate this, we ablate these anchors during training ($K_3 = 0$) and compare against the default configuration ($K_3 = 256$) in two deployment scenarios: (1) reranking a top-10 candidate pool and (2) retrieving from the rebuilt full-corpus index.

As shown in Table 11 (Appendix D.3), the results reveal a critical reversal. Without background anchors, the model performs slightly better as a reranker (47.1% vs. 46.9% EM). However, this advantage vanishes upon full-corpus deployment, where it achieves **no aggregate gain** over the reranking performance (47.1%→47.0%). In contrast, the R3PO model trained with background anchors sees its performance **scale effectively** from reranking (46.9%) to full-corpus retrieval (47.8%).

This directly validates our analysis in Theorem 2: training without anchors learns policies that do not generalize reliably beyond the candidate pool. The anchoring mechanism is essential for constraining score drift and ensuring that performance gains translate to a practical full-corpus setting.

7.3 Analysis of Efficiency and Design Choices

Higher Accuracy with Higher Efficiency. R3PO improves both accuracy and efficiency. As shown in Table 4, the improved retriever enables the agent to solve tasks with slightly fewer search calls on average (1.0%-1.2% reduction). This demonstrates that by providing more relevant evidence earlier, R3PO fosters a more direct and effective reasoning path.

Table 4: End-to-end accuracy and search-call counts. Avg. S is the average number of search actions per trajectory; ↑/↓ indicate better directions.

Method	Avg. EM ↑	Avg. S ↓
SEARCH-R1 + E5	46.5	1.93
R3PO + E5	**47.8** (+2.6%)	**1.91** (-1.0%)
SEARCH-R1 + Agentic-R	47.2	2.49
R3PO + Agentic-R	**48.3** (+2.2%)	**2.46** (-1.2%)
SEARCH-R1 + Qwen-0.6B	44.7	1.84
R3PO + Qwen-0.6B	**46.2** (+3.5%)	**1.82** (-1.1%)

Validating Principled Policy Construction. In Section 4, we argued that a positive affine transformation is the only shared scalar map preserving the geometry of a contrastively trained retriever (Pillar 1). Table 9 (Appendix D.2) tests this by comparing our native policy ($s_\phi/(\tau T)$) with other mappings. Our theoretically-grounded policy achieves the highest average EM (47.8%), confirming that respecting the retriever's native score structure is not just theoretically sound but also practically effective.

Validating State Representation. We compare R3PO to HARR (Zhang et al., 2026), an RL-based method that incorporates interaction history into the retriever state. As shown in Table 3, original HARR achieves 44.2% average EM with a query-plus-history state. Replacing it with our query-only formulation raises HARR to 45.5% EM, supporting our principle that retriever states should align with representations learned during pretraining. R3PO further improves this query-only setting to 46.2% EM, showing gains from both the principled state design and our policy optimization framework.

8 Conclusion

We introduced R3PO, a framework that unifies retriever and reasoner optimization in agentic RAG via a shared reinforcement learning objective. Our approach is founded on a principled policy construction that preserves the retriever's efficient, deployable inner-product structure. To solve the critical candidate-to-corpus chasm, R3PO introduces background documents as anchors during training, a theoretically-grounded mechanism that ensures policies learned on small candidate sets generalize to the full corpus. This unified approach yields consistent relative EM gains of 2.2%-3.5% across diverse benchmarks, demonstrating that it is a powerful and complementary method for building truly end-to-end optimized agents.

Reproducibility Statement

Sections 4–7 specify the theory, objectives, and evaluation. Appendix B provides proofs and candidate-support analyses; Appendix C records implementation details, hyperparameters, and the public Stage-I recipe. We will release code and complete configurations upon publication.

References

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. Research: Learning to reason with search for llms via reinforcement learning. In *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.

Conor Durkan, Iain Murray, and George Papamakarios. On contrastive learning for likelihood-free inference. In *International conference on machine learning*, pp. 2771–2781. PMLR, 2020.

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*, pp. 6609–6625, 2020.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke 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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1601–1611. Association for Computational Linguistics, 2017.

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

Wenhan Liu, Xinyu Ma, Yutao Zhu, Yuchen Li, Daiting Shi, Dawei Yin, and Zhicheng Dou. Agentic-r: Learning to retrieve for agentic search. 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. 15987–16005. Association for Computational Linguistics, 2026.

Xingxian Liu, Dongshuai Li, Tao Wen, Jiahui Wan, Gui Ling, Fuyu Lv, Dan Ou, and Haihong Tang. Taosearchemb: A multi-objective reinforcement learning framework for dense retrieval in taobao search. *arXiv e-prints*, pp. arXiv–2511, 2025.

Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3698–3707, 2018.

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 *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, pp. 1607–1612. AAAI Press, 2010.

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

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

Guangxin Song, Xing Fang, Mingmin Jin, Jing Wang, Bokang Wang, Zhentao Song, Junjie Bai, and Jianbo Zhu. Ssr-grpo: Integrating supervision and semantic ids into reinforcement learning for dense retrieval in e-commerce. *arXiv preprint arXiv:2608.19595*, 2026.

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. *arXiv preprint arXiv:2503.05592*, 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554, 2022.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *CoRR*, abs/2212.03533, 2022.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. R^3: Reinforced ranker-reader for open-domain question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 5981–5988. AAAI Press, 2018.

Shaowei Wei, Chong Huang, Songtao Fang, Jin Zhang, Zhuojun Wang, and Chengfu Huo. Learning from what you retrieve: Online rl fine-tuning for semantic retrieval. *arXiv preprint arXiv:2608.30753*, 2026.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher 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.

Hansi Zeng, Liam Collins, Bhuvesh Kumar, Neil Shah, and Hamed Zamani. Cosearch: Joint training of reasoning and document ranking via reinforcement learning for agentic search. *arXiv preprint arXiv:2604.17555*, 2026.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *CoRR*, abs/2506.05176, 2025.

Yicheng Zhang, Zhen Qin, Zhaomin Wu, Wenqi Zhang, and Shuiguang Deng. Reinforcement fine-tuning for history-aware dense retriever in rag. *arXiv preprint arXiv:2602.03645*, 2026.

Jiawei Zhou and Lei Chen. Optimizing retrieval for RAG via reinforcement learning. In *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.

Yuqi Zhou, Sunhao Dai, Changle Qu, Liang Pang, Jun Xu, and Ji-Rong Wen. Learning to retrieve from agent trajectories. In Alistair Moffat, Falk Scholer, Hannah Bast, Marc Najork, and Min Zhang (eds.), *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2026, MelbourneVICAustralia, July 20-24, 2026*, pp. 2544–2555. ACM, 2026.

A Literature Review

The Evidence-Selection Bottleneck in Agentic Search. The agentic RAG paradigm has spurred methods that train language policies to interleave reasoning and search based on task outcomes. Works like Search-R1 (Jin et al., 2025), ReSearch (Chen et al., 2025), and R1-Searcher (Song et al., 2025) use RL to teach agents *when* and *what* to search. However, they treat the dense retriever as a fixed, external tool. While valuable, improving query generation does not resolve the evidence selection bottleneck. Oracle experiments in CoSearch (Zeng et al., 2026) confirm this, showing that promoting better documents from the *already-retrieved* candidate pool yields substantial performance gains. This motivates extending outcome-driven adaptation directly to the retriever.

