Right Routing, Right Answering: Joint Path-Answer Preference Optimization for Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) often suffers from noisy or irrelevant retrievals, which can substantially undermine the quality of generated answers. While recent methods like Retrieval Preference Optimization (RPO) empower LLMs to adaptively decide whether to use retrieved content, they primarily focus on improving this routing decision. A critical oversight is the lack of supervision over answer quality within a selected path. Consequently, even with correct routing, the correctness of the final answer is not adequately guaranteed. To address this, we propose Joint Path-Answer Preference Optimization (JPAPO), a novel framework that jointly optimizes both *path routing* and *within-path answering*. Our solution is simple yet effective, tackling this dual challenge through three strategically designed preference pairs, ensuring both easy integration and scalability. Extensive experiments across diverse benchmarks and LLM backbones demonstrate the framework's effectiveness, achieving improvements of up to 5.9% over RPO.

1 Introduction

Large language models (LLMs) have achieved strong performance on various commonsense tasks (Brown et al., 2020; Team et al., 2023; Touvron et al., 2023). However, their knowledge is constrained by static parameters, making it difficult to update rapidly (Dhuliawala et al., 2024; Huang et al., 2025; Ji et al., 2023; Sun et al., 2024; Xu et al., 2024b; Zhang et al., 2023). This limitation often leads to outdated information or hallucinations when tackling knowledge-intensive problems. To address this, Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Izacard and Grave, 2021) enhances LLMs by dynamically retrieving relevant information from external sources to guide the generation process. This integration of retrieved evidence significantly improves the factual accuracy and reliability of the model's outputs.

While promising, RAG remains critically dependent on retrieval quality. Noisy or irrelevant documents can introduce errors that propagate directly into the generated output, degrading its reliability (Su et al., 2024; Li et al., 2023; Yoran et al., 2023; Izacard and Grave, 2021). A further complication arises when retrieved information conflicts with the model's parametric knowledge, potentially causing the model to over-rely on external sources and produce hallucinations (Longpre et al., 2021; Xu et al., 2024a). Early attempts to mitigate these issues often involved routing mechanisms, i.e., deciding whether to invoke retrieval, or perform post-generation quality checks (Yan et al., 2024; Asai et al., 2024; Wang et al., 2025; Xiang et al., 2024). However, these approaches typically require auxiliary modules or multiple LLM calls, incurring substantial computational overhead during both training and inference.

Recent research has increasingly focused on endowing models with the intrinsic ability to adaptively incorporate provided content into its reasoning. The recent state-of-the-art approach, Retrieval Preference Optimization (RPO) (Yan et al., 2025), endow LLMs with this adaptive capability through reinforcement learning (Yan et al., 2025; Rafailov et al., 2023). Specifically, RPO trains the model to internally decide between two reasoning paths when retrieved content is provided: answering directly (using only its internal knowledge) or answering in a retrieval-augmented manner (incorporating retrieved information). This is achieved by constructing two types of preference pairs: (1) where a correct direct answer is preferred over an incorrect retrieval-augmented answer, and (2) where a correct retrieval-augmented answer is preferred over an incorrect direct answer.

While RPO effectively improves routing between direct and retrieval-augmented answering, it

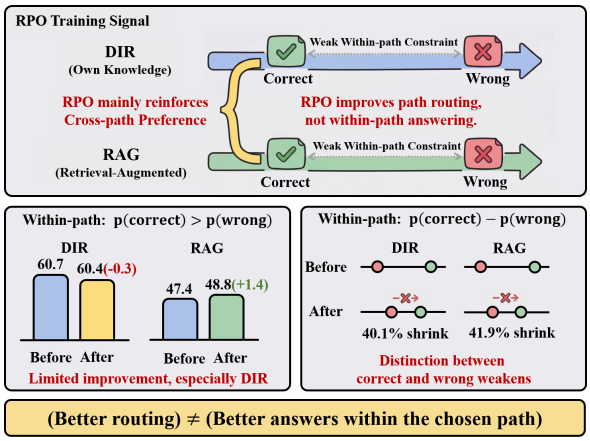

Figure 1: Analysis on the ASQA dataset, revealing that RPO improves path routing but does not adequately enhance within-path answering.

provides no direct supervision over answer quality within a chosen path. Consequently, the correctness of answers generated along that path remains inadequately assured. To investigate how RPO influences answer correctness along a chosen reasoning path, we analyze the decoding probabilities—both before and after RPO training—of a given direct answer or retrieval-augmented answer from the LLM (i.e., Llama 3 (Grattafiori et al., 2024)).

