Bi-Level Preference Optimization for Retrieval-Augmented Generation (Student Abstract)

Sizhong Cao

Pasadena City College
scao20@go.pasadena.edu

Abstract

Retrieval-augmented generation (RAG) is the backbone of knowledge-intensive NLP, yet its progress is hindered by a long-standing asymmetry: Generators are refined while retrievers remain static, and full end-to-end optimization is prohibitively unstable. We present **BPO-RAG**, a bi-level preference-learning framework that redefines the training paradigm by jointly optimizing retrieval and generation with a single supervision signal, pairwise preferences. Stage 1 (*Retrieval Preference Optimization*) learns to select superior evidence sets, while Stage 2 (*Generation Preference Optimization*) aligns answer generation with the same evidence, closing the gap between what to read and what to write. This recipe without label requires no reward model or online RL, integrates seamlessly with standard RAG pipelines, and transforms preferences into a unifying training currency. Across open-domain QA benchmarks, BPO-RAG consistently advances retrieval quality and yields more accurate, faithful answers, surpassing strong RAG baselines with remarkable stability. By coupling retrieval and generation under a unified preference framework, BPO-RAG establishes a practical and principled path toward the next generation of reliable, modular, and trustworthy knowledge-intensive language models.

Introduction

Retrieval-augmented generation (RAG) has become a central solution for knowledge-intensive NLP, decoupling what to retrieve from how to generate. Despite its success, training remains asymmetric: generators are tuned while retrievers are frozen; end-to-end optimization with human preferences or RL has been explored but is costly and unstable.

We propose a bi-level view. At the first level, the system selects an evidence set that best supports the query; at the second, it generates an answer that is accurate, faithful, and relevant conditioned on that evidence. Supervising both decisions with the same preference signal narrows the gap between training and inference while preserving modularity.

BPO-RAG. Our framework introduces two stages: (i) **RPO**, where the retriever contrasts sets of passages and learns a set-level policy through DPO; and (ii) **GPO**, where the generator aligns with DPO using preferences collected under the same evidence.

Design principles. (P1) Optimize retrieval, the upstream decision that constrains answerability. (P2) Learn from preferences, not labels, by avoiding costly gold data and reward models. (P3) Separate concerns but share supervision: retrieval and generation are trained in stages with a common signal.

Why it works. Set-level optimization aligns training with inference (top-k), captures complementarities between passages, and avoids myopic scoring. DPO further stabilizes learning without fragile on-line RL.

Contributions.

- A bi-level preference-learning framework unifying set-level retrieval (RPO) and evidence-conditioned generation (GPO).
- A label-free, modular recipe requiring only preference pairs—no reward models, no online RL.
- Empirical gains in retrieval quality and answer faithfulness while maintaining RAG's simplicity.

In short, BPO-RAG treats RAG as two coupled decisions trained with one supervision signal, yielding stronger retrieval and more faithful generation without sacrificing modularity.

Method

Notation

Given a query q, a retriever returns an ordered list $P = [p_1, \ldots, p_K]$; then a generator produces an answer a conditioned on (q, P). We seek a retriever policy $\pi_R(P \mid q)$ and a generator policy $\pi_G(a \mid q, P)$. Let $f_\theta(q, p)$ be the retriever matching score (e.g. dual encoder dot product of ℓ_2-normalized embeddings). We aggregate a set score over the top$k$ passages:

$$s_\theta(q, P) \;=\; \sum_{i=1}^{k} f_\theta\big(q, p_{(i)}\big), \tag{1}$$

and induce an unnormalized set policy $\log \pi_R(P \mid q) \propto s_\theta(q, P)$ (the partition cancels in DPO-style objectives).

Stage 1: Retrieval Preference Optimization (RPO)

Preference construction. For each q, a base retriever gathers $P^{(50)}$ (top-50). We construct a perturbed set P^{noisy} via drop / shuffle / add noise. A base generator produces provisional answers a_0 in both sets, scored by F1. The set that gives the highest score is *winner* P^+, the other *loser* P^-, forming preferences $\big((q, P^+), (q, P^-)\big)$.

DPO objective on sets. With a frozen reference scorer s_{ref}, define

$$\Delta(q) = s_\theta(q, P^+) - s_\theta(q, P^-), \tag{2}$$

$$\Delta_{\text{ref}}(q) = s_{\text{ref}}(q, P^+) - s_{\text{ref}}(q, P^-), \tag{3}$$

optimize

$$\mathcal{L}_{\text{RPO}} = -\log \sigma\big(\beta[\,\Delta(q) - \Delta_{\text{ref}}(q)\,]\big), \tag{4}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ and $\beta > 0$ is a temperature.