Indirect Supervision from Downstream Utility. A prominent approach converts agent outcomes into proxy supervisory signals for the retriever. For instance, Agentic-R (Liu et al., 2026) and LRAT (Zhou et al., 2026) use complex pipelines involving LLM annotators or continuation rollouts to generate contrastive pairs or passage labels from agent trajectories. Similarly, R3 (Zhou & Chen, 2025) uses reinforced contrastive learning. These methods transform the RL problem into a supervised learning task, breaking the end-to-end feedback loop. While establishing the value of downstream utility, they align the retriever with an intermediate proxy, not the holistic success of the agent's trajectory. In contrast, R3PO avoids creating proxy labels and instead applies trajectory-level advantages directly to the likelihood of sampled document actions, maintaining a more direct RL paradigm.

Native Retriever Adaptation: Bypassing vs. Generalizing. One way to improve retrieval is to introduce auxiliary modules. CoSearch (Zeng et al., 2026) and R^3 (Wang et al., 2018) add a separate ranker trained with RL. This bypasses the challenge of adapting the native retriever, but at the cost of abandoning a single, efficient, and deployable full-corpus index. A more direct paradigm applies RL to the native retriever's scoring function. HARR (Zhang et al., 2026) and PAO (Wei et al., 2026) are key examples, applying RL directly to a native similarity-based policy. They establish the viability of native retriever RL but make simplifying assumptions, such as updating only the query encoder or using only positive advantages. More critically, they operate on the unexamined assumption that a policy learned over a small candidate set will generalize to a full-corpus index. R3PO builds on this direct RL paradigm but addresses these limitations head-on.

B Theory of native-policy reuse and candidate-conditioned adaptation

B.1 Proof and interpretation of the contrastive density-ratio identity

Equation (2) says that contrastive scores measure evidence for the positive-sampling process relative to the negative-sampling process. The proof follows the classification problem that InfoNCE actually solves: given a list of sampled documents, identify the position occupied by the positive. We write out that argument to explain why a density ratio, rather than the positive probability alone, appears in the score.

Setup. Fix a query q and a finite corpus $\mathcal{D}$. Abbreviate the positive conditional as $p_+(d) = p_{\mathrm{data}}(d \mid q)$ and the negative-sampling conditional as $\nu(d) = \nu(d \mid q)$. Assume both are strictly positive on $\mathcal{D}$. Each instance contains $M \geq 2$ documents, one positive and $M-1$ independent negatives, with the negatives also independent of the positive. Randomly place the positive at a position I drawn uniformly from $\{1, \ldots, M\}$. This random placement changes no contrastive information; it makes the identity of the positive an explicit classification target. Let $D_{1:M} = (d_1, \ldots, d_M)$ be the resulting list.

Step 1: compute which position is most likely to contain the positive. If position i is positive, that document is sampled from p_+ and every other document is sampled from ν. Therefore

$$\Pr(I = i, D_{1:M} \mid q) = \frac{1}{M} p_+(d_i) \prod_{j \neq i} \nu(d_j). \tag{14}$$

Applying Bayes' rule and dividing each term by the common positive factor $\prod_j \nu(d_j)$ gives

$$r_i := \Pr(I = i \mid q, D_{1:M}) = \frac{p_+(d_i)/\nu(d_i)}{\sum_{k=1}^{M} p_+(d_k)/\nu(d_k)}. \tag{15}$$

The ratio has a direct meaning: a document that often occurs among negatives is less diagnostic of the positive position, unless it is even more strongly favored by the positive distribution. Merely knowing its positive probability is not enough to solve this classification problem.

Step 2: match this posterior with the InfoNCE classifier. At contrastive temperature $\tau > 0$, a score function s assigns the model posterior

$$\widehat{r}_i = \frac{\exp(s(q, d_i)/\tau)}{\sum_{k=1}^{M} \exp(s(q, d_k)/\tau)}. \tag{16}$$

The population InfoNCE loss is the expected negative log-probability of the positive position. Conditional on the sampled list, its cross-entropy decomposes into the entropy of the true posterior plus a KL divergence:

$$\mathcal{L}_{\mathrm{NCE}}(s) = \mathbb{E}_{D_{1:M}}\left[-\sum_{i=1}^{M} r_i \log \widehat{r}_i\right] = \mathbb{E}_{D_{1:M}}\left[H(r) + D_{\mathrm{KL}}(r \| \widehat{r})\right]. \tag{17}$$

Here $H(r) = -\sum_i r_i \log r_i$ is independent of the model scores. If the score family can represent the Bayes posterior and reaches the population optimum, the KL term is zero and $\widehat{r} = r$ for every list with positive probability. This is the point at which the expressivity and optimality assumptions enter the argument.

Step 3: translate posterior odds into score differences. Matching the odds for any two document positions yields

$$\frac{s^*(q, d) - s^*(q, u)}{\tau} = \log \frac{p_+(d)}{\nu(d)} - \log \frac{p_+(u)}{\nu(u)}. \tag{18}$$

Strict positivity and $M \geq 2$ allow every document pair to co-occur. Fixing any reference document u consequently gives

$$\frac{s^*(q, d)}{\tau} = \log \frac{p_+(d)}{\nu(d)} + b(q), \qquad b(q) = \frac{s^*(q, u)}{\tau} - \log \frac{p_+(u)}{\nu(u)}. \tag{19}$$

This proves Eq. (2). The classification problem identifies score differences; it has no reason to fix the common offset $b(q)$. That is the source of the offset invariance used by Theorem 1, not an additional learned quantity that R3PO must estimate. This argument specializes established contrastive and conditional-NCE results (Oord et al., 2018; Ma & Collins, 2018; Durkan et al., 2020).

A small example. Suppose three documents have positive probabilities $(0.6, 0.3, 0.1)$ and negative-sampling probabilities $(0.3, 0.3, 0.4)$. The density ratios are $(2, 1, 0.25)$. Their normalized optimal scores are therefore $\log 2$, $\log 1$, and $\log 0.25$, up to the same additive offset. Softmax of these scores gives $(8/13, 4/13, 1/13)$, not the original positive probabilities. The first document is favored over the second by two-to-one odds because their negative-sampling probabilities match; the third is downweighted further because it is relatively common under the negative sampler. If the negative probabilities were instead uniform, their common denominator would cancel, and the normalized score distribution would equal the positive distribution.

Assumptions and the implemented reference. Under nonuniform negatives, recovering p_+ requires the corrected logit $s^*/\tau + \log \nu(d)$, which can change raw-similarity rankings and is not part of R3PO. If $p_+(d) = 0 < \nu(d)$, exact recovery requires a logit tending to $-\infty$, so a finite optimum need not exist. A fixed curated hard-negative set is also not equivalent to iid mixture negatives just because their expected denominators match. These qualifications concern the idealized pretraining characterization. The native policy and its reference-relative identity in Section 5 remain defined directly from the actual finite checkpoint.

B.2 Proof of Theorem 1 and Corollary 1

Precise structural assumptions. Let $g : \mathbb{R} \to \mathbb{R}$ be one shared pointwise map, independent of the query and candidate set. Fix any $T_0 > 0$. Require, for every $n \geq 2$, every score vector $u \in \mathbb{R}^n$, and every $c \in \mathbb{R}$,

$$\mathrm{softmax}(g(u + c\mathbf{1})/T_0) = \mathrm{softmax}(g(u)/T_0), \tag{20}$$

and require g to be strictly increasing. Applying g to a vector means applying it coordinatewise. These are conditions on the policy map, not an assumption that a fixed neural encoder realizes every shifted score vector.