As illustrated in Figure 1, our analysis on the ASQA dataset (Stelmakh et al., 2022) reveals that RPO training has a limited effect on reinforcing the model's preference for correct answers within a chosen path. For direct answering, the proportion of positive–negative pairs in which the positive answer received a higher probability decreased by 0.3%. Similarly, for retrieval-augmented answering, this proportion increased by only 1.4%. More notably, the gap between positive and negative answer probabilities narrowed by 40.1% for direct answering and 41.9% for retrieval-augmented answering.

These observations indicate that RPO's performance gains are driven more by improved routing decisions than by enhanced answer quality on a chosen path. An ideal model must therefore master two aspects: optimally selecting a reasoning path and generating reliable answers within it. Motivated by this insight, we propose Joint Path-Answer Preference Optimization (JPAPO), a novel framework that jointly optimizes both path routing and within-path answering. Specifically, we train the model using three distinct types of preference pairs: cross-path, direct in-path and retrieval in-path, resulting in a highly scalable solution that is straightforward to integrate.

Our contributions can be summarized as follows:

- We identify a fundamental weakness in existing methods like RPO: the lack of supervision over answer quality within a chosen path. Consequently, the quality of the final answer is not adequately ensured, creating a performance bottleneck.
- We propose a novel framework, JPAPO, to uniformly address the dual challenge of path routing and within-path answering for a LLM, offering a simple yet effective solution that significantly boosts LLM performance.
- Extensive experiments show that our approach achieves significant improvements over recent baselines. Comprehensive analyses, including ablation studies, probability rankings and gaps, and error analysis, further validate the superiority of our method.

2 Related Work

2.1 Retrieval-Augmented Generation

In RAG, retrieved results may contain noise or erroneous information and may conflict with the model's parametric knowledge, thus undermining the reliability of the generated content (Longpre et al., 2021; Xu et al., 2024a). To address this issue, existing research has primarily focused on improving the use of retrieved information in three lines: (1) assessing and filtering the quality of retrieved results **before generation** (Lewis et al., 2020; Yan et al., 2024; Wang et al., 2025); (2) generating multiple candidate answers based on different evidence **after generation**, and obtaining the final output through re-ranking or selection (Asai et al., 2024; Xiang et al., 2024); and (3) introducing dynamic decision-making mechanisms **during generation**, enabling the model to adaptively select and integrate retrieved information during reasoning and decoding (Yan et al., 2025). While these methods improve performances, the first two lines often require extra models or multiple generations, increasing cost and complexity. Our work therefore focuses on the third line, aiming to improve use retrieved information during generation.

2.2 Preference Optimization

As an effective technique for aligning model outputs with desired behaviors, preference optimiza-

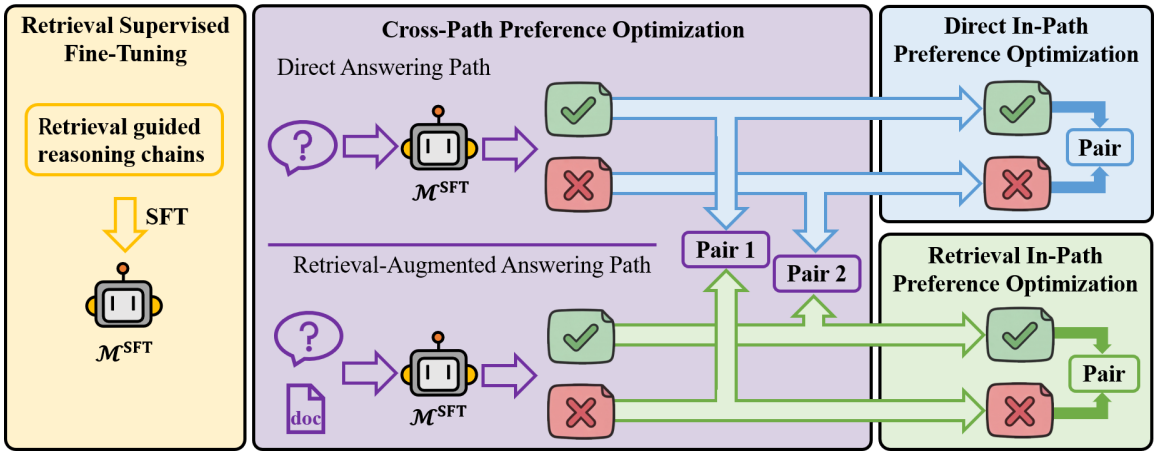

Figure 2: The overview of our Joint Path-Answer Preference Optimization (JPAPO) framework.