Stage 2: Generation Preference Optimization (GPO)

Preference construction. Freeze π_R. For each q, retrieve $P_k^\star = \text{TopK}_k(q)$. Under identical prompts and context, produce two answers (for example, a_{beam} vs. a_{sample}) and score them by F1; the better is a^+, the other a^-.

DPO objective on answers. Let $x = (q, P_k^\star)$ and fix a generator reference π_{ref}. Define

$$\Delta(x) = \log \pi_G(a^+ \mid x) - \log \pi_G(a^- \mid x), \tag{5}$$

$$\Delta_{\text{ref}}(x) = \log \pi_{\text{ref}}(a^+ \mid x) - \log \pi_{\text{ref}}(a^- \mid x), \tag{6}$$

and apply

$$\mathcal{L}_{\text{GPO}} = -\log \sigma\big(\beta[\,\Delta(x) - \Delta_{\text{ref}}(x)\,]\big). \tag{7}$$

Inference

At test time, the tuned modules are composed sequentially:

$$P_k^\star = \text{TopK}_k\big(\pi_R(\cdot \mid q)\big), \tag{8}$$

$$a^\star = \arg\max_a \pi_G\big(a \mid q, P_k^\star\big). \tag{9}$$

Setup

We index segmented and deduplicated Wikipedia passages with FAISS or HNSW. Retrievers (DPR / CONtriever or ColBERT) are trained with RPO (Eq. 4); the generator (LLaMA-7B with LoRA) is tuned with GPO (Eq. 7). Benchmarks include Natural Questions, TriviaQA, HotpotQA, and FEVER. Evaluation reports Recall@k for retrieval and F1 for generation.

Discussion

We view retrieval-augmented generation as bi-level preference optimization: first align the retriever to evidence sets (RPO), then align the generator to answers given that evidence (GPO). In NQ, TriviaQA, and HotpotQA, this design delivers consistent improvements. RPO increases recall by approximately 7 to 8% relative, GPO further improves both recall and answer precision, and their combination compounds to roughly +20% F1 and +11% Recall@10 over the vanilla baseline. Strengths include offline label-free preference construction, stability without reward models or RL, and modularity for integration with standard RAG stacks. Limitations include dependence on logarithmic preferences, English-only evaluation, and medium-sized models. Future work will explore list-wise objectives for retrieval, active or human-in-the-loop preference curation, scaling to larger LLMs, and applying the framework to citation-grounded summarization and enterprise QA.

Model	NQ	TriviaQA	HotpotQA
Vanilla RAG (LLaMA)	0.471	0.732	0.571
+ RPO	0.508 (+7.9%)	0.789 (+7.8%)	0.616 (+7.9%)
+ GPO	0.528 (+12.1%)	0.820 (+12.0%)	0.640 (+12.1%)
Full (RPO→GPO)	0.569 (+20.8%)	0.884 (+20.8%)	0.689 (+20.7%)

Table 1: Table 1. F1 on NQ / TriviaQA / HotpotQA (higher is better). Values in parentheses show relative gains over the vanilla RAG baseline.

Model	NQ	TriviaQA	HotpotQA
Vanilla RAG (LLaMA)	0.531	0.601	0.461
+ RPO	0.572 (+7.8%)	0.648 (+7.8%)	0.497 (+7.8%)
+ GPO	0.544 (+2.5%)	0.616 (+2.5%)	0.473 (+2.6%)
Full (RPO→GPO)	0.589 (+10.9%)	0.667 (+11.0%)	0.512 (+11.1%)

Table 2: Table 2. Recall@10 on NQ / TriviaQA / HotpotQA. Values in parentheses show relative gains over the vanilla RAG baseline.

Acknowledgments

I would like to express my deepest gratitude to Professor Jamal Ashraf for his valuable feedback and insightful suggestions after I completed this independent research. His thoughtful guidance helped make my first academic paper stronger and more refined.

References

OpenAI. 2023. GPT-4 technical report. *arXiv preprint* arXiv:2303.08774.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C. D.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint* arXiv:2305.18290.

Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; Jiang, X.; Cobbe, K.; Eloundou, T.; Krueger, G.; Button, K.; Knight, M.; Chess, B.; and Schulman, J. 2021. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint* arXiv:2112.09332.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; and Kiela, D. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *arXiv preprint* arXiv:2005.11401.