Necessity. Apply Eq. (20) to a two-element vector and compare odds. Because T_0 is the same on both sides,

$$g(x + c) - g(y + c) = g(x) - g(y) \quad \text{for all } x, y, c \in \mathbb{R}. \tag{21}$$

Set $y = 0$ and $h(x) = g(x) - g(0)$. Then $h(x+c) = h(x) + h(c)$. In particular, $h(0) = 0$, $h(-x) = -h(x)$, and $h(r) = rh(1)$ for every rational r. For any real x, choose rational sequences $r_m \uparrow x$ and $t_m \downarrow x$. Monotonicity gives $r_m h(1) \leq h(x) \leq t_m h(1)$, so $h(x) = \kappa x$, with $\kappa = h(1) > 0$. Thus $g(x) = \kappa x + g(0)$.

Sufficiency. For $g(x) = \kappa x + a$ with $\kappa > 0$, the map preserves order and a common score shift adds the constant κc to every logit. The softmax is unchanged. This proves Theorem 1. The proof is a direct application of the standard monotone-additive functional-equation argument.

Calibrating the free slope. Under uniform-negative population optimality, write $s_i^* = \tau \log p_i + b$, where p is a strictly positive pretraining distribution. At the calibration convention $T = 1$,

$$\mathrm{softmax}_i(g(s_i^*)) = \frac{p_i^{\kappa\tau}}{\sum_j p_j^{\kappa\tau}}. \tag{22}$$

For two unequal positive masses p_i, p_j, matching their odds requires $(p_i/p_j)^{\kappa\tau} = p_i/p_j$, hence $\kappa\tau = 1$. Conversely, $\kappa = 1/\tau$ recovers p. Requiring calibration for every positive nonuniform conditional therefore gives Corollary 1. Uniform distributions alone do not determine the slope. On a restricted candidate set, normalization recovers the corresponding conditional, not the original full-corpus masses.

Scale, temperature, and scope. At general T, the idealized distribution is proportional to $p_i^{1/T}$. Dividing a calibrated g by a different temperature does not preserve calibration at $T = 1$. With temperature free, different positive slopes describe the same temperature-scaled family. The corollary fixes a scale convention under explicit idealized assumptions; it proves neither downstream reward optimality nor an ordering of empirical results.

The theorem concerns shared scalar functions of the native score. A head depending on representations, a query-dependent slope, or a set-dependent transformation is not covered. For example, centering scores before applying a nonlinear map can respect common shifts, but that construction depends on the score set rather than a single shared pointwise g. Some projection heads preserve the original score at initialization and remain factorized. These constructions are not disproved by Theorem 1. Its purpose is to characterize the simplest policy map consistent with the stated reuse requirements.

B.3 Reference-relative adaptation and fixed-utility improvement

Let $f_\phi(d) = s_\phi(q, d)/(\tau T)$ and $\Delta_\phi = f_\phi - f_{\phi_0}$ on fixed (q, C). Substituting $f_\phi = f_{\phi_0} + \Delta_\phi$ into the softmax gives

$$p_\phi^C(d) = \frac{p_0^C(d) e^{\Delta_\phi(d)}}{\sum_{u \in C} p_0^C(u) e^{\Delta_\phi(u)}}. \tag{23}$$

This exact identity describes the implemented policy relative to initialization. It makes no assumption about the pretraining noise law or population optimality.

For a positive reference p_0 on a finite set, finite fixed utility Q, and $\beta_Q > 0$, put $Z = \sum_d p_0(d)e^{Q(d)/\beta_Q}$ and $p^*(d) = p_0(d)e^{Q(d)/\beta_Q}/Z$. Then

$$\mathbb{E}_p Q - \beta_Q D_{\mathrm{KL}}(p\|p_0) = \beta_Q \log Z - \beta_Q D_{\mathrm{KL}}(p\|p^*). \tag{24}$$

Nonnegativity of KL establishes the unique unrestricted optimum. Taking odds gives Eq. (7). This standard variational identity explains how utility can correct prior preferences (Peters et al., 2010). A shared finite encoder need not realize the unrestricted solution. In a sequential agent, Q denotes a continuation utility for a fixed local surrogate, not an intrinsic label for a document. The actual group-standardized, clipped GRPO objective is not asserted to solve this problem exactly.

B.4 Proof of Proposition 1: conditional, joint, and marginal KL

Fix q and a common candidate law μ. Extend $p_\phi^C(d)$ by zero outside C and define

$$J_\phi(C,d) = \mu(C)p_\phi^C(d), \qquad \bar{p}_\phi(d) = \sum_C J_\phi(C,d). \tag{25}$$

The common candidate law cancels in the joint likelihood ratio, giving

$$D_{\mathrm{KL}}(J_\phi\|J_0) = \sum_C \mu(C) D_{\mathrm{KL}}(p_\phi^C\|p_0^C) = \mathcal{K}_\mu. \tag{26}$$

Marginalizing out C cannot increase KL, so $D_{\mathrm{KL}}(\bar{p}_\phi\|\bar{p}_0) \le \mathcal{K}_\mu$. Pinsker's inequality, with natural logarithms and $\|P-Q\|_{\mathrm{TV}} = \frac{1}{2}\sum_x |P(x)-Q(x)|$, yields

$$\|J_\phi - J_0\|_{\mathrm{TV}} \le \sqrt{\mathcal{K}_\mu/2}, \qquad \|\bar{p}_\phi - \bar{p}_0\|_{\mathrm{TV}} \le \sqrt{\mathcal{K}_\mu/2}. \tag{27}$$

For a fixed candidate set, Pinsker separately gives $\|p_\phi^C - p_0^C\|_{\mathrm{TV}} \le \sqrt{D_{\mathrm{KL}}(p_\phi^C\|p_0^C)/2}$. An average bound on $\mathcal{K}_\mu$ does not imply the same bound for every C.

These are bounds on deviations from the actual checkpoint-induced reference under the same candidate law. They do not require calling that reference an exactly recovered pretraining data distribution. Nor does a chosen penalty coefficient β imply that $\mathcal{K}_\mu \le \varepsilon$. Changing the query distribution, candidate law, or support introduces additional differences not covered by this statement.

B.5 Proof of Theorem 2: what candidate co-occurrence identifies

Suppress the fixed query. For finite logits f, write $p_f^C(d) = e^{f(d)}/\sum_{u\in C} e^{f(u)}$. Since KL is nonnegative,

$$\mathcal{K}_\mu(f, f_0) := \mathbb{E}_C D_{\mathrm{KL}}(p_f^C\|p_{f_0}^C) = 0 \tag{28}$$

if and only if the two positive conditional distributions agree on every candidate set with positive probability. Equality of softmax odds is equivalent to $f(d) - f_0(d) = b_C$ for all $d \in C$. Thus every co-occurring pair has equal displacement. Equality propagates along paths in the comparison graph, proving necessity of componentwise constancy. Conversely, each candidate set lies inside a connected component, and the shared displacement cancels in its softmax. This proves sufficiency.

For a proper nonempty hard-only pool B with complement O, let $f_c = f_0 + c\mathbf{1}_B$. The conditional distribution on B is unchanged, but the full-corpus mass satisfies

$$P_{f_c}(O) = \frac{Z_O}{e^c Z_B + Z_O}, \qquad Z_S = \sum_{d\in S} e^{f_0(d)}, \quad P_f = \mathrm{softmax}_{\mathcal{D}}(f). \tag{29}$$

As $c \to -\infty$, this mass tends to one. Cross-pool rankings can therefore change without any within-pool KL. The statement identifies a score-space direction; it does not assume a particular shared encoder can realize arbitrary independent shifts for every query or that optimization necessarily follows it.

