

DVCQR: Dual-View Conversational Query Rewriting with Stage-wise Reinforcement Learning

Chenyi Li^{1,2,3}, Xinhui Tu^{1,2,3*}, Zaixiang Wang^{1,2,3}

¹Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning, Central China Normal University, Wuhan, China

²School of Computer, Central China Normal University, Wuhan, China

³National Language Resources Monitor and Research Center for Network Media, Central China Normal University, Wuhan, China

chenyi_li@mails.ccnu.edu.cn, tuxinhui@ccnu.edu.cn, 2316167644@mails.ccnu.edu.cn

Abstract

Conversational query rewriting (CQR) addresses context dependence in conversational search by rewriting each user query into a standalone form. Recent approaches leverage reinforcement learning (RL) to directly optimize retrieval effectiveness; however, they typically rely on a single rewrite, which struggles to accommodate the divergent preferences of sparse and dense retrievers and often suffers from conflicting optimization signals. We propose DVCQR, a Dual-View CQR framework that explicitly generates two complementary rewrites for each query: a sparse-view rewrite that emphasizes distinctive lexical anchors, and a dense-view rewrite that captures complete semantic constraints. Both rewrites are produced in a single pass via a structured reasoning process. To further mitigate objective conflicts, we introduce a stage-wise RL strategy that sequentially aligns the sparse and dense views with their corresponding retrievers using rank-based feedback. Extensive experiments on four benchmarks (TopiOCQA, QReCC, CAsT-19, and CAsT-20) demonstrate that DVCQR consistently outperforms state-of-the-art methods on most metrics under both sparse and dense retrieval settings, validating the effectiveness of dual-view rewriting and stage-wise retriever alignment.

1 Introduction

Conversational search has emerged as an increasingly important information retrieval paradigm with the rise of large language models (LLMs). It allows users to seek information through multi-turn dialogues with a system (Radlinski and Craswell, 2017; Dalton et al., 2020a; Gou et al., 2021). How-

*Corresponding author.

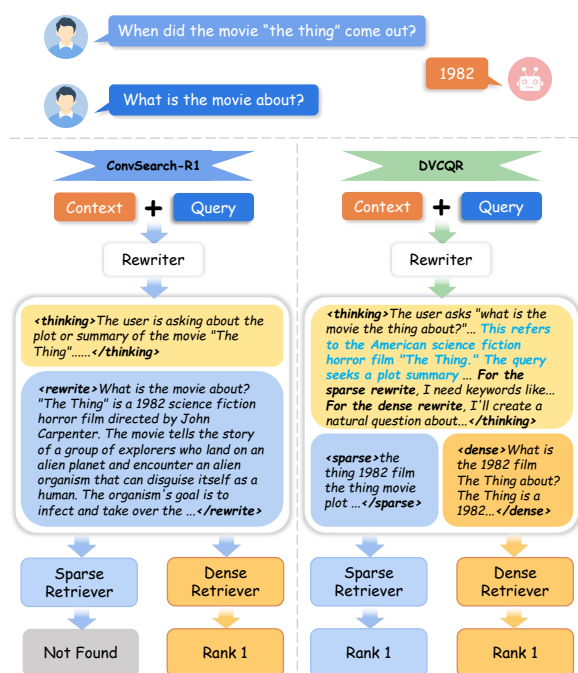


Figure 1: Comparison between ConvSearch-R1 and DVCQR. Rank 1 means the correct document is retrieved first, while Not Found means it is outside the top-100 results.

ever, conversational queries are often highly dependent on dialogue context, which introduces challenges such as coreference, ambiguity, and ellipsis, making it difficult to accurately infer the user’s underlying search intent (Qu et al., 2020; Anantha et al., 2021; Adlakha et al., 2022). CQR addresses this challenge by transforming the original context-dependent query into a self-contained query that can be handled by standard retrievers (Elgohary et al., 2019; Vakulenko et al., 2021; Yu et al., 2020).