tion has progressed significantly, evolving from initial RLHF (Ouyang et al., 2022) to more straightforward, direct loss objectives such as DPO (Rafailov et al., 2023). RLHF typically uses supervised fine-tuning, then trains a reward model and optimizes the policy with PPO (with variants like GRPO (Shao et al., 2024) improving stability and efficiency). To reduce complexity and cost, DPO removes explicit reward models and online sampling by directly optimizing a preference objective, with many extensions for robustness and controllability. Recently, RPO (Yan et al., 2025) extends preference optimization to RAG. Specifically, it improves robustness of RAG by adding regularization to the preference optimization to mitigate noise and training instability. Our method builds upon the RPO framework of using preference pairs and RL for RAG optimization. The key improvement lies in enhancing within-path answering capability through the construction of two distinct pair types: direct in-path and retrieval in-path.

3 Task Definition

Given a question q, a retriever first fetches a set of K relevant documents $D^r = \{d_1^r, \ldots, d_K^r\}$ from a source corpus D. A large language model $\mathcal{M}$ then produces an answer by conditioning on the tuple (q, D^r). This can be formalized by the following equation:

$$\hat{y}_{\text{rag}} = \mathcal{M}(q, D^r),$$

where $\hat{y}_{\text{rag}}$ is the generated answer.

4 Method

Our approach consists of two key steps: retrieval supervised fine-tuning (SFT) and preference optimization. First, we perform SFT on a dataset of retrieval-guided reasoning chains, which teaches the model to ground its responses in provided evidence, obtaining a strong base model. Second, we perform preference optimization to jointly improve the model's path routing and within-path answering capabilities. This step leverages three distinct types of preference pairs: cross-path, direct in-path and retrieval in-path. An overview of our method is illustrated in Figure 2.

4.1 Retrieval Supervised Fine-Tuning

Considering that supervised fine-tuning (SFT) provides a crucial foundation for subsequent preference optimization, our method begins by fine-tuning the base model $\mathcal{M}$ using a dataset of reasoning chains, $\mathcal{T} = \{\langle q_i, a_i, D_i^r, c_i \rangle\}_{i=1}^n$, in line with InstructRAG (Wei et al., 2024). Here, for each sample, a_i is the ground-truth answer to question q, and c_i is a reasoning chain. Each chain c_i is produced online by prompting $\mathcal{M}$ to generate a step-by-step rationale using q_i, a_i and retrieved context D_i^r. Training minimizes the negative log-likelihood:

$$\mathcal{M}^{\text{SFT}} = \arg\min_{\mathcal{M}} \mathbb{E}\big[-\log P_{\mathcal{M}}(c \mid q, D^r) \big],$$

where $\mathcal{M}^{\text{SFT}}$ is fine-tuned model.

The resulting model, $\mathcal{M}^{\text{SFT}}$, gains proficiency in synthesizing retrieved evidence with the query to produce conditioned answers. This provides a solid retrieval-augmented basis for sequential preference optimization.

4.2 Preference Optimization

Our preference optimization employs three distinct pair types: cross-path, direct in-path and retrieval in-path. This approach is designed to sufficiently optimize both the model's path routing and within-path answering capabilities.

Cross-Path Preference Optimization This component optimizes the model to choose between direct answering (using only parametric knowledge) and retrieval-augmented answering (using D^r). Following the RPO framework (Yan et al., 2025), we construct preference pairs for training. Specifically, for a given query given query q and retrieved context D^r, we prompt the SFT model $\mathcal{M}^{\text{SFT}}$ to sample both a direct answering $\hat{y}_{\text{dir}} = \mathcal{M}^{\text{SFT}}(q)$ and a retrieval-augmented answering $\hat{y}_{\text{rag}} = \mathcal{M}^{\text{SFT}}(q, D^r)$. The preferred response y^+ is assigned to the correct pathway's output, while the dispreferred response y^- is assigned to the incorrect one. This yields a preference pair (y^+, y^-) for standard DPO training, formalized by the objective:

$$\max_{\theta} \mathbb{E}\Bigg[\log\sigma\Bigg(\beta\log\frac{\mathcal{M}_\theta^{\text{SFT}}(y^+ \mid q, D^r)}{\mathcal{M}_{\text{ref}}^{\text{SFT}}(y^+ \mid q, D^r)} - \beta\log\frac{\mathcal{M}_\theta^{\text{SFT}}(y^- \mid q, D^r)}{\mathcal{M}_{\text{ref}}^{\text{SFT}}(y^- \mid q, D^r)}\Bigg) \pm \frac{\beta}{|D^r|}\log\frac{\mathcal{M}_\theta^{\text{SFT}}(D^r \mid q)}{\mathcal{M}_{\text{ref}}^{\text{SFT}}(D^r \mid q)}\Bigg],$$

where $\sigma(\cdot)$ is the sigmoid function $\sigma(z) = 1/(1 + e^{-z})$, and β is a temperature hyperparameter controlling the strength of the KL regularization. The last term is the retrieval reward: apply a "+" when the retrieval-augmented answering $\hat{y}_{\text{rag}}$ is preferred (encouraging reliance on retrieved evidence), and a "−" when the direct answering $\hat{y}_{\text{dir}}$ is preferred (penalizing unnecessary or noisy retrieval).

Direct In-Path Preference Optimization Once cross-path preference optimization is complete, we proceed to in-path optimization to improve generation quality under the chosen path. Specifically, this component refines the model's answer quality when the direct answering path is chosen. For each question q, we sample n direct answers $\{\hat{y}_{\text{dir}}^{(j)}\}_{j=1}^n$. Using the ground-truth answer a, we score these answers, selecting a preferred answer y^+ from the correct set and a dispreferred y^- from the incorrect set to form preference pairs. The training objective is formalized as:

$$\max_{\theta} \mathbb{E}\Bigg[\log\sigma\Bigg(\beta\log\frac{\mathcal{M}_\theta^{\text{CPPO}}(y^+ \mid q)}{\mathcal{M}_{\text{ref}}^{\text{CPPO}}(y^+ \mid q)} - \beta\log\frac{\mathcal{M}_\theta^{\text{CPPO}}(y^- \mid q)}{\mathcal{M}_{\text{ref}}^{\text{CPPO}}(y^- \mid q)}\Bigg)\Bigg],$$

where $\mathcal{M}^{\text{CPPO}}$ denotes the model trained using Cross-Path Preference Optimization.

Retrieval In-Path Preference Optimization Similarly, this component refines the model's answer quality when the retrieval-augmented answering path is chosen. For each question q and its retrieved context D^r, we sample n retrieval-augmented answers $\{\hat{y}_{\text{rag}}^{(j)}\}_{j=1}^n$ to construct preference pairs (y^+, y^-). The corresponding training objective is formalized as:

$$\max_{\theta} \mathbb{E}\Bigg[\log\sigma\Bigg(\beta\log\frac{\mathcal{M}_\theta^{\text{DIPPO}}(y^+ \mid q, D^r)}{\mathcal{M}_{\text{ref}}^{\text{DIPPO}}(y^+ \mid q, D^r)} - \beta\log\frac{\mathcal{M}_\theta^{\text{DIPPO}}(y^- \mid q, D^r)}{\mathcal{M}_{\text{ref}}^{\text{DIPPO}}(y^- \mid q, D^r)}\Bigg)\Bigg],$$

where $\mathcal{M}^{\text{DIPPO}}$ denotes the model trained using Retrieval In-Path Preference Optimization.

Through these three preference optimization strategies, the model thus becomes proficient not only at selecting the correct reasoning path but also at generating high-quality answers within a chosen path.

4.3 Inference

At inference time, we always provide the model with the query and retrieved evidence, i.e., (q, D^r). The model then implicitly routes between two behaviors: (1) direct reasoning that relies on its parametric knowledge, and (2) retrieval-augmented reasoning that incorporates D^r. Concretely, we decode a single response from $\mathcal{M}^{\text{JPAPO}}$ conditioned on (q, D^r); no additional external router is used. Formally, inference is:

$$\hat{y}_{\text{rag}} = \mathcal{M}^{\text{JPAPO}}(q, D^r),$$

where $\mathcal{M}^{\text{JPAPO}}$ denotes the model trained using our JPAPO framework.