For $C = B \cup U$, if each document outside B has positive probability of uniform inclusion and B is nonempty, every outside document co-occurs with B. The population support graph is connected. Exact zero population conditional KL then identifies the full-corpus softmax. Positive support is not exhaustive finite-sample coverage, and small nonzero KL requires quantitative conditioning assumptions to yield useful control.

B.6 Cross-pool curvature and the weighted comparison graph

On a fixed $C \supseteq B$, take $f_c = f_0 + c\mathbf{1}_B$ and $\alpha = p^C_{f_0}(B)$. Let

$$A(c) = \log(1 - \alpha + \alpha e^c), \qquad \alpha_c = \frac{\alpha e^c}{1 - \alpha + \alpha e^c}. \tag{30}$$

Within each block, the conditional preferences are unchanged. Direct substitution yields

$$D_{\mathrm{KL}}(p^C_{f_c} \| p^C_{f_0}) = c\alpha_c - A(c) = \tfrac{1}{2}\alpha(1-\alpha)c^2 + O(c^3). \tag{31}$$

The Fisher information for this block-shift direction at the reference is $\alpha(1-\alpha)$. It vanishes for hard-only support and becomes weak if either block has negligible probability mass. This explains why background count alone is not a calibration metric. The expansion is centered at the reference logits, not at an arbitrary current policy.

For a perturbation vector h and scalar t, the general local expansion is

$$\mathcal{K}_\mu(f_0 + th, f_0) = \tfrac{1}{2}t^2 \mathbb{E}_C \operatorname{Var}_{p^C_{f_0}}(h(d)) + O(t^3), \tag{32}$$

$$\mathbb{E}_C \operatorname{Var}_{p^C_{f_0}}(h(d)) = \sum_{u<v} w_{uv}(h(u) - h(v))^2 = h^\top L h, \tag{33}$$

$$w_{uv} = \mathbb{E}_C[\mathbf{1}\{u, v \in C\} p^C_{f_0}(u) p^C_{f_0}(v)] \quad (u \neq v). \tag{34}$$

Use any fixed corpus ordering for $u < v$, set $w_{uu} = 0$, and define the unnormalized Laplacian $L = \operatorname{diag}(W\mathbf{1}) - W$. Its nullspace consists of componentwise-constant shifts. Thus the same graph describes both exact identifiability and local regularization strength.

B.7 Quantitative stability and a ranking-margin condition

Fix one query and a finite corpus of at least two documents. Let $\delta = f - f_0$ satisfy $\max_d \delta(d) - \min_d \delta(d) \le R$ with $R \ge 0$, and write $K = \mathcal{K}_\mu(f, f_0)$. For the reference-weighted Laplacian above,

$$K \ge \frac{e^{-R}}{2} \delta^\top L \delta. \tag{35}$$

If the graph is connected, its spectral gap $\lambda_2(L) > 0$ yields

$$|\delta(u) - \delta(v)| \le 2\sqrt{\frac{e^R K}{\lambda_2(L)}} \quad \text{for every } u, v. \tag{36}$$

A reference margin $f_0(u) - f_0(v)$ exceeding this bound preserves $f(u) > f(v)$. A reference Top-k boundary gap exceeding it preserves membership. Scores and margins must use the same scaled logits as the policy.

Proof. Let $p^C_t = \operatorname{softmax}_C(f_0 + t\delta)$ and $A_C(t) = \log \mathbb{E}_{p^C_0} e^{t\delta}$. The log-partition derivatives give

$$D_{\mathrm{KL}}(p^C_1 \| p^C_0) = A'_C(1) - A_C(1) = \int_0^1 t \operatorname{Var}_{p^C_t} \delta \, dt. \tag{37}$$

The oscillation bound implies $p^C_t(d)/p^C_0(d) \ge e^{-tR}$. Since $\operatorname{Var}_p \delta = \min_b \mathbb{E}_p(\delta - b)^2$, $\operatorname{Var}_{p^C_t} \delta \ge e^{-tR} \operatorname{Var}_{p^C_0} \delta$. Integrating and averaging proves Eq. (35). A sharper multiplier of $\delta^\top L \delta$ is $[1 - (1 + R)e^{-R}]/R^2$ for $R > 0$, with limit $1/2$ at zero. For $\widetilde{\delta} = \delta - \bar{\delta}\mathbf{1}$, centered by the unweighted corpus mean,

$$\lambda_2(L)\|\widetilde{\delta}\|_2^2 \le \delta^\top L \delta \le 2e^R K. \tag{38}$$

Finally, $|\delta(u) - \delta(v)| \le \sqrt{2}\|\widetilde{\delta}\|_2$ gives Eq. (36).

The bound is a sufficient condition for score stability at the query being analyzed. A small spectral gap or large score displacement can make it loose. The configured KL coefficient and an aggregate sampled training loss do not establish the pointwise assumptions by themselves. These conditions distinguish identification of a score direction from quantitative control of its magnitude. Whether an adapted encoder retains performance on other retrieval tasks is a separate empirical question, not a conclusion of this score-space theorem.

B.8 WHY SAMPLER STABILITY IS NOT AUTOMATICALLY FULL-CORPUS STABILITY

Partition the corpus into nonempty B and O. Let $P_f = \mathrm{softmax}_{\mathcal{D}}(f)$, $\alpha = P_f(B)$, and $\alpha_0 = P_{f_0}(B)$. The KL chain rule gives

$$D_{\mathrm{KL}}(P_f \| P_{f_0}) = \mathrm{kl}(\alpha \| \alpha_0) + \alpha D_{\mathrm{KL}}(P_{f,B} \| P_{f_0,B}) \tag{39}$$

$$+ (1-\alpha) D_{\mathrm{KL}}(P_{f,O} \| P_{f_0,O}). \tag{40}$$

Hard-only regularization sees the within-B term but not the block mass or the outside conditional. For two documents, suppose a candidate law selects their pair with probability η and a singleton otherwise. Its graph is connected for every $\eta > 0$, yet $\mathcal{K}_\mu = \eta D_{\mathrm{KL}}(P_f \| P_{f_0})$. A fixed full-corpus discrepancy can therefore incur arbitrarily small conditional KL as $\eta \to 0$. This is consistent with both Proposition 1, which concerns the sampling marginal, and the spectral bound, whose graph weights become weak.

Even small full-corpus KL need not preserve Top-1 without a margin. Swapping the probabilities $(1/2+\epsilon, 1/2-\epsilon)$ reverses Top-1 while its KL from the original distribution is $2\epsilon \log[(1/2+\epsilon)/(1/2-\epsilon)]$, which tends to zero. Distributional proximity, score-margin stability, and retrieval-task retention are distinct claims.

B.9 INITIALIZATION AND REWARD VARIATION: A LIMITED DIAGNOSTIC

Suppose $G \geq 2$ rollout rewards for a fixed question are iid Bernoulli with success probability p. The only constant groups are all failures or all successes, so

$$\Pr(\text{nonconstant reward group}) = 1 - (1-p)^G - p^G. \tag{41}$$

This quantity vanishes at zero and one and is maximal at $p = 1/2$. An initialization that almost never succeeds can yield few nonzero standardized reward advantages, but greater success does not monotonically increase reward variation. Nonconstant rewards also do not guarantee a nonzero parameter gradient: score derivatives and aggregation matter. Constant reward advantages need not remove a nonzero KL gradient away from the reference. The implemented shaped reward can vary through containment and search cost even when answer correctness is constant, so Eq. (41) is a binary diagnostic, not a model of all Stage-II feedback.

