RAMP: Adapting One Agent to Multiple Retrievers via Behavioral Probing

Anonymous ACL submission

Abstract

Agentic RAG agents must generate effective queries across diverse retriever backends, yet current agents produce retriever-oblivious queries, leading to large performance gaps when the backend changes. We observe that different retrievers, despite dissimilar architectures, exhibit structured behavioral patterns in their document-return overlap, forming a continuous behavioral space that can guide retriever-adaptive query generation. We propose RAMP (**R**etriever-**A**daptive **M**ulti-retriever **P**olicy), which probes a black-box retriever with shared queries, encodes the resulting overlap patterns into a compact behavioral embedding, and injects it as soft tokens to condition query generation. A single RAMP model replaces four retriever-specific specialists without per-retriever retraining and generalizes to unseen retrievers by interpolating in behavioral space. On four QA benchmarks, RAMP matches 96–98% of four retriever-specific specialists (+3.8 EM over unconditioned training) and reaches 91.8% of the retrained upper bound on unseen retrievers, outperforming routing, few-shot, and fusion alternatives.

1 Introduction

Agentic RAG systems train LLM-based agents to interleave multi-step reasoning with retrieval, achieving strong performance on knowledge-intensive tasks (Lewis et al., 2020; Gao et al., 2024; Jin et al., 2025; Song et al., 2025; Jiang et al., 2025). In deployment, however, retrieval backends are heterogeneous: organizations maintain lexical indices (Robertson and Zaragoza, 2009), learned-sparse models (Formal et al., 2022), and dense encoders (Karpukhin et al., 2020; Wang et al., 2024) side by side—each rewarding a fundamentally different query style (Thakur et al., 2021). BM25 favors precise keywords (*"actor born Springfield"*), SPLADE rewards expanded terms (*"actor born Springfield movie role"*), and dense encoders prefer natural-language questions (*"Which actor was born in Springfield?"*). An agent trained with one retriever produces queries tailored to that retriever's preference; when deployed against a different backend, performance drops sharply—not because the new retriever is weaker, but because the query is wrong for it (Hu et al., 2026; Chen et al., 2025).

This query–retriever mismatch lacks a satisfactory solution. Training a separate specialist per retriever (Jin et al., 2025) produces strong in-distribution performance but requires N independent models for N backends, with no knowledge transfer across retriever types. Retriever routing (Lee et al., 2024) and Reciprocal Rank Fusion (Cormack et al., 2009) select or merge result lists, yet leave the query unchanged—the mismatch persists because the query itself is never adapted. Query reformulation (Ma et al., 2023) and adaptive retrieval methods (Jiang et al., 2023; Asai et al., 2024) do modify query or retrieval behavior, but condition on the question or generation state rather than on retriever identity, so they cannot distinguish retriever-specific preferences. The root cause shared by all these approaches is that they treat each retriever as an isolated, discrete entity. None models the continuous behavioral relationship among retrievers—the very structure that would enable an agent to generalize its query strategy to retrievers it has never seen during training.

We observe that different retrievers, despite dissimilar architectures, reveal structured and consistent behavioral patterns when probed with shared queries: their document-return distributions cluster into a smooth, low-dimensional space in which behavioral distance correlates with the degree of query-style adaptation needed. This observation motivates RAMP (**R**etriever-**A**daptive **M**ulti-retriever **P**olicy), a conditioning method that encodes a black-box retriever's document-return patterns into a compact behavioral embedding and injects it as soft tokens to condition all subsequent

query generation. Concretely, we send a fixed set of probe queries to the target retriever, compute Jaccard overlap against training retrievers, and project the resulting *behavioral profile*—an 11-dimensional feature vector—through a learned encoder into the embedding space; the entire process requires a single offline probe pass and no access to the retriever's architecture or parameters. In contrast to approaches that require per-retriever retraining or treat retriever identity as a discrete routing decision, RAMP adapts queries in a continuous space and generalizes to unseen retrievers by interpolating along the behavioral spectrum. The conditioning mechanism is orthogonal to the choice of agentic RL backbone and composes with any search-agent training framework.

Our contributions are as follows:

1. We introduce *Retriever Behavioral Probing*, a backbone-agnostic conditioning mechanism that characterizes black-box retrievers through document-return overlap alone.
2. A single RAMP model achieves 96–98% of four retriever-specific specialists while outperforming unconditioned multi-retriever training by +3.8 EM on average (+4.6 on HotpotQA), replacing four models with one.
3. RAMP generalizes zero-shot to unseen retrievers, reaching 91.8% of the retrained upper bound and outperforming nearest-neighbor specialist routing, few-shot exemplar conditioning, and retrieval fusion alternatives.

2 Related Work

RL-Trained Search Agents. Recent work trains LLM-based agents to interleave reasoning with retrieval via reinforcement learning. SEARCH-R1 (Jin et al., 2025) introduced RL training for search-augmented LLMs, where the agent learns to generate effective search queries through outcome-based reward. R1-Searcher (Song et al., 2025) improves training stability with a two-stage curriculum, and DeepRetrieval (Jiang et al., 2025) extends the paradigm to real web search engines. WebThinker (Li et al., 2025) trains deep research agents via preference optimization to dynamically navigate and extract web information. These systems advance search agent capabilities along distinct axes—training efficiency, reasoning depth, and tool diversity—but provide no explicit mechanism for adapting query generation when the retrieval backend changes. RAMP operates within the same training paradigm but removes the fixed-retriever assumption: a single agent adapts its query generation to whichever retriever it faces.

Multi-Retriever Strategies. A separate line of work addresses scenarios where multiple retrievers are available, but operates at levels that leave query generation unchanged. Reciprocal Rank Fusion (RRF; Cormack et al., 2009) merges ranked lists from multiple retrievers into a single result set, improving recall without modifying the query sent to any individual system. Retriever routing (Lee et al., 2024) learns to direct each query to the most suitable retriever from a pool of domain-specific experts; MoR (Kalra et al., 2025) extends this to dynamically mix sparse, dense, and human retrievers per query. These approaches optimize *which retriever to use* or *how to combine results*, but the query itself remains identical regardless of which retriever receives it. Query reformulation (Ma et al., 2023) does adapt the query, but for a single fixed retriever—it improves phrasing without conditioning on retriever identity. RAMP operates at the query generation level: it produces different queries for different retrievers, conditioned on each retriever's observed behavior.

Conditioned Generation and Tool Use. Prompt tuning (Lester et al., 2021), prefix tuning (Li and Liang, 2021), and ToolkenGPT (Hao et al., 2023) show that learned soft tokens or tool embeddings can steer LLM behavior without modifying weights. Tool-augmented LLMs (Schick et al., 2023; Qin et al., 2024; Patil et al., 2023) learn to *select* the right API but do not adapt generation style to a tool's behavioral characteristics. In meta-RL, PEARL (Rakelly et al., 2019) and VariBAD (Zintgraf et al., 2020) condition policies on latent task variables inferred from interaction trajectories. Adaptive retrieval methods (Jiang et al., 2023; Asai et al., 2024; Shi et al., 2024) adapt *when* to retrieve but assume a fixed retriever. RAMP applies the conditioning principle to retriever characterization, deriving the signal from document-return patterns via a single offline probe pass.

3 Method

Given a retrieval-augmented agent and a target retriever accessible only through its query-in, documents-out API, our goal is to adapt the agent's query generation to the target retriever's

preference without maintaining a separate model per retriever. RAMP achieves this through three components. §3.1 describes *Retriever Behavioral Probing*, which characterizes a black-box retriever by probing it with shared queries and encoding its document-return patterns into a compact profile. §3.2 presents *Conditioned Query Generation*, which transforms this profile into soft tokens prepended to the agent's input, conditioning all subsequent query generation on the retriever's behavior. §3.3 details the *Multi-Retriever RL Training* procedure that jointly optimizes the agent backbone, behavioral encoder, and soft-token projector across multiple retrievers. Figure 1 provides an overview.

3.1 Retriever Behavioral Probing

Retrieval backends are typically black-box services exposing only a query-in, documents-out interface. We therefore characterize each retriever by *what* it returns rather than *how* it is built: two retrievers that return similar document sets for the same probe queries are likely to support similar query strategies, regardless of their internal mechanisms.