5 Experiment

5.1 Setup

Datasets We evaluate our method on four benchmarks: PopQA (Mallen et al., 2023), TriviaQA (Joshi et al., 2017), Natural Questions (NQ) (Kwiatkowski et al., 2019) and ASQA (Stelmakh et al., 2022). Detailed dataset statistics are provided in Table 1. Evaluation metrics follow their conventional definitions: for PopQA, TriviaQA, and NQ, we use accuracy, checking if the generated text contains the correct answer. For ASQA, we adopt the official string-based exact match (str-em), which measures how well the output covers all valid answers to a question's constituent sub-questions.

Dateset	Train	Test	Retriever
ASQA	4353	948	GTR
PopQA	12868	1399	Contriever
NQ	79168	3610	DPR
TriviaQA	78785	11313	Contriever

Table 1: Dataset statistics and retriever configurations for the four benchmarks.

Baselines We compare our method against the following recent and representative baselines:

- **InstructRAG** (Wei et al., 2024) trains models with self-synthesized reasoning instructions via supervised fine-tuning.
- **RetRobust** (Lewis et al., 2020) improves robustness to noisy retrieval by fine-tuning on a mixture of retrieval-relevant and retrieval-irrelevant data.
- **Self-RAG** (Asai et al., 2024) is trained to output special reflection tokens to control adaptive retrieval and generation.
- **ASTUTERAG** (Wang et al., 2025) performs comparison between a LLM's internal knowledge and external retrievals to select the most credible information for final answer.
- **RPO** (Yan et al., 2025) constructs preference pairs for reinforcement learning by contrasting correct direct answers with incorrect retrieval-based answers (and vice-versa).

Implementation Details To ensure a fair comparison with all baselines, we adopt the same train/test splits and retrieval configurations as InstructRAG. We employ Contriever, DPR, and GTR as retrievers, setting the context length to $K = 5$ paragraphs per query. For answer sampling, we use a temperature of 0.7 and draw $n = 5$ samples per question. In preference optimization, we set $\beta = 0.1$ and initialize the reference model from the SFT checkpoint. Since RPO is not publicly available, we re-implement it based on the fine-tuned model released by InstructRAG. Our proposed method is similarly built upon the InstructRAG framework to maintain consistency across comparisons.

5.2 Main Results

Method	ASQA (em)	PQA (acc)	NQ (acc)	TQA (acc)
InstructRAG	47.6	66.8	65.7	78.7
RetRobust	40.5	56.5	54.2	71.5
Self-RAG	36.9	55.8	42.8	71.4
ASTUTERAG	-	44.4	52.2	84.1
RPO*	48.8	67.9	68.9	81.8
Ours	**50.0**	**70.1**	**74.8**	**84.5**

Table 2: Performance comparison on the four datasets. RPO* represents our reproduction within the InstructRAG framework. Results of other baselines are retrieved from (Wei et al., 2024).

As shown in Table 2, our method achieves superior performance over all existing baselines across the four benchmarks. Although RPO* represents a significant advance as the current SOTA, our method consistently surpasses it, with a notable 5.9% accuracy improvement on NQ. The key distinction lies in the training objective: while both approaches optimize path routing, our method additionally employs preference pairs to supervise answer quality within a selected path. The consistent performance gap highlights that improved routing does not adequately ensure answer correctness, revealing these two capabilities as potentially orthogonal. These findings confirm the effectiveness of our framework in addressing this dual challenge.

5.3 Ablation Study

We conduct an ablation study to evaluate the contribution of each component in our preference optimization framework, which comprises three distinct types of preference pairs: (1) **CP**: Cross-Path

Method	ASQA (em)	PQA (acc)	NQ (acc)	TQA (acc)
SFT	47.6	66.8	65.7	78.7
+DIP	48.1	66.8	69.8	82.1
+DIP&RIP	49.0	68.2	**74.9**	**84.7**
+CP	48.8	67.9	68.8	81.8
+CP&DIP	49.2	69.1	72.2	83.3
+CP&DIP&RIP	**50.0**	**70.1**	74.8	84.5

Table 3: Ablation study of preference optimization strategies across the four datasets.

Preference Optimization; (2) **DIP**: Direct In-Path Preference Optimization; (3) **RIP**: Retrieval In-Path Preference Optimization.