For an exactly uniform policy over a disjoint union of 10 hard and 256 distinct background documents, hard-pool probability is $10/266 \approx 3.76\%$. Deduplication can change this denominator. This number is not a success probability or a generic prediction for randomly initialized heads: head logits need not be uniform, and initialization-preserving heads exist. A native pretrained policy can concentrate on useful candidates, but its actual hard-pool mass, success rate, and group-variance rate should be measured.

B.10 INDEX COMPATIBILITY AND TEMPERATURE LIMITS

For every positive scale, sorting native policy logits is equivalent to sorting native inner products. Re-encoding documents with the adapted encoder therefore permits the same dense-index serving interface, subject to the approximation behavior of the chosen ANN search algorithm. This structural property does not prove optimal retrieval quality.

A single categorical policy as $T \to 0$ concentrates on its maximum-scoring document, not a Top-K set. With ties, probability can remain split among maximizers. Sequential sampling without replacement converges to the ordered Top-K list when scores are distinct, but R3PO's reported Stage-II training samples one document per call. Its deterministic Top-3 evaluation is a separate deployment choice whose consistency warrants ablation.

Auxiliary components are not categorically non-indexable. A bilinear head $E(q)^\top W E(d)$, for example, can be factorized into transformed embeddings. In contrast, general query–document cross-attention and autoregressive list generation do not supply the fixed-vector inner-product scorer used by R3PO's index. The comparison is about the actual scoring function, not an architectural label.

B.11 SCOPE RELATIVE TO THE REPORTED IMPLEMENTATION

Initial current-to-behavior importance ratios equal one whenever the two policies and their temperature, query, and candidate support match. This holds for native policies and other parameterizations alike; it is not a consequence unique to pretraining calibration.

The fixed-utility problem in Appendix B.3 optimizes a distribution on one fixed candidate set while holding its utility values constant. The practical update in Section 5 instead uses trajectory rewards, group-standardized advantages, clipping, and shared encoder parameters across queries. Per-trajectory action-count normalization also changes the weighting of trajectories. Consequently, this GRPO update is a surrogate rather than the exact fixed-utility optimizer or, in general, an unbiased gradient of expected raw return. Query-only retrieval can alias distinct reasoning histories. A shared encoder couples queries, whereas the identification and stability results are pointwise in the query. The exact conditional KL used in the analysis is not identical to every sampled estimator; the original Stage-I recipe reports a low-variance KL estimator.

The mathematical presentation assigns zero reward advantage to constant-reward groups and zero action loss to empty action sets. These conventions avoid undefined normalization; they are separate from numerical stabilization choices in an implementation. A sampled estimator evaluated under behavior-policy actions is not automatically an estimator of current-policy KL after policy updates without an appropriate sampling or weighting argument. None of the theoretical guarantees is inferred from a particular realized estimator value.

C IMPLEMENTATION DETAILS

Following the two-stage process in Figure 2, we first train the reasoner (Stage I) and then freeze it to train the retriever (Stage II). The following subsections detail both training stages and the full-corpus evaluation configuration.

C.1 STAGE I: LLM OPTIMIZATION

We initialize the reasoner from `Qwen/Qwen2.5-7B-Instruct` and follow the Search-R1 GRPO recipe (Jin et al., 2025), using veRL 0.7.0 and the merged NQ and HotpotQA training sets. Table 5 records our configuration.[1] The reasoner samples complete search–reasoning trajectories and receives only final-answer EM rewards. Retrieved passages are masked out of the action loss. As in Search-R1, the clipped GRPO objective is accompanied by a reference-policy KL penalty during LLM training. The dense retriever remains frozen and returns $K = 3$ passages for each query.

Search-R1 implementation and baseline strength. The original Search-R1 repository is a common starting point for subsequent agentic-search training implementations. In our reproduction attempts, its older training stack exhibited substantial discrepancies between training and rollout inference, together with instability that frequently caused runs to fail before completing a single epoch. Such early failures limit the amount of training and can depress baseline performance. We therefore implement the Search-R1 recipe on veRL 0.7.0 and complete two full training epochs. Our resulting Search-R1 baseline achieves substantially higher performance than the Search-R1 baselines reported in other work. We report this implementation difference to contextualize the baseline strength, rather than attributing the cross-work performance gap solely to the framework version. All paired comparisons in this paper use this same trained reasoner, so the reported R3PO gains measure retriever post-training on top of the stronger baseline.

C.2 STAGE II: RETRIEVER OPTIMIZATION

The trained reasoner is frozen. E5, Agentic-R, and Qwen-0.6B all use the same merged NQ and HotpotQA training sets as Search-R1. For each query, a frozen copy of the initial retriever supplies $K_1 = 10$ retrieved documents, which are combined with $K_3 = 256$ documents sampled uniformly

[1] The official Search-R1 NQ–HotpotQA scripts provide the original recipe; our implementation uses veRL 0.7.0.

Table 5: Our Search-R1 GRPO configuration for Stage-I LLM post-training.

Setting	Value
Actor learning rate	1×10^{-6}
Learning-rate warmup ratio	0.285
Training batch size	512
GRPO group size (rollouts per question)	5
Training epochs	2
Clipping coefficient ϵ	0.2
Rollout temperature	1.0
Reference KL coefficient	0.001 (`low_var_kl`)
Prompt / initial-input token limit	4096 / 2048
Response / observation token limit	500 / 500
Retriever passages per query	3

from the corpus to form $C_1(q)$. The frozen retriever defines $\pi_{\text{ref}}^{\text{ret}}$ on $C_1(q)$; its trainable copy re-encodes the query and candidates and samples $K_2 = 1$ document. Table 6 lists the backbone-specific settings. We use base score scales $\tau = 0.01$ for E5 and Agentic-R, consistent with their reported contrastive training scale, and $\tau = 0.02$ for Qwen-0.6B, with sampling temperature $T = 1$ for all backbones. For Qwen-0.6B, 0.02 is the selected implementation scale; we do not identify it with an independently verified pretraining temperature. These settings define the effective scale τT without asserting exact pretraining calibration of a finite checkpoint. The retriever KL coefficient is $\beta_{\text{ret}} = 0.001$ for all backbones.

Table 6: Backbone-specific retriever post-training hyperparameters.

Retriever	Learning rate	Base score scale τ	KL coefficient β_{ret}
E5	2×10^{-5}	0.01	0.001
Agentic-R	2×10^{-5}	0.01	0.001
Qwen-0.6B	6×10^{-6}	0.02	0.001

Table 7 gives the shared Stage-II settings. Training uses the same batch size and GRPO group size as Stage I, but runs for one epoch. Both stages use clipping coefficient $\epsilon = 0.2$ and reference KL coefficient 0.001.

Table 7: Shared GRPO configuration for Stage-II retriever post-training.

Setting	Value
Training batch size	512
GRPO group size (rollouts per question)	5
Training epochs	1
Clipping coefficient ϵ	0.2
Reference KL coefficient β_{ret}	0.001

The trajectory reward combines answer EM, answer containment, and search count with weights 1, 0.5, and 0.2, respectively, with the last term subtracted. Group-relative advantages update sampled document actions while the candidate index remains fixed. The E5 policy-logit ablation fixes $\tau = 0.01$ and $T = 1$ and compares $s/(5\tau T)$, $2s/(\tau T)$, $s/(\tau T)$, and $s^5/(\tau T)$ (Table 9). After post-training, we re-encode the full corpus with the updated retriever and rebuild its index; evaluation returns deterministic Top-K passages with $K = 3$ from this regenerated corpus representation, rather than merely reranking candidates produced by the frozen retriever.