Given N training retrievers $\mathcal{R}_{\text{train}} = \{R_1, \ldots, R_N\}$, where each $R_i : \mathcal{Q} \to \mathcal{D}^k$ maps a query to k documents, we characterize each retriever by sending a shared set of probe queries and comparing the returned document sets across retrievers via Jaccard similarity. The comparison operates at the document-set level: for the same probe query, we measure overlap in *which* documents different retrievers return, treating the top-k results as an unordered set rather than conditioning the agent on retrieved document texts themselves (we compare alternative representations in Appendix B.3).

Behavioral Distance. We quantify retriever similarity using a shared probe set $\mathcal{Q}_P = \{q_1, \ldots, q_M\}$. The *behavioral distance* between retrievers R_i and R_j is:

$$d_B(R_i, R_j) = 1 - \frac{1}{M} \sum_{m=1}^{M} J\big(R_i(q_m),\, R_j(q_m)\big) \tag{1}$$

where $J(A, B) = |A \cap B| / |A \cup B|$ is Jaccard similarity over the returned document sets. This is a pseudometric preserving symmetry, non-negativity, and the triangle inequality (Lipkus, 1999) (distinct retrievers can have zero distance if they return identical sets for all probes), and serves as the basis for our generalization analysis.

Behavioral Profile. Given $M = 500$ probe queries (from Natural Questions), we compute an $(N{+}7)$-dimensional *behavioral profile* $\mathbf{b}_R \in \mathbb{R}^{N+7}$ for any retriever R—training or unseen—using the N training retrievers as fixed reference anchors:

1. **Reference agreement** (Nd): mean Jaccard similarity against each training retriever, $\bar{J}(R, R_j) = \frac{1}{M} \sum_m J(R(q_m), R_j(q_m))$ for $j = 1, \ldots, N$.
2. **Document diversity** (3d): title vocabulary diversity, document length coefficient of variation, and unique title ratio across all probe results.
3. **Query–document alignment** (3d): query-term coverage, lexical matching score, and semantic similarity.
4. **Behavioral uniqueness** (1d): fraction of documents returned exclusively by R and not by any reference retriever.

In our main experiments, $N{=}4$, yielding an 11-dimensional profile. The profile definition is *uniform*: both training and unseen retrievers produce a fixed-dimensional vector against the same reference anchors, enabling zero-shot transfer to new retrievers with only a single offline probe pass and no retraining. Reference agreement (Nd) provides the core discriminative signal; the remaining dimensions capture complementary aspects of retriever behavior.

Behavioral Encoder. The profile is transformed into a behavioral embedding via a learnable encoder ϕ:

$$\mathbf{e}_R = \phi(\mathbf{b}_R) \in \mathbb{R}^d \tag{2}$$

where ϕ is a 2-layer MLP ($(N{+}7) \to 64 \to 64$, ReLU; $11 \to 64 \to 64$ in our experiments). Because $\mathbf{e}_R$ conditions the policy likelihoods in the GRPO objective, policy-gradient updates propagate through the soft-token projector back to ϕ, allowing the encoder to learn which behavioral features matter most for query adaptation. Probe data is computed offline once per retriever and the profile is robust to hyperparameter choices (Appendix B.5).

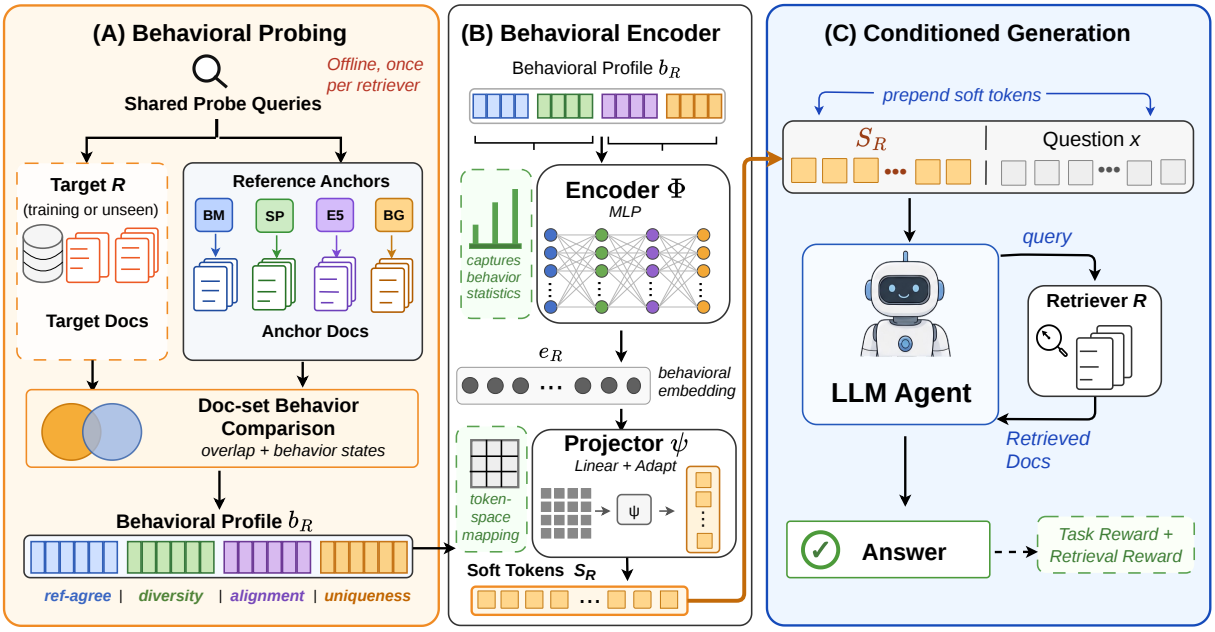

Figure 1: Overview of RAMP. **(A)** Behavioral Probing: shared probe queries are sent to both the target retriever and N reference anchors; document-set overlap and behavioral statistics are aggregated into an 11-dimensional behavioral profile. **(B)** Behavioral Encoder: a learned encoder compresses the profile into a behavioral embedding, which a two-stage projector maps to k soft tokens in the LLM's embedding space. **(C)** Conditioned Generation: the soft tokens are prepended to the agent's input, conditioning query generation on the retriever's behavior; task and retrieval rewards jointly optimize the encoder, projector, and policy via GRPO.

3.2 Conditioned Query Generation

Rather than conditioning on a discrete retriever-ID token—which cannot represent unseen backends—we project the continuous behavioral embedding $\mathbf{e}_R$ into k soft tokens in the LLM's embedding space and prepend them to the input, so that the retriever's behavioral signature conditions all subsequent query generation.

Soft Prompt Projection. The projection from $\mathbf{e}_R$ to soft tokens proceeds in two stages:

$$\mathbf{S}_R = \text{Adapt}\big(\text{reshape}(\mathbf{W}_p \mathbf{e}_R + \mathbf{b}_p)\big) \in \mathbb{R}^{k \times d_{\text{model}}} \tag{3}$$

In the first stage, a linear projector $\mathbf{W}_p \in \mathbb{R}^{(k \cdot d_{\text{mid}}) \times d}$ maps $\mathbf{e}_R$ into k vectors of dimension d_{mid}. This stage is responsible for preserving inter-retriever differentiation: the ratio between output and input dimensionality is kept small so that soft tokens remain well-separated across retrievers. In the second stage, a trainable linear layer $\text{Adapt} : \mathbb{R}^{d_{\text{mid}}} \to \mathbb{R}^{d_{\text{model}}}$ expands each vector to the LLM's hidden size; this stage handles dimension alignment without compressing the inter-retriever signal. Projecting directly from d to d_{model} in a single step causes all retrievers to produce nearly identical soft tokens (Appendix B.4). The resulting k tokens are prepended to the agent's input sequence at each generation step. This design requires only embedding-level access to the agent backbone and does not modify the autoregressive architecture.

3.3 Multi-Retriever RL Training

If the behavioral encoder were trained separately, it would not receive signal about which profile dimensions matter for downstream query quality. We therefore optimize the encoder, projector, and policy jointly under the same RL objective.

We train the conditioned policy $\pi_\theta(q \mid x, \mathbf{e}_R)$—where q denotes the full action sequence of interleaved reasoning and search queries—jointly with the behavioral encoder ϕ and the projection parameters ψ using Group Relative Policy Optimization (GRPO; Shao et al., 2024). The training objective is:

$$\max_{\theta,\phi,\psi} \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{x,\, q \sim \pi_\theta(\cdot \mid x, \mathbf{e}_{R_i})} \big[r(x, q, R_i) \big] \tag{4}$$

The behavioral profiles $\{\mathbf{b}_{R_i}\}$ are computed offline and remain fixed throughout training; only θ, ϕ, and ψ are updated, sharing the same learning rate and optimizer without staged or separate pre-training.