Ablated variants are obtained by combining different pairs to train the retrieval supervised fine-tuning (SFT) model. The results are presented in Table 3. First, progressively adding DIP and RIP to the SFT model yields consistent performance gains across all datasets. Although SFT is not explicitly trained on CP, it retains some adaptive capability; it still can benefit from the explicit quality supervision provided by DIP and RIP. Second, "+CP" variant outperforms SFT, confirming the value of optimizing the model to choose between answering paths. Finally, performance improves consistently as DIP and RIP are added to CP, suggesting that the contributions of routing (CP) and within-path quality (DIP/RIP) are complementary and largely orthogonal. These results collectively verify the effectiveness of our complete method.

5.4 Further Analysis

5.4.1 Performance on Different LLMs

Setting	Llama	Qwen	DeepSeek
SFT	66.8	64.2	67.1
RPO*	67.9	68.3	68.3
Ours	**70.1**	**68.7**	**69.0**

Table 4: Performance of three LLM backbones on the PopQA dataset.

Table 4 presents the performance of our method on the PopQA dataset across three different LLM backbones: Llama[1], Qwen[2] and DeepSeek[3]. Our method achieves the best performance with each backbone. Compared to RPO*, our method yields gains of +0.4 on Qwen and +0.7 on DeepSeek, with a larger improvement on Llama. These results demonstrate the robustness of our proposed framework across diverse model architectures and initializations.

[1]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[2]https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

[3]https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-8B

5.4.2 Performance on Cross-Domain Tasks

Setting		ASQA (em)	PQA (acc)	NQ (acc)	TQA (acc)
ASQA	RPO*	48.8	66.5	65.6	76.4
	Ours	**50.0**	**67.0**	**66.5**	**77.3**
PQA	RPO*	46.7	67.9	64.4	77.2
	Ours	**50.7**	**70.1**	**68.5**	**80.2**
NQ	RPO*	48.4	66.3	68.8	77.8
	Ours	**55.7**	**69.5**	**74.8**	**82.2**
TQA	RPO*	49.6	**65.9**	67.5	81.8
	Ours	**50.8**	65.5	**69.8**	**84.5**

Table 5: Performance comparison between RPO* and our method across 16 cross-domain pairs. Rows indicate the training domain; columns indicate the testing domain.

To evaluate the generalization capability of our method, we conduct cross-domain experiments comparing it with RPO*. In this setup, models are trained on one dataset (source) and evaluated on another (target), creating 16 distinct source-target pairs from our four benchmarks.

The results are presented in Table 5. Our method achieves superior performance over RPO* in nearly all cross-domain pairs. This advantage indicates that our improvements stem from enhanced model capability rather than overfitting to the training data. By jointly optimizing path routing and within-path answering, our framework demonstrates stronger robustness and superior generalization across diverse knowledge domains.

5.4.3 Analysis on Within-Path Answering

Probability Rankings We analyze probability rankings to assess the model's within-path answering capability. Specifically, for both direct and retrieval-augmented answering, we separate sampled answers into positive (correct) and negative (incorrect) sets. Given the retrieved context, the model is prompted to decode each answer, and we compute its decoding probability. For a preference pair (y^+, y^-) of direct (or retrieval-augmented)

Setting	Method		ASQA		PopQA		NQ		TriviaQA	
			DIR	RAG	DIR	RAG	DIR	RAG	DIR	RAG
Train	SFT	$p(y_l) > p(y_r)$	904	466	1939	929	13126	7091	10214	4420
		$p(y_l) \leq p(y_r)$	582	516	1211	840	9588	5966	7300	4113
	RPO*	$p(y_l) > p(y_r)$	899	480	1973	1035	13801	8032	11322	5706
		$p(y_l) \leq p(y_r)$	587	502	1177	734	8913	5025	6192	2827
	Ours	$p(y_l) > p(y_r)$	908	512	2127	1261	16573	9751	14002	6849
		$p(y_l) \leq p(y_r)$	578	470	1023	508	6141	3306	3511	1684
Test	SFT	$p(y_l) > p(y_r)$	195	78	145	89	604	326	1491	623
		$p(y_l) \leq p(y_r)$	132	109	116	57	443	312	1040	680
	RPO*	$p(y_l) > p(y_r)$	187	82	151	92	630	361	1612	814
		$p(y_l) \leq p(y_r)$	140	105	110	54	417	277	919	489
	Ours	$p(y_l) > p(y_r)$	189	83	154	104	702	435	1940	953
		$p(y_l) \leq p(y_r)$	138	104	107	42	345	202	591	350

Table 6: Probability ranking results for positive–negative answer pairs on the four datasets. DIR: direct answering; RAG: retrieval-augmented answering.

answering, a ranking of $p(y_l) > p(y_r)$ indicates the model correctly assigns higher likelihood to the positive answer. A higher proportion of such pairs reflects stronger within-path answering performance. Table 6 presents these results on the training and testing sets.