D Additional Experiments

D.1 Reward ablation

Table 8: Ablation of trajectory reward components for E5 retriever post-training.

Retriever reward	Avg. EM (%)
R_{ans}	47.1
$R_{\text{ans}} + \lambda_{\text{doc}} R_{\text{doc}}$	47.2
$R_{\text{ans}} + \lambda_{\text{doc}} R_{\text{doc}} - \lambda_{\text{search}} N_{\text{search}}$	**47.8**

In Table 8, relative to the frozen E5 baseline (46.5), outcome reward alone reaches 47.1; answer containment yields a further 0.2% relative gain, and our search-count reward yields another 1.3% relative gain to reach 47.8. The reported configurations improve as these signals are added, with the largest observed increments from outcome correctness and the search-count penalty.

D.2 Policy Mapping and Effective Temperature

Experimental setup. We compare four policy-logit configurations for the E5 retriever to examine the policy construction discussed in Section 4. The base score scale is $\tau = 0.01$ and the nominal temperature is $T = 1$. We evaluate the native scaling $s_\phi/(\tau T)$, two alternative linear scalings, and the nonlinear mapping $s_\phi^5/(\tau T)$, reporting average end-to-end EM across the seven benchmarks.

Table 9: Sensitivity to the policy mapping and effective temperature on E5.

Policy logit $f_\phi(q, d)$	Avg. EM (%)
$s_\phi/(5\tau T)$	46.0
$2s_\phi/(\tau T)$	46.8
$s_\phi/(\tau T)$ (ours)	**47.8**
$s_\phi^5/(\tau T)$	47.1

Results and analysis. Table 9 shows that the native scaling achieves the highest average EM among the four tested configurations (47.8%), compared with 46.0% for $s_\phi/(5\tau T)$, 46.8% for $2s_\phi/(\tau T)$, and 47.1% for $s_\phi^5/(\tau T)$. The linear alternatives change the effective temperature while preserving ranking and common-offset invariance. The fifth-power mapping remains increasing but changes score-gap geometry and does not preserve common-offset invariance. These results support the selected policy configuration in this setting.

D.3 Generalization Beyond the Training Retrieval Pool

Experimental setup. We evaluate the same R3PO-post-trained E5 retriever under four retrieval configurations to test whether its improvements extend beyond reranking a small candidate pool. In the first three configurations, the frozen first-stage retriever supplies its top 10, 20, or 50 candidates, and the post-trained retriever re-encodes and ranks these candidates. In the fourth configuration, we re-encode the entire corpus with the post-trained retriever, rebuild the dense index, and retrieve directly from this full-corpus index. The trained reasoner is held fixed, and all configurations return the top $K = 3$ passages per search. We compare against SEARCH-R1 with frozen E5 and report EM on the same seven QA benchmarks.

Results and analysis. Table 10 compares the same adapted E5 retriever under frozen-candidate reranking and rebuilt full-corpus retrieval. Average EM rises from 46.9 for top-10 reranking to 47.3 for top-20, 47.7 for top-50, and 47.8 for full-corpus retrieval. These results show that the learned scorer remains useful beyond the narrow hard pool used during training. They also do not demonstrate superiority to an external reranking method evaluated with a different reasoner, corpus, or retrieval budget.

Table 10: EM (%) for different evaluation configurations of the R3PO-post-trained E5 retriever. Top-N reranking uses N candidates from the frozen first-stage retriever; full-corpus retrieval uses an index rebuilt with the post-trained retriever. Bold marks the best score in each column, including ties. Parentheses show relative gains over frozen E5, computed before rounding the dataset averages.

	Single-hop QA				Multi-hop QA					
Method	NQ	TriviaQA	PopQA	Avg.	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.	Overall Avg.
SEARCH-R1 + E5	50.1	65.7	46.7	54.2	47.8	46.0	22.3	47.2	40.8	46.5
R3PO + E5										
Top-10 reranking	49.5	**66.3**	46.9	54.2 (+0.1%)	48.5	45.9	22.9	48.0	41.3 (+1.2%)	46.9 (+0.7%)
Top-20 reranking	50.2	65.9	48.8	55.0 (+1.5%)	48.9	47.0	23.1	47.2	41.6 (+1.8%)	47.3 (+1.6%)
Top-50 reranking	**50.3**	65.8	**49.1**	**55.1** (+1.7%)	49.1	47.3	23.4	**48.8**	42.2 (+3.2%)	47.7 (+2.5%)
Full-corpus retrieval	50.1	65.9	**49.1**	55.0 (+1.6%)	**49.3**	**47.5**	**23.6**	**48.8**	**42.3** (+3.6%)	**47.8** (+2.6%)

Background-document ablation: reranking versus retrieval. We further compare training without background documents ($K_3 = 0$) against the default $K_3 = 256$ configuration, evaluating each as a top-10 reranker and as a full-corpus retriever under the protocols above. Here, K_3 controls the number of uniformly sampled background documents during training, whereas top-10 refers to the candidate budget at evaluation. The default results are repeated from Table 10 for direct comparison.

Table 11: Effect of background documents on E5 reranking and full-corpus retrieval, measured by end-to-end EM (%). Dataset order and the unweighted seven-dataset average follow Table 2. Bold marks the better result within each evaluation protocol, including ties.

	Single-hop QA			Multi-hop QA				
Training	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
Top-10 reranking								
$K_3 = 0$	**49.7**	65.7	**48.4**	**49.6**	**47.9**	**23.5**	44.8	**47.1**
$K_3 = 256$	49.5	**66.3**	46.9	48.5	45.9	22.9	**48.0**	46.9
Full-corpus retrieval								
$K_3 = 0$	49.6	**65.9**	48.0	49.1	47.0	22.4	47.2	47.0
$K_3 = 256$	**50.1**	**65.9**	**49.1**	**49.3**	**47.5**	**23.6**	**48.8**	**47.8**

Table 11 reveals a reversal between the two deployment modes. Without background documents, the adapted model is a slightly stronger top-10 reranker on average: EM increases from 46.9 to 47.1, with improvements on five of seven datasets, despite decreases on TriviaQA and Bamboogle. As a full-corpus retriever, however, the same ablation is weaker: average EM decreases from 47.8 to 47.0, with lower scores on six datasets and a tie on TriviaQA. Moreover, expanding evaluation from top-10 reranking to full-corpus retrieval raises average EM from 46.9 to 47.8 with background training, but yields no aggregate gain without it (47.1 to 47.0). Thus, stronger performance within a restricted candidate pool does not necessarily imply stronger corpus-wide retrieval.

This pattern is consistent with the candidate-conditioned identifiability analysis in Theorem 2: hard-pool training can improve within-pool ordering while leaving score relationships to excluded documents insufficiently constrained. Background comparisons help constrain those relationships, which matters when the adapted scorer must retrieve directly from the entire corpus. The ablation supports the practical role of background documents in candidate-to-corpus generalization; end-to-end EM alone does not directly measure score drift or establish its causal mechanism. The small reranking advantage is descriptive, as these results do not include uncertainty estimates.

D.4 RETRIEVER-ONLY RECALL AT THREE

We additionally evaluate retrieval quality in isolation, without answer generation. Pass@3 records whether at least one of the top-three retrieved passages contains the answer; we report the unweighted average across the seven benchmarks.