Retriever Scheduling. At each block boundary $t \in \{0, K, 2K, \ldots\}$, we sample a retriever index $i_t \sim \text{Uniform}(\{1, \ldots, N\})$; all K consecutive batches within the block use that retriever:

$$R_{\text{batch}_j} = R_{i_t}, \quad \forall\, j \in [t,\ t{+}K) \tag{5}$$

The uniform sampling ensures that each retriever receives approximately equal training coverage over the course of training. Within each block, the G rollouts used for advantage estimation all face the same retriever.

Reward. The per-trajectory reward combines a task signal and a retrieval signal:

$$r = r_{\text{task}} + \lambda \cdot r_{\text{ret}} \tag{6}$$

where $r_{\text{task}} \in \{0, 1\}$ is exact match of the final answer and r_{ret} is title recall against gold supporting facts:

$$r_{\text{ret}} = \frac{|\text{retrieved titles} \cap \text{gold titles}|}{|\text{gold titles}|} \in [0, 1] \tag{7}$$

The retrieved titles are the union of all documents retrieved across the multiple search steps within a single trajectory. We set $\lambda = 0.3$.

Optimization. For each question x and the block's retriever R_i, we sample G rollout trajectories from $\pi_\theta(\cdot \mid x, \mathbf{e}_{R_i})$ and compute group-normalized advantages $\hat{A}^{(g)}$. The clipped surrogate loss is:

$$\mathcal{L}(\theta, \phi, \psi) = -\mathbb{E}_{x,\, R_i}\Big[\frac{1}{G}\sum_{g=1}^{G} \min\big(\rho^{(g)}\hat{A}^{(g)},$$

$$\text{clip}(\rho^{(g)}, 1{\pm}\varepsilon)\,\hat{A}^{(g)}\big)\Big] + \beta\, D_{\text{KL}}[\pi_\theta \| \pi_{\text{ref}}] \tag{8}$$

where the importance ratio is

$$\rho^{(g)} = \frac{\pi_\theta\big(q^{(g)} \mid x,\, \mathbf{e}_{R_i}\big)}{\pi_{\text{old}}\big(q^{(g)} \mid x,\, \mathbf{e}_{R_i}\big)}. \tag{9}$$

Since $\mathbf{e}_{R_i}$ is produced by the learnable encoder ϕ and enters the policy through the projection ψ, the gradient of $\mathcal{L}$ propagates through the full chain $\phi \to \mathbf{e}_R \to \mathbf{S}_R \to \pi_\theta$, updating all three parameter groups jointly.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on four multi-hop QA benchmarks: HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). All use the Wikipedia 2018 (wiki-18) corpus with 21M passages.

Retrievers. We use four *training* retrievers spanning three paradigms—BM25 (Robertson and Zaragoza, 2009) (lexical), SPLADE-v2 (Formal et al., 2022) (learned sparse), E5 (Wang et al., 2024) and BGE-base (Xiao et al., 2024) (dense)—and four *unseen* retrievers for generalization evaluation: SPLADE++, Contriever (Izacard et al., 2022), E5-mistral-7B, and GTE-Qwen2-7B (Li et al., 2023). Pairwise behavioral distances d_B (Definition 3.1) to the nearest training retriever are reported in §4.3.

Baselines. All methods share the same base model (Qwen3-8B-Instruct; Qwen Team, 2025), training recipe, and evaluation protocol (see Appendix D for reproduction details). We compare along two axes—*single- vs. multi-retriever training* and *conditioning type*: (1) **Naive Agent**: no RL training; (2) **SEARCH-R1**: single-retriever specialist (one model per retriever); (3) **SEARCH-R1×4**: oracle ensemble of all four specialists; (4) **SEARCH-R1-Multi**: multi-retriever training, no conditioning; (5) **RAMP (name)** and (6) **RAMP (desc.)**: multi-retriever training with textual conditioning (retriever name or natural-language description). Three additional cross-paradigm alternatives (NN Specialist Routing, Few-Shot Exemplar, RRF) are introduced in §4.3.

Implementation. We instantiate RAMP within a SEARCH-R1-style agentic training environment (Jin et al., 2025). All models are trained on the merged NQ + HotpotQA training set. Full hyperparameters, reward design, and reproducibility details are in Appendix D. We report Exact Match (EM) as the primary metric, with F1 and Recall@10 in Appendix C.

4.2 Core Results: One Model Across Many Retrievers

Table 1 and Figure 2 present results across all four benchmarks. The results validate our central hypothesis: retriever-adaptive conditioning enables a single model to approach specialist-level performance without per-retriever retraining.

The query–retriever mismatch problem is severe. The specialist transfer matrix (Figure 2a) quantifies the cost of serving a mismatched retriever: a BM25-trained specialist loses up to 12.2 EM when deployed on a dense retriever, because its keyword-style queries fail to activate semantic matching. The mismatch is asymmetric—dense-to-dense transfer (E5↔BGE) loses relatively little,

Method	#M	HotpotQA					2WikiMultihopQA					MuSiQue					Bamboogle				
		BM	SPL	E5	BGE	Avg	BM	SPL	E5	BGE	Avg	BM	SPL	E5	BGE	Avg	BM	SPL	E5	BGE	Avg
Naive Agent	0	31.2	27.4	29.1	28.6	29.1	28.4	23.6	24.9	24.4	25.4	16.2	11.8	13.2	13.1	13.6	35.2	32.0	32.8	34.4	33.6
SEARCH-R1 (BM25)	1	**46.4**	35.1	34.4	33.9	37.5	**44.8**	31.8	30.9	34.2	35.5	**21.2**	14.6	14.2	17.4	16.9	**45.6**	38.4	36.8	43.2	41.0
SEARCH-R1 (SPL)	1	35.8	**42.1**	35.4	35.6	37.2	33.2	**39.8**	33.4	33.8	35.1	15.6	**19.6**	14.9	15.6	16.5	40.8	**46.4**	37.6	42.4	41.8
SEARCH-R1 (E5)	1	34.9	36.2	**42.8**	39.2	38.3	32.4	34.1	**40.4**	36.8	35.9	15.1	15.8	**19.4**	16.9	16.8	38.4	40.0	**45.6**	44.0	42.0
SEARCH-R1 (BGE)	1	34.2	35.6	39.4	**42.1**	37.8	32.9	33.6	37.1	**40.6**	36.1	15.2	15.6	17.4	**19.4**	16.9	37.6	39.2	42.4	**47.2**	41.6
SEARCH-R1-Multi	1	39.6	36.9	37.4	36.8	37.7	38.1	35.4	34.9	34.8	35.8	18.8	16.1	16.4	16.4	16.9	41.6	40.0	40.8	40.0	40.6
RAMP (name)	1	43.4	40.2	40.4	39.9	41.0	40.8	38.1	37.9	37.2	38.5	19.2	17.1	17.2	17.2	17.7	43.2	43.2	42.4	42.4	42.8
RAMP (desc.)	1	43.8	40.8	41.1	40.4	41.5	41.6	38.6	38.9	38.4	39.4	19.6	17.6	18.1	17.8	18.3	44.0	44.0	44.0	43.2	43.8
RAMP (ours)	1	44.9	41.3	41.8	41.1	**42.3**	42.7	39.3	39.6	38.4	**40.0**	19.9	18.6	19.1	18.8	**19.1**	44.0	44.8	44.8	44.8	**44.6**
SEARCH-R1×4 (oracle)	4	46.4	42.1	42.8	41.9	43.3	44.8	39.8	40.4	40.6	41.4	21.2	19.6	19.4	19.4	19.9	45.6	46.4	45.6	47.2	46.2

Table 1: Main results (EM) across four benchmarks. BM = BM25, SPL = SPLADE-v2.