Recently, ConvSearch-R1 (Zhu et al., 2025) leverages retrieval-feedback RL to generate a single rewrite via explicit chain-of-thought reasoning,

achieving strong gains on benchmarks such as TopiOCQA (Adlakha et al., 2022) and QReCC (Anantha et al., 2021) (As shown in Figure 1). However, practical retrieval systems typically combine sparse and dense retrievers (Luo et al., 2023; Arivazhagan et al., 2023), whose query-form preferences differ fundamentally: sparse retrievers (e.g., BM25 (Robertson and Zaragoza, 2009)) rely on compact lexical anchors, whereas dense retrievers (e.g., ANCE (Xiong et al., 2021)) favor semantically explicit natural-language constraints. This discrepancy induces a structural form tension, where a single rewrite is systematically pulled between incompatible objectives, making it difficult to serve both paradigms well. A natural solution is to jointly optimize one rewrite using a scalarized reward from both retrievers, yet in practice sparse and dense rank-based signals frequently conflict and partially cancel during training, yielding weak effective learning progress and optimization plateaus (§ 4.5). More broadly, compressing heterogeneous retriever preferences into a single output and a single scalar objective risks objective interference and unstable optimization.

These observations motivate a two-fold decoupling principle for retriever-aligned CQR: decouple the output space by producing retriever-specific rewrites, and decouple the optimization process to reduce reward conflict. Accordingly, we propose **DVCQR**, a **Dual-View CQR** framework, which uses structured reasoning to generate, in a single pass, a sparse-view rewrite emphasizing lexical anchors and a dense-view rewrite emphasizing semantic constraints. DVCQR further employs stage-wise retrieval-feedback RL to sequentially align the two views with their corresponding retrievers, mitigating objective conflicts and yielding consistent improvements on TopiOCQA, QReCC, CAsT-19 (Dalton et al., 2020b), and CAsT-20 (Dalton et al., 2020a). Notably, DVCQR does not imply heavier inference: each view is intentionally concise and retriever-tailored, which reduces issued tokens and improves sparse retrieval latency (Appendix I).

Our contributions are summarized as follows:

- We propose DVCQR, a dual-view structured reasoning and generation framework, which explicitly produces the reasoning process together with a sparse-view rewrite and a dense-view rewrite in a single generation pass.
- We introduce a stage-wise RL scheme from retrieval feedback for dual-view rewrites, which

continuously optimizes the sparse view and the dense view in successive stages to mitigate objective conflicts.

- We conduct systematic experiments on benchmarks and the results show that DVCQR achieves consistent improvements across multiple retrieval metrics.

2 Related Work

2.1 Conversational Search

Conversational search enables users to seek information through multi-turn dialogue (Radlinski and Craswell, 2017), but conversational queries are often context-dependent, causing coreference ambiguity, ellipsis, and underspecified intent. Existing approaches mainly fall into Conversational Dense Retrieval (CDR) and Conversational Query Rewriting (CQR). CDR fine-tunes a dense retriever to encode full dialogue context, which increases computation and couples the solution to a specific retriever (Yu et al., 2021; Robertson and Zaragoza, 2009). In contrast, CQR rewrites each context-dependent query into a self-contained query that can be issued to standard retrieval systems (Mao et al., 2023b).

Early CQR work trained sequence-to-sequence rewriters on human rewrite annotations (Lin et al., 2020). Later studies improved controllability and effectiveness by leveraging retrieval feedback, including iterative refinement, preference-based objectives, and RL over ranking outcomes (Wu et al., 2022; Ye et al., 2023; Mo et al., 2023; Mao et al., 2023a; Jang et al., 2024; Zhang et al., 2024; Yoon et al., 2025; Zhu et al., 2025). With LLMs, prompt-based and distillation-based strategies further enable richer rewrites or multiple candidates per turn (Mao et al., 2023b; Lai et al., 2025a; Mo et al., 2024). Moreover, combining sparse and dense retriever signals during training can improve robustness across retrieval settings (Lai et al., 2025b). Different from prior single-rewrite methods, DVCQR generates sparse-view and dense-view rewrites and aligns them with corresponding retrievers via stage-wise RL.