RPO* yields inconsistent improvements over the SFT baseline and can even degrade probability rankings. For example, on the ASQA test set for direct answering, the number of correctly ranked pairs $p(y_l) > p(y_r)$ decreases from 195 to 187; a similar drop occurs on the training set. This aligns with the Main Results, where RPO* provides only limited gains on ASQA and PopQA, reflecting its weak discrimination between correct and incorrect answers within an answering path.

In contrast, our method demonstrates consistent and substantial improvements across all datasets for both training and testing splits. It not only mitigates or reverses the regressions observed with RPO* (e.g., on ASQA and PopQA) but also amplifies gains where RPO* shows some benefit (e.g., on NQ and TriviaQA). For instance, on the TriviaQA test set for retrieval-augmented answering, our method improves the count of correctly ranked pairs by 53.0% over SFT, compared to only 9.7% for RPO*. Similarly, on the NQ training set for direct answering, our method achieves a 26.2% increase versus 5.1% for RPO*. These robust trends across both data splits and answering modes underscore the strong generalization capability of our framework.

Probability Gap Changes we further investigate the effect of our method and RPO* on within-path answering by analyzing changes in the probability margin between positive and negative answers relative to the SFT baseline. Here, $\Delta \uparrow$ counts pairs where this margin increases, and $\Delta \downarrow$ counts pairs where it decreases. The results are shown in Figure 7.

Across both training and testing splits, and for both direct (DIR) and retrieval-augmented (RAG) answering, our method consistently yields more $\Delta \uparrow$ and fewer $\Delta \downarrow$ compared to RPO*. This indicates a stronger tendency to widen the discriminative gap between correct and incorrect answers. Crucially, this margin-amplification pattern generalizes robustly from the training set to the test set.

5.4.4 Error Analysis

We conduct an error analysis to examine how our method and RPO* affect the model's predictions. Figure 3 presents the category transition matrices for both approaches across the datasets.

Overall, our method demonstrates a stronger tendency than RPO* to transition samples from non-ideal states toward the ideal C/C state, while maintaining the stability of samples already in C/C. Specifically: (1) for I/I samples (both answers in-

Setting	Method		ASQA		PopQA		NQ		TriviaQA	
			DIR	RAG	DIR	RAG	DIR	RAG	DIR	RAG
train	RPO*	$\Delta\uparrow$	890	570	2011	1132	14947	8572	12101	6012
		$\Delta\downarrow$	596	412	1139	637	7767	4485	5413	2521
	Ours	$\Delta\uparrow$	917	668	2235	1367	17067	9790	14174	6895
		$\Delta\downarrow$	569	314	915	402	5647	3267	3340	1638
test	RPO*	$\Delta\uparrow$	184	104	154	86	681	399	1724	875
		$\Delta\downarrow$	143	83	107	60	366	239	807	428
	Ours	$\Delta\uparrow$	190	107	169	101	713	440	1958	962
		$\Delta\downarrow$	137	80	92	45	334	198	573	341

Table 7: Results of probability gap changes for positive–negative answer pairs on the four datasets. DIR: direct answering; RAG: retrieval-augmented answering.