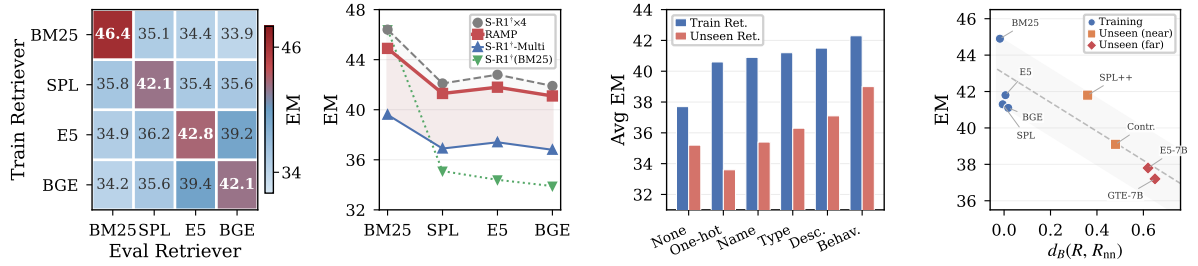

Figure 2: Visual summary on HotpotQA. (a) Cross-retriever transfer matrix: specialists degrade off-diagonal. (b) Per-retriever EM for RAMP (red), S-R1(BM25) (green), and the 4-specialist oracle (grey). (c) Conditioning signal comparison on training and unseen retrievers. (d) Generalization vs. behavioral distance d_B.

while lexical-to-dense transfer suffers the largest drops—because dense encoders tolerate keyword-like inputs but BM25's exact-match scoring is brittle to natural-language phrasing. This structure motivates conditioning: any solution must bridge the paradigm gap, not merely average over retriever-specific query styles.

Conditioning resolves the multi-retriever conflict. Without conditioning, multi-retriever training (SEARCH-R1-Multi) averages over conflicting query preferences, improving cross-retriever robustness but underperforming each specialist on its matched retriever. Adding a conditioning signal lets the policy modulate its query style per retriever, progressively closing the gap as the signal captures finer behavioral differences (Figure 2c): name conditioning reaches 41.0 avg EM on HotpotQA, description 41.5, and behavioral probing 42.3 vs. the specialist oracle at 43.3. The gradient across signal types indicates that *what* the model knows about the retriever matters more than *whether* it receives any signal at all.

One model recovers specialist-level performance. With behavioral conditioning, a single RAMP model closes the gap to the 4-specialist oracle to within 1–2 EM per retriever across all four benchmarks (Figure 2b), reaching 96–98% of oracle performance with one set of weights instead of four. The residual gap concentrates on BM25, whose keyword-centric preference diverges most from the dense-retriever queries that dominate multi-retriever training; on the three non-BM25 retrievers, RAMP reaches 95–99% of each specialist. Answer F1 and Retrieval Recall@10 confirm the same trends (Appendix C).

Method	SPL++ $d_B=0.36$	Cont. $d_B=0.48$	E5-7B $d_B=0.62$	GTE $d_B=0.71$	Avg
Naive Agent	27.9	25.4	29.4	28.6	27.9
S-R1 (BM25)	36.6	31.2	31.8	31.9	32.9
S-R1-Multi	38.2	34.1	34.4	33.9	35.2
RAMP (one-hot)	37.1	32.8	31.9	32.4	33.6
RAMP (name)	38.9	34.1	34.2	34.4	35.4
RAMP (desc.)	40.2	36.1	35.9	36.2	37.1
RAMP (type)	39.8	36.4	35.1	33.9	36.3
RAMP (behav.)	**41.8**	**39.1**	**37.8**	**37.2**	**39.0**
S-R1 (retrained)	42.8	40.4	44.1	42.8	42.5

Table 2: Generalization to unseen retrievers on HotpotQA (EM), ordered by behavioral distance d_B (shown below each retriever name). S-R1 (retrained) = per-retriever upper bound.

4.3 Generalization and Alternative Baselines

A unique advantage of behavioral probing is that it enables *zero-shot transfer* to unseen retrievers:

Method	Type	Train	Unseen	Cost	U-R
S-R1-Multi	None	37.7	35.2	1×	✓
Few-Shot Exemp.	ICL	40.1	36.2	1×	△
RRF (4-ret.)	Fusion	40.8	N/A	4×	×
NN Spec. Route	Route	—	36.7	1×	✓
RAMP (desc.)	Text	41.5	37.1	1×	△
RAMP (behav.)	Behav.	**42.3**	**39.0**	1×	✓
S-R1×4	Oracle	43.3	42.5*	1×	×

Table 3: Cross-paradigm comparison on HotpotQA (EM). Train/Unseen = avg over 4 training/unseen retrievers. U-R = unseen-ready (✓ = zero-shot, △ = needs proxy, × = not applicable). *Retrained upper bound.

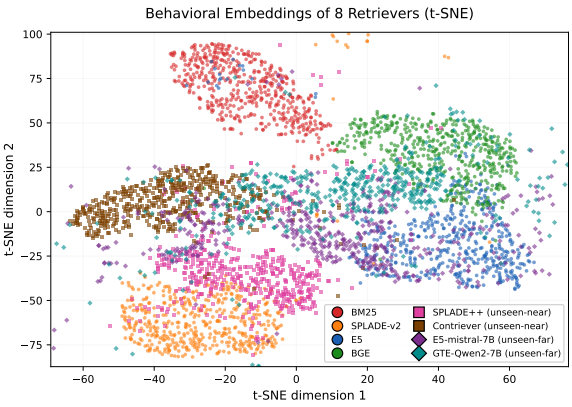

Figure 3: t-SNE (van der Maaten and Hinton, 2008) visualization of behavioral profiles for all 8 retrievers (4 training + 4 unseen). Same-type retrievers cluster together; unseen Contriever lies between the sparse and dense clusters. No unseen-retriever supervision is used.

	BM25	SPL.	E5	BGE
Avg. length (words)	4.4	6.2	11.1	9.3
NL sentence (%)	10	25	90	78
Stopword ratio (%)	7	16	38	33

Table 4: Query style analysis. RAMP produces keyword queries for BM25 and natural-language questions for dense retrievers.

compute the new retriever's behavioral profile, encode it, and generate adapted queries—no retraining required. Table 2 evaluates this capability on four unseen retrievers, ordered by ascending behavioral distance d_B to the nearest training retriever.

Behavioral probing enables strong generalization. RAMP (behavioral) reaches 91.8% of the retrained upper bound on average (39.0 vs. 42.5 EM), and the gap degrades gracefully with behavioral distance rather than failing abruptly (Figure 2d). Retrievers with low behavioral distance (SPLADE++, $d_B = 0.36$) recover 97.7% of retrained performance; those with higher distance (E5-7B, GTE, $d_B > 0.6$) still recover 85–87%. This gradient is consistent with our calibrated generalization bound (Theorem 4, Appendix A.3).

Alternative baselines cannot match RAMP. Table 3 compares three cross-paradigm alternatives. NN Specialist Routing (selecting the nearest training specialist) remains 2.3 EM below RAMP on unseen retrievers because routing is brittle when no close specialist exists. Few-Shot Exemplar conditioning provides modest gains but transfers poorly to unseen retrievers, where proxy exemplars from the nearest training retriever offer diminishing returns. RRF achieves competitive training-retriever performance at 4× retrieval cost but is inapplicable to unseen single-retriever deployment, confirming that *query adaptation* is more effective than *result fusion*.

4.4 Analysis

What the model learns: behavioral embeddings preserve retriever structure. The behavioral embedding space organizes retrievers by their functional similarity rather than by architecture or name. The type structure metric $\rho = 0.83$ (point-biserial correlation) confirms that same-type retrievers are systematically closer in behavioral space than cross-type pairs (Appendix Table 13). This structure emerges from document-return overlap alone and transfers to unseen retrievers without any fine-tuning: Figure 3 shows that unseen SPLADE++ clusters near training SPLADE-v2, while unseen Contriever falls between the sparse and dense clusters. The behavioral proximity directly predicts generalization: SPLADE++ ($d_B = 0.36$) achieves the highest zero-shot performance among unseen retrievers (41.8 EM, Table 2).

What the model generates: retriever-adapted query styles. We analyze 500 queries generated for the same set of questions under each retriever's conditioning (Table 4). The query style varies continuously with the conditioning signal: BM25-conditioned queries are concise keywords (4.4 words, 10% natural-language sentences), E5-conditioned queries are full questions (11.1 words, 90% NL sentences), and SPLADE-conditioned queries fall in between with moderate term expansion. This confirms that RAMP does not merely route to a binary sparse/dense mode but interpolates along the behavioral spectrum to produce a query style calibrated to each retriever's preference.