2.2 RLVR-based Retrieval

Reinforcement Learning from Verifiable Rewards (RLVR) has recently been explored as a means to enhance LLM reasoning and tool-use. The core idea is to use an automated verifier or critic to

score a model’s output and treat that score as a reward signal for policy optimization, thus enabling large-scale fine-tuning without requiring human-provided labels (Guo et al., 2025). In the context of information retrieval, several works have integrated RLVR into retrieval-augmented generation frameworks to encourage better query planning and execution. For example, Search-R1 (Jin et al., 2025) uses RL to optimize multi-turn search queries and reasoning paths for open-domain QA, and Deep-Retrieval (Jiang et al., 2025) trains an agent via RL to interact with a live search engine alongside a learned dense retriever. In conversational search, ConvSearch-R1 (Zhu et al., 2025) similarly applies RL to coordinate the reasoning process with retrieval actions over multiple turns. These studies demonstrate that using verifiable retrieval feedback as an optimization signal can significantly improve retrieval effectiveness and reasoning performance without manual annotation.

3 Methodology

3.1 Task Formulation

Given a conversational context $C = \{(q_i, r_i)\}_{i=1}^{n-1}$ and the current user query q_n , where (q_i, r_i) is the i -th query–response pair. CQR aims to generate a context-independent query Q_n that fully reflects the current user intent. In DVCQR, the model produces dual rewrites together with an explicit reasoning trace:

$$T, S, D \leftarrow \text{DVCQR}(C, q_n), \quad (1)$$

where T denotes the reasoning trace, S is a sparse-view rewrite for sparse retrieval, and D is a dense-view rewrite for dense retrieval.

3.2 Overview

Figure 2 illustrates the three-stage training pipeline of DVCQR. In Stage I, we perform supervised dual-view warm-up by distilling teacher-generated reasoning traces and dual rewrites, yielding a base policy π_{SFT} that produces a reasoning trace T along with sparse and dense rewrites (S, D) . Stage II applies GSPO-based RL with sparse retriever feedback to optimize S for lexical alignment. Stage III further applies GSPO using dense retriever feedback to optimize D for semantic matching. By sequentially optimizing sparse and dense views, DVCQR decouples conflicting objectives and produces a final policy π_{D} that generates retriever-

aligned dual rewrites guided by coherent intermediate reasoning.

3.3 Dual-View Structured Reasoning

We propose a structured reasoning prompt that explicitly decomposes conversational query rewriting into sequential steps, producing an interpretable reasoning trace T alongside two complementary rewrites. The model first resolves contextual intent by identifying the target entity and queried aspect. It then derives a sparse-view rewrite S by aggregating salient lexical anchors into a compact keyword-style query, and a dense-view rewrite D as a concise natural-language question augmented with a brief answer that preserves intent, makes constraints explicit, and avoids ambiguous references. The prompt is shown in Appendix Table 11.

3.4 Stage I: Supervised Dual-View Warm-Up

Following prior findings that supervised warm-up is critical for stabilizing retrieval-feedback RL (Zhu et al., 2025), we first perform a supervised alignment stage to initialize the model with correct task behavior and output structure.

Dual-View Distillation A stronger teacher LLM is prompted with (C, q_n) to generate a reasoning trace and dual-view rewrites (T, S, D) . We retain only instances where the gold passage is ranked first by both sparse (S) and dense (D) retrievers. Details are provided in Appendix B.

Supervised Fine-Tuning We then fine-tune the rewriter using (C, q_n) as input, supervising it to reproduce the distilled (T, S, D) . This produces π_{SFT} , which consistently follows the structured format and provides a strong, task-aligned initialization for subsequent RL. The training prompt is shown in Appendix Table 13.