Results and analysis. Table 12 shows that R3PO improves average Pass@3 for all three backbones: from 63.5 to 66.4 for E5, from 65.4 to 67.6 for Agentic-R, and from 60.2 to 61.9 for Qwen-0.6B.

Table 12: Retriever-only Pass@3 (%) on single-hop and multi-hop QA. Rows pair each frozen retriever with its R3PO-post-trained counterpart. Bold indicates the better result within each pair, including ties. Parentheses show relative gains over the paired baseline, computed before rounding the dataset averages.

	Single-hop QA				Multi-hop QA					
Method	NQ	TriviaQA	PopQA	Avg.	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.	Overall Avg.
SEARCH-R1 + E5	74.0	**80.6**	63.6	72.7	70.8	58.3	**42.8**	**54.4**	56.6	63.5
R3PO + E5	**76.1**	78.9	**73.0**	**76.0** (+4.5%)	**72.9**	**63.3**	**46.2**	**54.4**	**59.2** (+4.6%)	**66.4** (+4.6%)
SEARCH-R1 + Agentic-R	76.4	**81.3**	65.5	74.4	73.6	62.4	43.9	54.4	58.6	65.4
R3PO + Agentic-R	**77.0**	81.1	**69.6**	**75.9** (+2.0%)	**76.0**	**63.9**	**45.3**	**60.0**	**61.3** (+4.6%)	**67.6** (+3.4%)
SEARCH-R1 + Qwen-0.6B	68.7	73.8	65.8	69.4	65.9	60.2	37.2	49.6	53.2	60.2
R3PO + Qwen-0.6B	**70.8**	**75.3**	**68.4**	**71.5** (+3.0%)	**68.8**	**60.7**	**39.2**	**50.4**	**54.8** (+2.9%)	**61.9** (+2.9%)

These correspond to relative gains of 4.6%, 3.4%, and 2.9%, respectively. Across the 21 paired dataset–backbone comparisons, R3PO improves 18, ties one, and decreases two. The improvements are therefore observed across all three backbone families rather than being attributable to a single retriever or benchmark.

Crucially, Pass@3 is computed directly from the retrieved passages before the reasoner generates a final answer. The gains therefore demonstrate that retriever post-training increases the probability that the top-three results contain answer-bearing evidence. They cannot be explained solely by an incidental perturbation of the retrieved context that makes the LLM more likely to emit the correct answer despite unchanged evidence recall. This retriever-only result complements the end-to-end EM gains in Table 2: the downstream improvements are accompanied by a measurable increase in the recall of the underlying evidence, rather than arising only from the fixed reasoner's response to altered context.

The largest relative gain in average Pass@3 occurs for E5, at 4.6%, followed by Agentic-R at 3.4% and Qwen-0.6B at 2.9%. For E5, the largest absolute improvement occurs on PopQA, where Pass@3 increases from 63.6 to 73.0, while gains are also observed on NQ, HotpotQA, 2Wiki, and MuSiQue. The already specialized Agentic-R retriever improves on six of seven datasets and rises from 65.4 to 67.6 on average, providing evidence that outcome-driven retriever optimization remains complementary to prior supervised retrieval adaptation. Qwen-0.6B shows a smaller aggregate gain but improves consistently on all seven benchmarks, increasing from 60.2 to 61.9 on average. The two observed regressions are confined to TriviaQA, where E5 decreases from 80.6 to 78.9 and Agentic-R decreases slightly from 81.3 to 81.1; E5 additionally ties on Bamboogle. Thus, the adaptation is not uniformly beneficial for every dataset–backbone pair, although average retrieval recall improves for every backbone.

E LIMITATIONS

Trajectory-level credit. R3PO does not explicitly solve credit assignment among retrieval actions. All documents sampled in a trajectory share the same outcome-based advantage, so the update cannot determine which particular retrieval decision caused success or failure. Action-level or counterfactual credit assignment is an important direction for future work.

Staged rather than end-to-end training. Our implementation first post-trains the LLM with a frozen retriever and then post-trains the retriever with the LLM frozen. It therefore requires two training stages and does not directly optimize both components end to end. Joint or alternating optimization may yield stronger coordination and remove this separation.

Scope of the theory. The scalar-map characterization is restricted to shared pointwise functions under explicit invariance and ordering requirements. Scale calibration additionally assumes an idealized uniform-negative population optimum. Neither result rules out more expressive policies or predicts empirical performance ordering. The reference bounds concern a fixed query and candidate law; quantitative score stability requires the bounded-displacement and graph-conditioning assumptions in Appendix B.7. These are sufficient conditions, not an unconditional guarantee of retention across tasks.

Training and deployment. Query-only retrieval can alias distinct reasoning histories. Training samples one document per search, whereas evaluation returns three passages (Appendix C); a matched document-count ablation remains useful. Rebuilding a dense index establishes interface compatibility, not retained general retrieval quality. The policy-mapping experiment compares the reported configurations; further performance comparisons could control sampling entropy and optimization budgets, while common-offset invariance can be checked directly without retraining.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and manuscript organization. AI assistance also included mathematical consistency checks and literature synthesis.

F CASE STUDIES

To examine how R3PO retriever training affects agentic search, we compare paired trajectories with the same frozen reasoner before and after Stage-II retriever optimization. Tables 13–16 present four examples: eliminating a redundant search, recovering a multi-hop evidence chain, resolving a scope mismatch, and preventing an incorrect entity association. Each comparison preserves every search query and the final answer from the original runs. Retrieved passages are quoted as excerpts, with omissions marked by ellipses and document numbers retaining their original retrieval ranks. Green highlights supporting evidence; red highlights distracting entities or relations in failed trajectories. These examples provide qualitative evidence of how retriever adaptation changes the information available to the reasoner.

F.1 REDUCING REDUNDANT RETRIEVAL

Table 13 illustrates an efficiency gain without a change in answer correctness. Before R3PO retriever training, the first search returns suburban-area descriptions that do not identify BA Connect's headquarters. The reasoner must reformulate the query before it can resolve Didsbury and retrieve its population. After R3PO training, the same initial query retrieves the headquarters relation directly, enabling the next search to target the census figure. Both trajectories return 26,788, but the trained retriever reduces the number of searches from three to two by supplying the missing intermediate entity earlier.

Table 13: Earlier retrieval of the bridge entity removes one search while preserving the correct answer.

Question: In the 2011 census what was the population of suburban area in which BA Connect is headquartered ?

Before R3PO retriever training	**After R3PO retriever training**
Search 1. *What is the suburban area in which BA Connect is headquartered* **Retrieved passages (original excerpts):** **Doc 1 (Chicago metropolitan area):** … The Chicago metropolitan area, or Chicagoland, is the metropolitan area that includes the city of Chicago, Illinois, and its suburbs. … **Doc 2 (Deerfield, Illinois):** … The population was 18,225 at the 2010 census, a decline of 175 from 2000. …	**Search 1.** *What is the suburban area in which BA Connect is headquartered* **Retrieved passages (original excerpts):** **Doc 1 (BA Connect):** … Headquartered in **Didsbury**, Manchester, England, it operated a network of domestic and European services from a number of airports in the United Kingdom on behalf of British Airways. …
Search 2. *BA Connect headquarters location* **Retrieved passages (original excerpts):** **Doc 1 (BA Connect):** … Headquartered in **Didsbury**, Manchester, England, it operated a network of domestic and European services from a number of airports in the United Kingdom on behalf of British Airways. …	**Search 2.** *2011 census population of Didsbury, Manchester* **Retrieved passages (original excerpts):** **Doc 1 (Didsbury):** … Didsbury is a suburban area of Manchester, England, on the north bank of the River Mersey, south of Manchester city centre. The population at the 2011 census was **26,788**. …
Search 3. *Population of Didsbury, Manchester in 2011 census* **Retrieved passages (original excerpts):** **Doc 1 (Didsbury):** … The population at the 2011 census was **26,788**. …	*The agent has already returned its final answer.*
Final answer: 26,788 Correct (EM = 1); 3 search calls.	**Final answer: 26,788** Correct (EM = 1); 2 search calls.