Variant	Train	Unseen
RAMP (full model)	**42.3**	**39.0**
– behavioral → description	41.5	37.1
– behavioral → type label	41.2	36.3
– behavioral → name	40.9	35.4
– behavioral → one-hot	40.6	33.6
– all conditioning (= S-R1-Multi)	37.7	35.2
– $\mathcal{R}_{\text{retrieval}}$ ($\lambda = 0$)	40.7	37.4
– trainable ϕ (frozen encoder)	41.1	37.5
– SPLADE in training	41.8	37.1

Table 5: Ablation study on HotpotQA (average EM). "Train" = average over 4 training retrievers. "Unseen" = average over 4 unseen retrievers.

Why behavioral probing outperforms textual conditioning. Two data patterns point to *signal content* as the source of the gap. First, the advantage widens on unseen retrievers: 0.8 EM on training retrievers (42.3 vs. 41.5) but 1.9 EM on unseen ones (39.0 vs. 37.1)—a format-only effect would remain constant, yet the gap grows precisely where textual descriptions become approximate. Second, textual labels cannot separate behaviorally distinct retrievers within the same category: E5 and BGE are both "dense," yet $d_B = 0.59$ and their query styles differ measurably (Table 4), a distinction that only observation-derived embeddings can capture.

4.5 Ablations and Robustness

Table 5 isolates the contribution of each component. The conditioning signal ablations (top block) are analyzed in detail in §4.3 and §4.4; here we focus on the remaining design choices.

Retrieval reward. Removing $\mathcal{R}_{\text{retrieval}}$ ($\lambda = 0$) reduces performance by 1.6 EM on both training and unseen retrievers, because the encoder loses per-step supervision on which query patterns lead to relevant documents for each retriever.

Trainable encoder. Freezing the behavioral encoder costs 1.2 EM on training retrievers but 1.5 EM on unseen retrievers. The larger unseen gap suggests that end-to-end training teaches the encoder to emphasize behavioral dimensions that generalize across the retriever spectrum, not merely those that differentiate the four training retrievers.

Retriever diversity. Removing SPLADE from training reduces unseen-retriever performance by 1.9 EM. Without SPLADE, the training set loses its only learned-sparse representative; the behavioral space collapses to a lexical-vs-dense axis, and SPLADE++ — the unseen retriever closest to the removed SPLADE — suffers the largest individual drop. This confirms that paradigm diversity during training directly improves the coverage of the behavioral embedding space.

Backbone	Conditioning	Train	Unseen
GRPO	None	37.7	35.2
GRPO	RAMP	**42.3**	**39.0**
Reinforce++	None	38.8	35.4
Reinforce++	RAMP	**42.2**	**38.7**

Table 6: Backbone independence on HotpotQA (avg EM). GRPO = SEARCH-R1-style single-stage; Reinforce++ = R1-Searcher-inspired two-stage RL.

Backbone independence. To verify that RAMP's gains are not specific to the SEARCH-R1 training framework, we re-implement the conditioning mechanism within a two-stage RL setup inspired by R1-Searcher (Song et al., 2025), which uses Reinforce++ instead of GRPO and a two-stage reward curriculum. Table 6 shows consistent gains under both backbones (+4.6/+3.8 with GRPO, +3.4/+3.3 with Reinforce++), confirming that the conditioning mechanism is orthogonal to the RL algorithm. Sensitivity analysis of behavioral profile design is in Appendix B.5.

5 Conclusion

We presented RAMP, a conditioning method that enables a single agentic RAG model to generate retriever-adapted queries by encoding a black-box retriever's document-return patterns into soft tokens—without modifying the agent architecture or training algorithm. The key design choice is *behavioral probing*: characterizing each retriever through the document sets it returns for shared probe queries, producing a compact profile that applies uniformly to training and unseen retrievers alike.

A single RAMP model matches 96–98% of four retriever-specific specialists, generalizes zero-shot to unseen retrievers at 91.8% of the retrained upper bound, and transfers across RL backbones. The core insight is that retriever preferences form a continuous behavioral spectrum navigable through observation alone, a principle we expect to extend to other tool-use scenarios with heterogeneous backends.

Limitations

Task scope. We evaluate RAMP exclusively on multi-hop question answering, where query reformulation is a natural bottleneck. Whether the behavioral probing framework transfers to other retrieval-intensive tasks—such as fact verification, open-domain dialogue, or long-form generation with citation—remains to be validated. The core conditioning mechanism is task-agnostic, but the reward design and training distribution may require adaptation for different downstream objectives.

Retriever coverage during probing. Behavioral profiles are constructed from a fixed set of shared probe queries drawn from a single corpus (Wikipedia 2018). Retrievers deployed on substantially different domains (e.g., biomedical or legal corpora) may exhibit behavioral patterns not well captured by Wikipedia-based probes, potentially requiring domain-specific probe sets for optimal profiling.

Base model scale. All experiments use a single base model scale (Qwen3-8B). While the conditioning mechanism is architecture-agnostic and the backbone independence experiment (§4.5) confirms algorithm-level generality, we have not verified whether the gains hold at substantially larger or smaller model sizes, where the base model's intrinsic ability to adapt query style may differ.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of the 12th International Conference on Learning Representations*.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. 2025. BrowseComp-Plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*.

Gordon V Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 758–759.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From distillation to hard negative sampling: Making sparse neural IR models more effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2353–2359.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. 2023. Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings. In *Advances in Neural Information Processing Systems*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Tiansheng Hu, Yilun Zhao, Canyu Zhang, Arman Cohan, and Chen Zhao. 2026. SAGE: Benchmarking and improving retrieval for deep research agents. *arXiv preprint arXiv:2602.05975*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. *arXiv preprint arXiv:2503.00223*.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.

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

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Jushaan Singh Kalra, Xinran Zhao, To Eun Kim, Fengyu Cai, Fernando Diaz, and Tongshuang Wu. 2025. MoR: Better handling diverse queries with a mixture of sparse, dense, and human retrievers. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergei Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781.

Hyunji Lee, Luca Soldaini, Arman Cohan, Minjoon Seo, and Kyle Lo. 2024. Routerretriever: Exploring the benefits of routing over multiple expert embedding models. *arXiv preprint arXiv:2409.02685*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 4582–4597.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. 2025. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2356–2362.

Alan H Lipkus. 1999. A proof of the triangle inequality for the Tanimoto distance. *Journal of Mathematical Chemistry*, 26(1-3):263–265.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *arXiv preprint arXiv:2305.14283*.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2023. Gorilla: Large language model connected with massive APIs. *arXiv preprint arXiv:2305.15334*.

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

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerber, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. Toolllm: Facilitating large language models to master 16000+ real-world APIs. In *Proceedings of the 12th International Conference on Learning Representations*.

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Kate Rakelly, Aurick Zhou, Deirdre Quillen, Chelsea Finn, and Sergey Levine. 2019. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 8371–8384.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 641–649.

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

Luisa M Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. 2020. VariBAD: A very good method for Bayes-adaptive deep RL via meta-learning. In *Proceedings of the 8th International Conference on Learning Representations*.

A Theoretical Analysis

This appendix provides theoretical analysis on three aspects: (1) the relationship between conditioning signal quality and performance (§A.1), (2) probe query sample complexity (§A.2), and (3) performance degradation bounds for unseen retrievers (§A.3).

A.1 Identification-Regret Framework

Cross-Evaluation Matrix. When specialist j serves retriever i, the performance gap relative to the matched specialist defines the *cross-retriever regret*. Table 7 shows the full regret matrix computed from Table 1 (HotpotQA); the worst-case regret is $\Delta_{\max} = 12.2$ EM.

	π^*_{BM}	π^*_{SPL}	π^*_{E5}	π^*_{BGE}	max
BM25	0	10.6	11.5	**12.2**	12.2
SPLADE	**7.0**	0	5.9	6.5	7.0
E5	**8.4**	7.4	0	3.4	8.4
BGE	**8.2**	6.5	2.9	0	8.2

Table 7: Cross-retriever regret $\Delta_{ij} = V_i(\pi^*_i) - V_i(\pi^*_j)$ (EM) on HotpotQA, where $V_i(\pi^*_j)$ is specialist j's performance on retriever i. $\Delta_{\max} = 12.2$ (BM25 retriever with BGE specialist).