3.5 Stage II: Sparse-Optimized RL

Effective reward design is essential for stable RL (Devidze et al., 2021). Following rank-incentive shaping (Zhu et al., 2025), we adopt a rank-based main reward, and augment it with multi-aspect auxiliary rewards to densify supervision (Wang et al., 2025). In addition, we include a format reward R_f to discourage invalid outputs, ensuring adherence to the required rewrite structure (see Appendix C for details).

Reward Design In Stage II, we optimize the sparse rewrite S using a rank-based reward

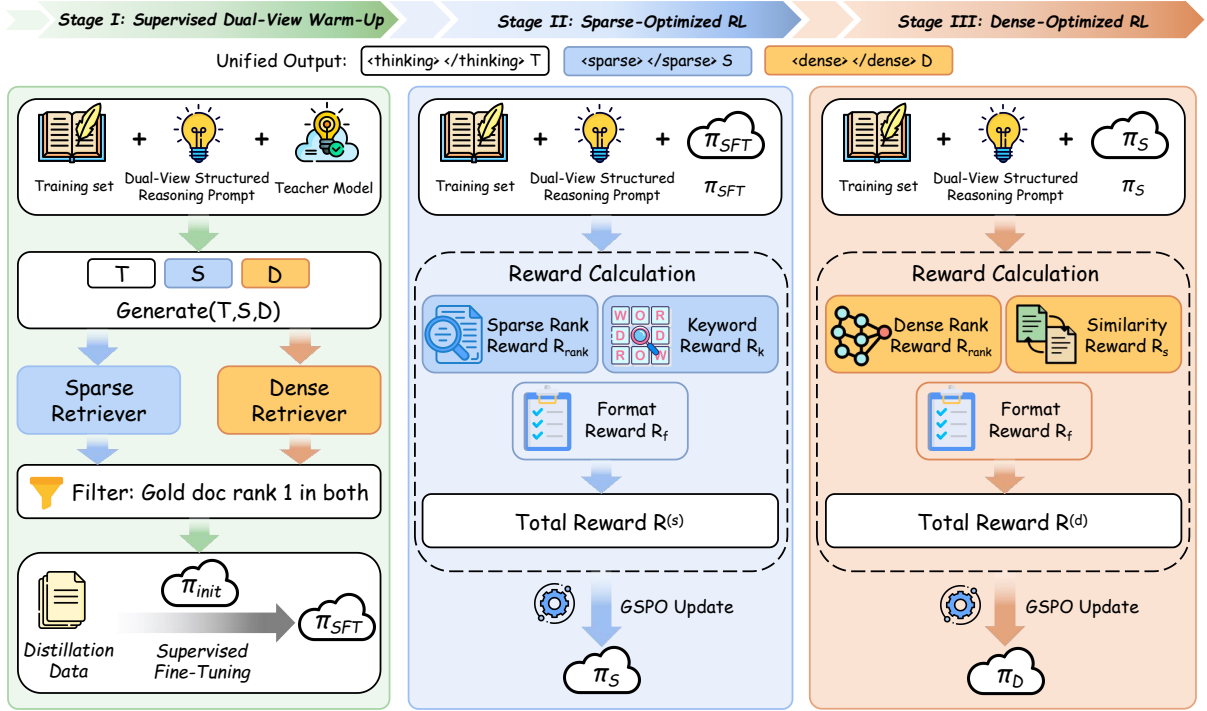


Figure 2: Overview of DVCQR framework.

$R_{rank}(r_s)$, where r_s denotes the sparse retriever rank of the gold passage. The reward assigns higher values to top-ranked results ($r_s \in [1, 10]$), reduced but non-zero values for lower-ranked yet relevant results ($r_s \in (10, 100]$), and zero reward otherwise:

$$R_{rank}(r_s) = \begin{cases} 1 - 0.5 \left(\frac{r_s-1}{9}\right)^{0.7}, & 1 \leq r_s \leq 10, \\ 0.5 \left(\frac{100-r_s}{90}\right)^{1.3}, & 10 < r_s \leq 100, \\ 0, & r_s > 100, \end{cases} \quad (2)$$