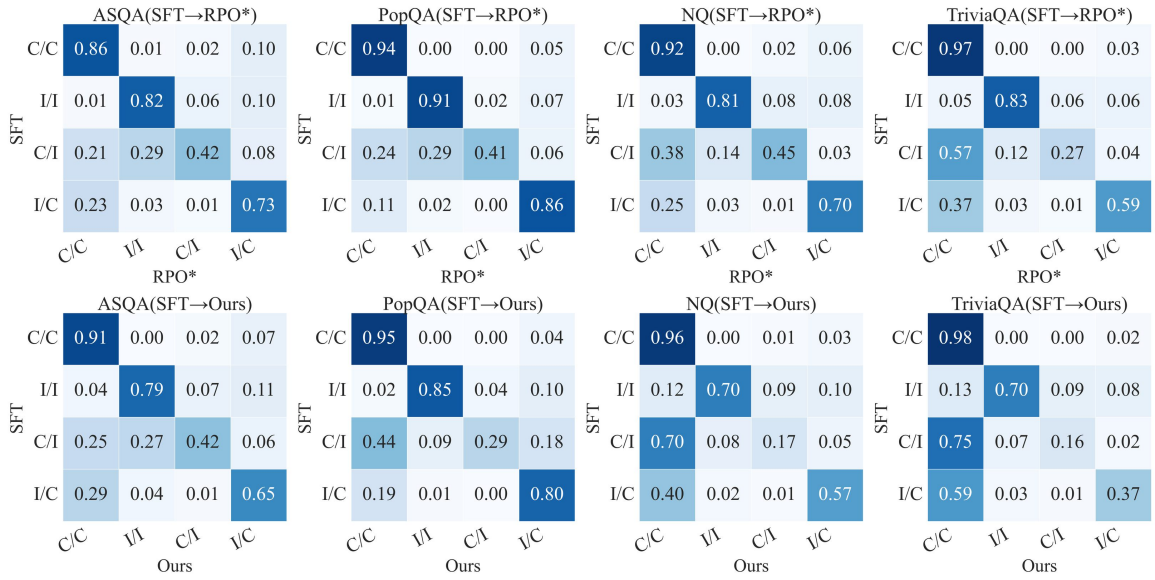

Figure 3: Category transition results of our method and RPO* across datasets. Each state is denoted as X/Y, where X and Y indicate the correctness of the direct and retrieval-augmented answers, respectively (C : correct, I : incorrect).

correct), our method reduces the proportion that remain incorrect and converts more into other states, indicating superior error correction; (2) for C/I samples (correct direct, incorrect retrieval), our method promotes transitions to C/C while suppressing degradation to worse states like I/I, effectively mitigating errors introduced by noisy retrieval; (3) for I/C samples (incorrect direct, correct retrieval), our method further encourages upgrades to C/C demonstrating an ability to advance from "retrieval-dependent correctness" to full reliability.

6 Conclusion

This paper identifies a key limitation in retrieval preference optimization (RPO) for open-domain RAG: while it effectively improves path routing, it offers limited direct supervision for answer quality within a chosen path, resulting in unstable correctness and consistency. To address this, we propose Joint Path-Answer Preference Optimization (JPAPO), a novel framework that jointly optimizes both path routing and within-path answering, building upon a retrieval supervised fine-tuning base. Extensive experiments on PopQA, NaturalQuestions, ASQA and TriviaQA demonstrate consistent improvements over strong baselines. These gains are further validated through ablation studies and in-depth analyses of probability rankings, probability gaps and category transitions.

7 Limitations

A limitation of our method is its reliance on repeated sampling to construct three distinct types of preference pairs, which increases computational overhead. Furthermore, our evaluation has been conducted on only four open-domain QA tasks. Future work should assess the method's generality using larger backbone models and a broader range of datasets.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics: ACL 2024*, pages 3563–3578.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pages 874–880.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2023. Large language models with controllable working memory. In *Findings of the association for computational linguistics: ACL 2023*, pages 1774–1793.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. *arXiv preprint arXiv:2109.05052*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. 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)*, pages 9802–9822.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. *arXiv preprint arXiv:2204.06092*.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S Siegel, Michael Tang, and 1 others. 2024. Bright: A realistic and challenging benchmark for reasoning-intensive retrieval. *arXiv preprint arXiv:2407.12883*.

Kai Sun, Yifan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2024. Head-to-tail: How knowledgeable are large language models (llms)? aka will llms replace knowledge graphs? In *Proceedings of the 2024 conference of the North American chapter of the association for computational linguistics: human language technologies (volume 1: long papers)*, pages 311–325.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan O Arik. 2025. Astute rag: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30553–30571.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2024. Instructrag: Instructing retrieval-augmented generation via self-synthesized rationales. *arXiv preprint arXiv:2406.13629*.

Chong Xiang, Tong Wu, Zexuan Zhong, David Wagner, Danqi Chen, and Prateek Mittal. 2024. Certifiably robust rag against retrieval corruption. *arXiv preprint arXiv:2405.15556*.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024a. Knowledge conflicts for llms: A survey. *arXiv preprint arXiv:2403.08319*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024b. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation.

Shi-Qi Yan, Quan Liu, and Zhen-Hua Ling. 2025. Rpo: Retrieval preference optimization for robust retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5228–5240.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.