Theorem 1 (Identification-Regret Bound)**.** *Let e be a conditioning signal with Bayes-optimal retriever classification error $P_{\text{err}}(e)$. For a parameterized conditioned policy class $\Pi^{\text{cond}}_\Theta = \{\pi_\theta(\cdot|x, e) : \theta \in \Theta\}$, the optimal achievable policy π^*_Θ satisfies:*

$$\bar{V}^{\text{expert}} - \bar{V}(\pi^*_\Theta) \le \underbrace{P_{\text{err}}(e) \cdot \Delta_{\max}}_{\text{identification error}} + \underbrace{\epsilon_{\text{cap}}(\Theta, e)}_{\text{capacity gap}} \tag{10}$$

where $\bar{V}^{\text{expert}} = \frac{1}{N}\sum_i V_i(\pi^*_i)$ and $\epsilon_{\text{cap}}(\Theta, e) = [\bar{V}(\pi^*_{\text{ITS}}) - \max_{\pi\in\Pi_\Theta} \bar{V}(\pi)]^+$ is the approximation error of the policy class.

Proof sketch. Construct an *Identify-Then-Specialize* (ITS) oracle: classify the retriever from e, then execute the matched specialist. Each misclassification incurs at most $\Delta_{\max}$, so the ITS gap is $\le P_{\text{err}}(e) \cdot \Delta_{\max}$. Since the learned policy may not perfectly implement ITS, adding the capacity gap ϵ_{cap} yields Eq. 10.

Remark (two-term decomposition). The first term $P_{\text{err}} \cdot \Delta_{\max}$ measures whether the signal carries enough *information* to identify retrievers—a property of the signal itself, independent of the model. The second term ϵ_{cap} measures whether the model can *utilize* the signal effectively—depending on signal form and model architecture compatibility. When $P_{\text{err}} = 0$ (behavioral probing and name both achieve this), the gap is entirely ϵ_{cap}: behavioral probing yields $\epsilon_{\text{cap}} = 0.9$ EM vs. name's 1.8 EM, suggesting that continuous embeddings are more effectively utilized by the policy network than discrete text tokens.

Corollary 2 (Fano Information-Theoretic Constraint)**.** *Under a uniform retriever prior, Fano's inequality lower-bounds the identification error in terms of the mutual information between the conditioning signal and the retriever identity:*

$$P_{\text{err}}(e) \;\ge\; \frac{\log N - I(e;\, R) - \log 2}{\log(N-1)} \tag{11}$$

To keep the identification-induced regret below ϵ, the signal must satisfy

$$I(e;\, R) \;\ge\; \log N - \log 2 - \frac{\epsilon \log(N-1)}{\Delta_{\max}} \quad \textit{(bits)}. \tag{12}$$

With 4 retrievers and $\Delta_{\max}$=12.2, keeping the identification error below 1 EM requires $I(e;R) > 0.87$ bit (43% of the 2-bit maximum).

Low-information content features face a large identification bottleneck: raw Transformer encoding achieves only 21% accuracy ($P_{\text{err}} = 0.79$), and even the best post-processed Transformer variant reaches only 46% (Table 12). Both remain far below document-set features, which reach 100% classification accuracy, confirming that abandoning content-level encoding (§3.1) is justified by the signal's information-theoretic limits rather than insufficient optimization.

Empirical verification. Table 8 verifies Theorem 1 across conditioning signals on HotpotQA ($\bar{V}^{\text{expert}} = 43.3$, $\Delta_{\max} = 12.2$).

Signal	P_{err}	Id. error	Actual gap	ϵ_{cap}
Behavioral	0.00	0.0	1.0	**1.0**
Description	∼0.00	0.0	1.8	**1.8**
Name	∼0.00	0.0	2.3	**2.3**
Unconditioned	0.75	9.2	5.6	†
Trans. enc.	0.79	9.6	‡	‡

Table 8: Empirical verification of Theorem 1. Id. error $= P_{\text{err}} \times \Delta_{\max}$; for $P_{\text{err}} = 0$ the gap reduces to capacity loss ϵ_{cap}. † When $P_{\text{err}} > 0$, ϵ_{cap} cannot be isolated because the bound uses worst-case $\Delta_{\max}$ rather than confusion-weighted regret. ‡ Transformer encoding was not used as a conditioning signal in full RL training; only classification accuracy is reported.

A.2 Probe Sample Complexity

Theorem 3 (Probe Sample Complexity)**.** *Let $d_{\min} = \min_{i \neq j} d_B(R_i, R_j)$ be the minimum pairwise behavioral distance among N training retrievers. Using M i.i.d. probe queries, a sufficient condition to correctly distinguish all retriever pairs with probability $\geq 1 - \delta$ is:*

$$M \geq \frac{2}{d_{\min}^2} \ln \frac{N(N-1)}{\delta} \tag{13}$$

Proof sketch. Each empirical distance is a bounded sample mean, so Hoeffding's inequality controls its deviation from the true distance. Setting the tolerance to half the minimum gap and applying a union bound over all retriever pairs yields the stated sample requirement.

The minimum behavioral distance among our four retrievers is 0.589 (E5–BGE pair, Table 11), giving a theoretical minimum of 32 probes (δ=0.05)—far below the 500 used in practice. This explains the robustness in Table 16: even 100 probes loses only 0.7 EM. Probe requirements grow logarithmically in the number of retrievers.

Remark (self-correcting probe requirement). Theorems 1 and 3 interact favorably: retriever pairs with small behavioral distance d_B require more probes to distinguish (Theorem 3), yet produce lower confusion regret (Theorem 1), because near-identical retrievers respond similarly to the same queries. Empirically, the closest pair (E5–BGE, d_B=0.589) incurs only 3.4 EM regret upon misidentification, while the most distant pair (BM25–BGE, d_B=0.817) incurs 12.2 EM (Table 7). The probe budget therefore concentrates on pairs where confusion is costly, while underprobing of similar retrievers is inherently low-risk.

A.3 Calibrated Generalization Bound

Theorem 4 (Calibrated Generalization Bound)**.** *For* RAMP *policy π_θ and unseen retriever R_{new} with nearest training retriever R_{nn}, assume four Lipschitz/smoothness conditions with constants L_r, L_b, L_ϕ, L_π (reward, profile, encoder, and policy, respectively). Then:*

$$\left|V(\pi_\theta, R_{\text{new}}) - V(\pi_\theta, R_{\text{nn}})\right| \leq C \cdot d_B(R_{\text{new}}, R_{\text{nn}}) \tag{14}$$

where $C = L_r + r_{\max} L_\pi L_\phi L_b$ and $r_{\max} = 1 + \lambda = 1.3$.

Proof sketch. Decompose the gap into **(I) reward drift** (bounded by $L_r \cdot d_B$ via S1) and **(II) policy drift** (chaining S2–S4 and applying the simulation lemma gives $r_{\max} L_\pi L_\phi L_b \cdot d_B$). Summing yields the stated bound.

Remark (practical scope). We treat Theorem 4 as a *diagnostic framework* rather than a worst-case guarantee: the Lipschitz constants in S1–S4 are not individually estimated, and the theoretical composite C may exceed the empirically observed value due to the looseness of S1 (EM is discontinuous per-sample, though Lipschitz in expectation) and S4 (autoregressive TV distance can grow with sequence length). We therefore calibrate C directly from data below, which yields tighter and more actionable predictions.

Empirical calibration of C. We estimate $\hat{C}$ from training retrievers using Table 1 and Table 11:

$$\hat{C} = \max_{i \neq j} \frac{\left|V(\pi_\theta, R_i) - V(\pi_\theta, R_j)\right|}{d_B(R_i, R_j)} \tag{15}$$

Pair	d_B	$\lvert V_i - V_j \rvert$	ratio
BM25–SPL	0.824	3.6	4.4
BM25–E5	0.799	3.1	3.9
BM25–BGE	0.817	3.8	**4.7**
SPL–E5	0.761	0.5	0.7
SPL–BGE	0.784	0.2	0.3
E5–BGE	0.589	0.7	1.2

Table 9: Calibration of $\hat{C}$ from training retrievers (HotpotQA). Leave-one-out range: $[3.9, 4.7]$.

This gives $\hat{C} = 4.7$, enabling forward predictions: for any unseen retriever R_u with behavioral distance d_u to its nearest training retriever, $V(R_u) \geq V(R_{\text{nn}}) - 4.7 \cdot d_u$.

Observation (non-linear degradation at high d_B). Low-d_B retrievers (SPLADE++, Contriever) reach 97.2% of retrained specialists, while high-d_B retrievers (E5-7B, GTE-7B) reach only 86.3% (Table 2)—degradation accelerates beyond the training-retriever anchor region. Expanding paradigm diversity of training retrievers (e.g., removing SPLADE drops unseen performance by 1.9 EM, Table 5) is the key to extending the effective generalization radius.