We further introduce an auxiliary keyword-coverage reward R_k to encourage the sparse rewrite S to include salient topical cues. Using the ground-truth topic and topic section annotations, we extract informative keywords and form two sets: entity-related terms E and aspect-related terms A . The coverage of S over these sets is defined as:

$$Cov_E = \frac{|S \cap E|}{|E|}, \quad Cov_A = \frac{|S \cap A|}{|A|}, \quad (3)$$

$$R_k = 0.5 \cdot Cov_E + 0.5 \cdot Cov_A,$$

where Cov_E and Cov_A quantify the proportion of ground-truth entity and aspect keywords, respectively, that are covered by the sparse rewrite S .

The total reward in Stage II is defined as a weighted combination of the rank-based, and keyword-coverage rewards:

$$R^{(s)} = w_1 \cdot R_{rank}(r_s) + w_2 \cdot R_k, \quad (4)$$

where w_1 , and w_2 are scalar coefficients controlling the contribution of each term. This reward is optimized using the GSPO algorithm (§ 3.7) to train the model.

3.6 Stage III: Dense-Optimized RL

Reward Design In Stage III, we optimize the dense rewrite using dense-retriever feedback. We reuse the rank-based reward defined in Eq. 2, computing it with the dense-retrieval rank r_d of the gold passage, which serves as the primary optimization signal for semantic alignment.

For the dense rewrite, we further introduce a semantic similarity auxiliary reward R_s . We encode the gold passage into a normalized embedding \mathbf{G}' using a semantic encoder, and similarly encode the model-generated dense rewrite D into \mathbf{D}' . The auxiliary reward is defined as the non-negative cosine similarity between these representations:

$$R_s = \max\left(0, \cos(\mathbf{D}', \mathbf{G}')\right), \quad (5)$$

where $\cos(\mathbf{D}', \mathbf{G}')$ is the cosine similarity between the normalized dense rewrite and gold passage representations.

The total reward in Stage III mirrors that of Stage II, combining rank-based, and semantic similarity rewards:

$$R^{(d)} = w_1 \cdot R_{rank}(r_d) + w_2 \cdot R_s, \quad (6)$$

which is optimized using the GSPO algorithm (§ 3.7) to further fine-tune the model.

3.7 GSPO Training Algorithm

We optimize all RL stages using GSPO (Zheng et al., 2025). Similar to GRPO (Shao et al., 2024), GSPO employs within-group normalized advantages, but defines and clips the importance ratio at sequence level, which improves stability for long outputs and large models. Given G candidate rewrites $\{y_i\}_{i=1}^G$ sampled from the old policy under the same input x , the objective is:

$$J_{\text{GSPO}}(\theta) = E \left[\frac{1}{G} \sum_{i=1}^G \min \left(s_i(\theta) A_i^b, \text{clip}(s_i(\theta), 1 - \epsilon, 1 + \epsilon) A_i^b \right) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right], \quad (7)$$

where $s_i(\theta) = \left(\frac{\pi_\theta(y_i|x)}{\pi_{\theta_{\text{old}}}(y_i|x)} \right)^{1/|y_i|}$, $A_i^b = \frac{r(x, y_i) - \text{mean}_j r(x, y_j)}{\text{std}_j r(x, y_j)}$, ϵ is the clipping range, β controls KL regularization, and π_{ref} is the reference policy. In Stage II, we initialize π_θ from π_{SFT} with $\pi_{\text{ref}} = \pi_{\text{SFT}}$ to obtain π_S ; in Stage III, we initialize from π_S with $\pi_{\text{ref}} = \pi_S$, yielding the final policy π_D .

4 Experiments

4.1 Experimental Setup

Settings Following Zhu et al. (2025), we evaluate on TopiOCQA (Adlakha et al., 2022), QReCC (Anantha et al., 2021), CAsT-19 (Dalton et al., 2020b), and CAsT-20 (Dalton et al., 2020a). We use BM25 as the sparse retriever and ANCE (Xiong et al., 2021) as the dense retriever, reporting single run MRR, NDCG@3, and Recall@K (R@K). Details are in Appendix D.1 and D.2.