F.2 MAINTAINING THE CORRECT ENTITY ACROSS RETRIEVAL HOPS

Table 14 shows how retrieving the correct bridge entity supports multi-hop search. Before R3PO training, the initial results concern corporate executives rather than the film's producers. A later result introduces Michael E. Uslan in an executive-producer role, and the reasoner follows this entity in its next query; the trajectory ends with the unsupported answer Amazon. After R3PO training, the first result explicitly names Peter Guber alongside Jon Peters, and the second establishes his connection to Mandalay Entertainment. R3PO thus enables the fixed reasoner to complete the required film–producer–company chain in two searches instead of three.

Table 14: Retrieving the correct producer supports a complete two-hop evidence chain.

Question: The 1989 Batman film was produced by Jon Peters and a man that is the CEO of what company?

Before R3PO retriever training	**After R3PO retriever training**
Search 1. *The 1989 Batman film was produced by Jon Peters and a man that is the CEO of what company?* **Retrieved passages (original excerpts):** **Doc 2 (Jeff Bezos):** … He is the founder, chairman, and chief executive officer (CEO) of **Amazon**. …	**Search 1.** *The 1989 Batman film was produced by Jon Peters and a man that is the CEO of what company?* **Retrieved passages (original excerpts):** **Doc 1 (Batman (1989 film)):** … Batman is a 1989 American superhero film directed by Tim Burton and produced by **Jon Peters and Peter Guber**, based on the DC Comics character of the same name. …
Search 2. *Who was the co-producer of the 1989 Batman film with Jon Peters?* **Retrieved passages (original excerpts):** **Doc 1 (Michael E. Uslan):** … Uslan is best known as an **executive producer** with Benjamin Melniker of all of the modern Batman films to date, starting with Tim Burton's 1989 film, …	**Search 2.** *Peter Guber is the CEO of what company?* **Retrieved passages (original excerpts):** **Doc 1 (Peter Guber):** … He is Chairman and CEO of **Mandalay Entertainment**. …
Search 3. *Michael E. Uslan is the CEO of what company?* **Retrieved passages (original excerpts):** **Doc 1 (Microsoft):** … Microsoft Corporation (MS) is an American multinational technology company with headquarters in Redmond, Washington. … **Doc 2 (Apple Inc.):** … Apple is the world's largest information technology company by revenue and the world's third-largest mobile phone manufacturer after Samsung and Huawei. …	*The agent has already returned its final answer.*
Final answer: Amazon Incorrect (EM = 0); 3 search calls.	**Final answer: Mandalay Entertainment** Correct (EM = 1); 2 search calls.

F.3 DISTINGUISHING ANSWER RELEVANCE FROM TOPICAL OVERLAP

Table 15 isolates a change in retrieved evidence under an identical query and search budget. Before R3PO training, the results foreground Klose's all-time scoring record, which was broken during the 2014 World Cup. This is topically related but does not answer who scored most in that tournament; the returned Klose passage even distinguishes his two goals in 2014 from his career total. After R3PO training, tournament-specific statistics explicitly identify James Rodríguez and his six-goal Golden Boot. The answer changes from Miroslav Klose to James Rodríguez with one search in both runs, illustrating improved alignment between retrieved evidence and the precise question.

Table 15: Tournament-specific evidence corrects the answer under an unchanged one-search budget.

Question: who was the top scorer in 2014 world cup

Before R3PO retriever training	**After R3PO retriever training**
Search 1. *who was the top scorer in 2014 world cup*	**Search 1.** *who was the top scorer in 2014 world cup*
Retrieved passages (original excerpts): **Doc 1 (FIFA World Cup):** (2002–2014) is the **all-time top scorer** at the finals, with 16 goals. He broke Ronaldo of Brazil's record of 15 goals (1998–2006) during the 2014 semi-final match against Brazil. . . . **Doc 2 (Miroslav Klose):** . . . He also scored four times in the 2010 World Cup and twice at the 2014 World Cup, the latter tournament where he overtook Ronaldo's then-record of 15 goals to top the **all-time list**. . . .	**Retrieved passages (original excerpts):** **Doc 1 (2014 FIFA World Cup statistics):** . . . **James Rodríguez** was awarded the Golden Boot for scoring six goals, the first time that a Colombian player received the award. . . . **Doc 2 (2014 FIFA World Cup statistics):** . . . James Rodríguez was awarded the Golden Boot for scoring **six goals**, the first time that
Final answer: Miroslav Klose Incorrect (EM = 0); 1 search call.	**Final answer: James Rodríguez** Correct (EM = 1); 1 search call.

F.4 PREVENTING PROPAGATION OF AN INCORRECT ENTITY ASSOCIATION

Table 16 illustrates how a partially matching passage can redirect an entire trajectory. Before R3PO training, repeated retrieval of *Marriage Lines* preserves Richard Briers but substitutes Prunella Scales for the actor named in the question. The third query explicitly adopts this incorrect association, and additional searching reinforces the wrong answer. After R3PO training, the same first query retrieves a passage that names both the sitcom and Wilton's role as the wife of Briers's character. The correct answer, *Ever Decreasing Circles*, follows after one search rather than three. The relevant improvement is retrieval of the complete requested relation, not merely another passage mentioning Briers.

Table 16: Direct evidence for the requested actor relationship prevents an erroneous search continuation.

Question: In which sitcom did Penelope Wilton play the wife of Richard Briers?

Before R3PO retriever training	**After R3PO retriever training**
Search 1. *In which sitcom did Penelope Wilton play the wife of Richard Briers*	**Search 1.** *In which sitcom did Penelope Wilton play the wife of Richard Briers*
Retrieved passages (original excerpts): **Doc 1 (Marriage Lines):** … The series gave Richard Briers and **Prunella Scales**, its lead stars, a significant boost in their careers. …	**Retrieved passages (original excerpts):** **Doc 2 (Penelope Wilton):** … She is known for starring opposite Richard Briers in the BBC sitcom "**Ever Decreasing Circles**" (1984–89); … **Doc 3 (Penelope Wilton):** situation comedy, "Ever Decreasing Circles", which ran for five years. **She played Ann, long suffering wife of Martin (Briers)**, an obsessive and pedantic "do-gooder". …
Search 2. *Penelope Wilton played the wife of Richard Briers in which sitcom*	*The agent has already returned its final answer.*
Retrieved passages (original excerpts): **Doc 1 (Marriage Lines):** … The series gave Richard Briers and **Prunella Scales**, its lead stars, a significant boost in their careers. …	
Search 3. *Prunella Scales and Richard Briers in Marriage Lines*	*The agent has already returned its final answer.*
Retrieved passages (original excerpts): **Doc 3 (Richard Briers):** … In 1961, Briers was cast in the leading role in "Marriage Lines" (1961–66) with **Prunella Scales** playing his wife. …	
Final answer: Marriage Lines Incorrect (EM = 0); 3 search calls.	**Final answer: Ever Decreasing Circles** Correct (EM = 1); 1 search call.