B Behavioral Probing: Design, Validation, and Sensitivity

This appendix provides a comprehensive account of the behavioral probing pipeline, covering (1) the behavioral profile dimensions, (2) probe data and retriever overlap, (3) the feature representation search, (4) the amplification ratio and soft-token design, and (5) downstream sensitivity analysis.

B.1 Behavioral Profile Dimensions

B.2 Probe Data and Retriever Overlap

We use 500 probe queries from Natural Questions, with the following type distribution: *who* (35%), *when* (24%), *what* (13%), *where* (9%), *how* (6%), and other (13%). Average query length is 9.1 words. Each query is sent to all retrievers with top-10 retrieval.

Table 11 shows the pairwise document-set overlap between the four training retrievers, measured by mean Jaccard similarity across all 500 probes. The overlap structure confirms that retriever behavior varies substantially: BM25 and dense retrievers share <20% of their results, while E5 and BGE (both dense) overlap at 41%.

Group	Dim.	Computation
Reference agreement (4d)	d_1	$\bar{J}(R, R_{\text{BM25}})$
	d_2	$\bar{J}(R, R_{\text{SPLADE}})$
	d_3	$\bar{J}(R, R_{\text{E5}})$
	d_4	$\bar{J}(R, R_{\text{BGE}})$
Document diversity (3d)	d_5	Title vocab diversity
	d_6	Doc length CV
	d_7	Unique title ratio
Query–doc alignment (3d)	d_8	Query term coverage
	d_9	Lexical match score
	d_{10}	Semantic similarity
Uniqueness (1d)	d_{11}	Exclusive doc fraction

Table 10: The 11 dimensions of the behavioral profile. All dimensions use the $N = 4$ training retrievers as fixed reference anchors.

	BM25	SPL	E5	BGE
BM25	1.000	0.176	0.201	0.183
SPLADE	0.176	1.000	0.239	0.216
E5	0.201	0.239	1.000	0.411
BGE	0.183	0.216	0.411	1.000

Table 11: Pairwise mean Jaccard similarity of top-10 document sets across 500 probe queries. E5 and BGE (both dense) show the highest overlap; BM25 (lexical) is most distinct from all others.

B.3 Feature Representation Search

We tested 120+ feature configurations across 7 rounds on 7 A800 GPUs. Table 12 summarizes the major categories. We evaluate features using the Fisher ratio (F: between-retriever / within-retriever variance) and simulated encoder classification accuracy (5-trial average on held-out probes).

Signal vs. noise decomposition. By the law of total variance, any feature $z = f(q, R_i)$ decomposes into between-retriever signal $V_R = \mathbb{E}_q[\text{Var}_R[z|q]]$ and within-retriever noise $V_Q = \text{Var}_q[\mathbb{E}_R[z|q]]$, with Fisher ratio $F = V_R/V_Q$. Content-level features are dominated by query noise ($V_Q \gg V_R$) because all retrievers return topically relevant documents for the same query; set-level features bypass query semantics entirely by comparing document *identity* rather than *content*, yielding $F_{\text{set}} \gg 1$.

Why Transformer encoding fails. All four retrievers receive the *same* 500 probe queries, so the query text dominates the Sentence-Transformer output. Variance decomposition confirms 99.2% of feature variance is within-retriever (driven by query

Feature	Dim	F	Acc
Transformer-based (50+ variants):			
Raw encoding	3584	0.008	21%
+ cross-subtract	3584	0.044	30%
+ top-10% discrim. dims	358	0.097	38%
+ Softmax reweight	3584	0.182	42%
+ LDA projection	3	0.321	46%
BoW / TF-IDF (10+ variants):			
BoW on doc titles	5000	0.001	—
TF-IDF + cross-sub	5000	0.002	—
Set-level features:			
Jaccard matrix	**4**	**4.49**	**100%**
Extended Jaccard	12	2.36	100%
Behavioral profile:			
Full (Jac. + quality)	**11**	**—**	**100%**
Query-type Jaccard	24	—	100%

Table 12: Feature representation search. F = Fisher ratio; Acc = encoder classification accuracy. Jaccard achieves 560× F-ratio improvement over Transformer encoding with only 4 dimensions.

differences) and only 0.8% is between-retriever. Pairwise mean-feature cosine similarities exceed 0.99 for all retriever pairs. Despite testing 50+ post-processing methods (mean removal, PCA, LDA, softmax reweighting, dimension selection), the best configuration achieves only $F = 0.321$ and 46% encoder accuracy—insufficient for reliable conditioning.

Why BoW/TF-IDF fails. Word-level features (BoW, TF-IDF) on returned document titles yield $F < 0.004$—worse than Transformer encoding. Different retrievers return documents about similar topics, so word-level overlap is high. The discriminative signal lies in *which specific documents* are returned, not *what words* they contain.

Why Jaccard works. Jaccard similarity operates at the *document-set level*: it compares whether two retrievers return the same documents, completely bypassing query-text noise. With only 4 dimensions (mean Jaccard against each training retriever), the F-ratio jumps to 4.49 and encoder accuracy reaches 100%.

Behavioral profile adds type structure. While both Jaccard (4d) and the full behavioral profile (11d) achieve 100% classification accuracy, they differ in embedding geometry. Table 13 shows that the 11-dimensional profile correctly reflects paradigm relationships: same-type retrievers (E5↔BGE) have high cosine similarity (+0.51), while cross-type pairs (BM25↔E5) are dissimilar (−0.12). Jaccard alone treats all four retrievers as roughly equidistant.

Metric	Jac. (4d)	Behav. (11d)
BM25↔SPL cos	−0.24	+0.39
E5↔BGE cos	−0.15	+0.51
Sparse↔Dense cos	−0.88	−0.38
Type structure ρ	0.69	**0.83**

Table 13: Embedding geometry. The 11d profile preserves retriever-paradigm relationships; 4d Jaccard treats all retrievers as equidistant.

B.4 Amplification Ratio and Soft-Token Design

Even with 100% encoder accuracy, the soft prompts injected into the LLM can still be identical across retrievers if the projector network fails to preserve differentiation. We discovered that the *amplification ratio*—the ratio of projector output dimensionality to bottleneck dimensionality—is the critical factor.

γ	Config	Acc	P-cos
2×	B=256, 2t×256	100%	−.195
4×	B=256, 4t×256	100%	−.013
8×	B=64, 8t×64	100%	−.094
16×	B=256, 8t×512	87%	+.512
32×	B=256, 8t×1024	47%	+.781
64×	B=64, 1t×4096	25%	+1.00
128×	B=64, 2t×4096	25%	+1.00

Table 14: Amplification ratio (γ) vs. soft-prompt differentiation. P-cos = mean inter-retriever prompt cosine (lower = more differentiated). At $\gamma \geq 64$, all retrievers produce identical prompts.

Table 14 shows a sharp phase transition: at amplification ratio ≤8×, soft prompts are well-differentiated (negative cosine); beyond 16×, differentiation degrades rapidly; at ≥64×, all retrievers produce identical prompts. Our design (d=64, d_{mid}=256, k=2) yields an amplification ratio of 8×, at the boundary of reliable operation.

Two-stage projection resolves the dimension mismatch. The LLM requires d_{model}=4096 per token, but projecting directly from a 64-dimensional bottleneck to 4096 gives amplification >60×. We resolve this with a two-stage design: the projector maps to a compact space ($k \times d_{\text{mid}}$) with controlled

amplification ($\leq 8\times$), then a trainable adapter expands to d_{model}.

Adapter	Acc	P-cos
Direct Linear	98%	−.125
MLP (64-256-4096)	100%	−.178
Residual (frozen+δ)	99%	−.232
LoRA (rank=4)	100%	−.226
Random orth. (frozen)	99%	−.205

Table 15: Adapter architecture comparison (input: 11d, bottleneck 64, Qwen3 hidden size 4096). Acc = retriever-identification accuracy; P-cos = mean inter-retriever prompt cosine. Even a frozen random projection achieves 99% accuracy.

Table 15 shows that even a *frozen random orthogonal projection* achieves 99% accuracy, confirming that the adapter's role is spatial expansion rather than learned transformation. We use a trainable linear adapter in our final design because, at the full 4096-dimensional output, frozen adapters produce positively correlated prompts across retrievers ($\cos = +0.361$), while trainable adapters maintain differentiation ($\cos = -0.214$).