Baselines We categorize baselines into three groups: (1) CQR methods without explicit retriever alignment, including T5QR (Lin et al., 2020), InfoCQR (Ye et al., 2023), LLM4CS (Mao et al., 2023b), and CHIQ-Fusion (Mo et al., 2024); (2) approaches aligned to a single retriever, such as CONQRR (Wu et al., 2022), EDIRCS (Mao et al., 2023a), ConvGQR (Mo et al., 2023), IterCQR (Jang et al., 2024), AdaQR (Zhang et al., 2024), RETPO (Yoon et al., 2025), and ConvSearch-R1 (Zhu et al., 2025); and (3) methods jointly aligned to both sparse and dense retrievers, including AdaCQR (Lai et al., 2025b) and AdaRewriter (Lai et al., 2025a). Additional details are provided in Appendix D.3.

Implementation Details We adopt DeepSeek-V3.2-Exp (DeepSeek-AI, 2025) as the teacher model and use Qwen2.5-3B and Qwen3-4B as rewriter backbones. BM25 and ANCE retrievers are implemented with PyTerrier (Macdonald and Tonellotto, 2020). Stage I employs LLaMA-Factory (Zheng et al., 2024) for supervised fine-tuning, while Stages II and III use MS-Swift (Zhao et al., 2025) for GSPO-based RL. Further details are provided in Appendix D.4.

4.2 Main Results

Table 1 reports results on TopiOCQA and QReCC under both sparse retrieval (BM25) and dense retrieval (ANCE). Overall, DVCQR achieves consistent gains across both retriever types, and attains the best performance on top-ranked effectiveness metrics such as MRR, NDCG@3, and R@10.

Compared with the strongest reasoning-based RL baseline leveraging retrieval feedback, ConvSearch-R1, DVCQR with a Qwen2.5-3B backbone consistently improves performance under both sparse (BM25) and dense (ANCE) retrieval settings. When equipped with a stronger backbone, Qwen3-4B, the performance margins further widen. In addition, relative to AdaCQR and AdaRewriter, which enforce a single unified rewrite for both retrievers, DVCQR achieves superior effectiveness on the majority of evaluation metrics. These results substantiate our core motivation: explicitly decoupling lexical anchors and semantic constraints into dual-view rewrites, and reinforcing them through stage-wise retrieval feedback, effectively mitigates the objective interference inherent in a single-rewrite paradigm. See Appendix A for a detailed paradigm-level discussion. Notably, even with a parameter scale comparable to ConvSearch-R1 (3B-4B), DVCQR outperforms multiple substantially larger LLM-based baselines across most metrics. We also analyze efficiency and find that DVCQR cuts issued tokens and speeds up BM25 (Appendix I).

4.3 Cross-Dataset Generalization

Table 2 reports zero-shot cross-dataset evaluation results on CAsT-19 and CAsT-20. Overall, DVCQR remains effective under out-of-domain conditions and consistently preserves strong performance for both BM25 and ANCE. Under a fair comparison with a Qwen2.5-3B backbone, DVCQR yields stable and consistent gains across datasets. When scaled to a stronger backbone,