Token count. A single soft token suffices for retriever identification (100% accuracy). Additional tokens improve signal stability (SNR increases from 588 to 1805 with 1→16 tokens) with diminishing returns. We use k=2 as the efficiency-optimal choice.

B.5 Downstream Sensitivity Analysis

We validate that the final profile design is robust to hyperparameter choices by varying three design dimensions on HotpotQA. All experiments use the same trained RAMP checkpoint; only the input profile is recomputed.

Probe count. Varying M from 100 to 1000, performance changes by at most 0.8 EM on training retrievers and 1.2 EM on unseen (Table 16). The marginal gain from 500 to 1000 probes is only 0.1/0.2 EM (Train/Unseen), confirming that $M = 500$ is sufficient. Even $M = 250$ yields only 0.3/0.4 EM degradation.

Probe depth. Performance peaks at top-10 with a smooth inverted-U pattern: top-3 is slightly worse (-0.5/-0.8 EM Train/Unseen) because small document sets increase Jaccard variance, while top-20 is marginally worse (-0.2/-0.4 EM) because lower-ranked documents across retrievers tend to converge, diluting the discriminative signal.

Configuration	Train	Unseen
Probe count (M):		
$M = 100$	41.5	37.8
$M = 250$	41.9	38.6
$M = 500$ (default)	**42.3**	**39.0**
$M = 1000$	42.4	39.2
Probe depth (top-k):		
top-3	41.8	38.2
top-5	42.1	38.7
top-10 (default)	**42.3**	**39.0**
top-20	42.1	38.6
Profile dimensions:		
Full (11d)	**42.3**	**39.0**
Jaccard + Unique (5d)	41.7	38.2
Jaccard only (4d)	41.2	37.4
Non-Jaccard (7d)	39.1	35.6
Diversity + Align (6d)	38.5	34.8

Table 16: Sensitivity of behavioral profile design on HotpotQA (EM). "Train" = average over 4 training retrievers; "Unseen" = average over 4 unseen retrievers.

Profile dimensions. Reference agreement (Jaccard, d_1–d_4) is the core signal: Jaccard-only retains 97.4% of training performance and 95.9% of unseen performance. Removing Jaccard entirely (Non-Jaccard, 7d) causes the largest drop (-3.2 Train, -3.4 Unseen), though it still outperforms unconditioned training (39.1 vs. 37.7 in Table 5). Behavioral uniqueness (d_{11}) is the most efficient supplementary dimension, contributing 0.5–0.8 EM with a single feature. The remaining diversity and alignment dimensions (d_5–d_{10}) provide consistent marginal gains, especially on unseen retrievers (+0.8 EM).

Summary. Across all three dimensions, the maximum variation on training retrievers is $\leq$0.9 EM (excluding configurations that remove the core Jaccard signal). This robustness stems from the high signal-to-noise ratio of Jaccard features: document-set overlap is a stable statistic that converges with moderate probe counts and is insensitive to retrieval depth.

C Supplementary Metrics

Tables 17–18 report Answer F1 and Retrieval Recall@10, supplementing the EM results in the main text.

D Implementation Details

Retriever Services. Each retriever runs as an independent FastAPI service on a dedicated port. BM25 uses Pyserini's (Lin et al., 2021) pre-built

Method	HotpotQA					2WikiMultihopQA					MuSiQue					Bamboogle				
	BM	SPL	E5	BGE	Avg	BM	SPL	E5	BGE	Avg	BM	SPL	E5	BGE	Avg	BM	SPL	E5	BGE	Avg
Naive Agent	43.8	39.5	40.7	40.4	41.1	41.0	36.2	37.0	36.9	37.8	26.0	22.0	24.1	24.3	24.1	48.0	43.2	44.0	46.8	45.5
S-R1-Multi	52.0	48.8	49.4	47.4	49.4	50.6	47.4	46.6	46.2	47.7	28.2	25.8	25.8	25.8	26.4	52.0	50.2	50.4	49.0	50.4
RAMP (name)	55.0	51.8	51.2	50.4	52.1	53.4	50.4	49.0	48.0	50.2	30.0	26.4	26.2	26.2	27.2	54.0	53.4	52.6	52.4	53.1
RAMP (desc.)	55.4	52.2	52.2	51.0	52.7	54.2	51.6	50.8	48.6	51.3	30.4	27.0	27.4	26.8	27.9	55.0	54.0	53.6	52.6	53.8
RAMP (ours)	55.6	52.9	53.6	52.9	53.8	55.0	52.4	51.8	50.8	52.5	31.0	28.0	28.0	27.8	28.7	55.8	55.2	54.8	55.0	55.2
S-R1×4	56.0	53.0	54.4	54.6	54.5	55.2	52.4	53.0	51.6	53.1	29.2	30.4	30.2	29.0	29.7	54.0	55.0	56.2	57.2	55.6

Table 17: Main results (Answer F1), mirroring Table 1. Conclusions are consistent with EM.

Method	Unseen Retrievers (Answer F1)					Training Retrievers (Recall@10)				
	SPL++ $d_B = 0.36$	Cont. $d_B = 0.48$	E5-7B $d_B = 0.62$	GTE $d_B = 0.71$	Avg	BM25	SPL.	E5	BGE	Avg
S-R1(BM25)	44.2	41.8	42.6	42.2	42.7	60.6	44.8	43.4	43.6	48.1
S-R1-Multi	49.4	45.9	46.6	46.1	47.0	55.1	51.4	51.9	51.6	52.5
RAMP (one-hot)	48.2	44.6	44.1	44.4	45.3	—	—	—	—	—
RAMP (name)	50.4	46.1	46.4	46.6	47.4	—	—	—	—	—
RAMP (desc.)	51.4	48.1	48.2	48.4	49.0	—	—	—	—	—
RAMP (type)	51.2	48.4	47.2	46.4	48.3	—	—	—	—	—
RAMP (behav.)	**53.4**	**51.2**	**50.1**	**49.4**	**51.0**	**59.8**	**57.2**	**58.1**	**56.8**	**58.0**
S-R1 (retrained)	54.4	52.4	53.1	52.2	53.0	—	—	—	—	—
S-R1×4	—	—	—	—	—	60.6	58.8	59.2	58.8	59.4

Table 18: Unseen-retriever generalization (Answer F1, left) and training-retriever Recall@10 (right) on HotpotQA.

index; E5 uses the SEARCH-R1 pre-built dense index; SPLADE and BGE indexes are built using their respective encoders over the full wiki-18 corpus with FAISS (Johnson et al., 2021).

Parameter	Value
Base model	Qwen3-8B-Instruct
Hidden size	4096
RL algorithm	GRPO
Learning rate	1×10^{-6}
Batch size	128
Rollout group size	4
Max new tokens	2048
Training steps	400
Retrieval top-k	3
Probe queries (N)	500
Probe top-k	10
Profile dimensions (d_b)	11
Encoder bottleneck (d)	64
Soft tokens (k)	2
Projector intermediate (d_{mid})	256
Retrieval reward weight (λ)	0.3
Block size (K)	8

Table 19: Hyperparameter settings.

Hyperparameters.

Training Details. The base model is Qwen3-8B-Instruct (Qwen Team, 2025). All models are trained with GRPO for 400 steps, batch size 128, learning rate 1×10^{-6}, and 4 rollout responses per prompt, with top-3 retrieval on the wiki-18 corpus. The retrieval reward r_{ret} is applied only to HotpotQA samples (which provide gold supporting fact annotations); NQ samples receive r_{task} only. Block size $K = 8$ for retriever switching. Behavioral probing uses 500 probe queries from Natural Questions with top-10 retrieval (probe depth is independent of training/eval retrieval depth). All experiments use 4×A100 80GB GPUs.

Reproduction Notes. We follow the same retrieval setup as the original SEARCH-R1 (Jin et al., 2025): GRPO, top-3 retrieval, and the wiki-18 corpus, while updating the base model to Qwen3-8B-Instruct. Public Search-R1-style Qwen3-8B results are around 37 EM on HotpotQA with E5 under comparable retrieval settings; our S-R1(E5) achieves 42.8 EM after retriever-specific training. Absolute calibration may vary with rollout scheduling and random seed, but all baselines and RAMP share identical training and evaluation settings, so relative comparisons remain fair. To enable external verification, we will release all training configurations, random seeds, evaluation scripts, model checkpoints, and probe caches upon acceptance.