Type	Method	Backbone	TopiOCQA				QReCC			
			MRR	NDCG@3	R@10	R@100	MRR	NDCG@3	R@10	R@100
Sparse (BM25)	T5QR	T5-Base	11.3	9.8	22.1	44.7	33.4	30.2	53.8	86.1
	CONQRR	T5-Base	-	-	-	-	38.3	-	60.1	88.9
	EDIRCS	T5-Base	-	-	-	-	41.2	-	62.7	90.2
	ConvGQR	T5-Base	12.4	10.7	23.8	45.6	44.1	41.0	64.4	88.0
	IterCQR	T5-Base	16.5	14.9	29.3	54.1	46.7	44.1	64.4	85.5
	InfoCQR	ChatGPT-3.5	-	-	-	-	48.9	46.3	66.4	-
	LLM4CS	ChatGPT-3.5	18.9	17.7	33.7	-	47.8	45.0	69.1	-
	AdaQR	Mistral-7B	20.3	18.0	37.1	66.2	50.6	48.0	69.6	-
	RETPO	LLaMA-2-7B	28.3	26.5	48.3	73.1	50.0	47.3	69.5	89.5
	CHIQ-Fusion	LLaMA-2-7B	25.6	23.5	44.7	-	54.3	51.9	78.5	-
	AdaCQR	T5-Base	28.3	26.5	48.9	71.2	55.1	52.5	76.5	93.7
	AdaRewriter	LLaMA-3.1-8B	30.7	28.8	51.3	-	56.2	53.8	78.8	-
	ConvSearch-R1	Qwen2.5-3B	35.2	33.5	57.8	79.9	56.5	54.8	76.3	88.1
	DVCQR	Qwen2.5-3B	37.4	<u>35.7</u>	60.3	82.8	<u>57.9</u>	56.0	77.2	89.9
DVCQR	Qwen3-4B	39.5	38.0	63.7	84.1	59.4	57.6	79.4	<u>91.5</u>	
Dense (ANCE)	T5QR	T5-Base	23.0	22.2	37.6	54.4	34.5	31.8	53.1	72.8
	CONQRR	T5-Base	-	-	-	-	41.8	-	65.1	84.7
	EDIRCS	T5-Base	-	-	-	-	42.1	-	65.6	85.3
	ConvGQR	T5-Base	25.6	24.3	41.8	58.8	42.0	39.1	63.5	81.8
	IterCQR	T5-Base	26.3	25.1	42.6	62.0	42.9	40.2	65.5	84.1
	InfoCQR	ChatGPT-3.5	-	-	-	-	43.9	41.3	65.6	-
	LLM4CS	ChatGPT-3.5	27.7	26.7	43.3	-	44.8	42.1	66.4	-
	AdaQR	Mistral-7B	38.1	36.6	61.3	79.9	43.4	40.8	65.6	-
	RETPO	LLaMA-2-7B	30.0	28.9	49.6	68.7	44.0	41.1	66.7	84.6
	CHIQ-Fusion	LLaMA-2-7B	38.0	37.0	61.6	-	47.2	44.2	<u>70.7</u>	-
	AdaCQR	T5-Base	38.5	37.6	58.4	75.0	45.8	42.9	67.3	83.8
	AdaRewriter	LLaMA-3.1-8B	40.3	39.7	61.9	-	47.5	44.7	69.8	-
	ConvSearch-R1	Qwen2.5-3B	51.4	51.3	72.0	85.7	49.7	47.7	69.8	81.6
	DVCQR	Qwen2.5-3B	<u>52.5</u>	<u>52.1</u>	<u>72.7</u>	<u>86.6</u>	<u>50.9</u>	48.8	<u>70.7</u>	82.9
DVCQR	Qwen3-4B	53.5	52.9	73.4	87.8	52.3	50.1	71.4	84.3	

Table 1: Main results on TopiOCQA and QReCC. **Bold** and underlined numbers indicate the best and second-best results, respectively.

Type	Method	Backbone	CAst-19				CAst-20			
			MRR	NDCG@3	R@10	R@100	MRR	NDCG@3	R@10	R@100
Sparse (BM25)	ConvSearch-R1	Qwen2.5-3B	54.6	29.9	8.2	26.7	38.5	16.8	6.9	<u>23.4</u>
	DVCQR	Qwen2.5-3B	55.4	29.6	8.8	<u>27.5</u>	<u>39.3</u>	<u>17.1</u>	<u>7.1</u>	23.3
	DVCQR	Qwen3-4B	60.3	<u>29.8</u>	9.4	29.6	40.6	18.0	7.5	23.6
Dense (ANCE)	CHIQ-Fusion	LLaMA-2-7B	73.3	50.5	12.9	-	54.0	38.0	<u>19.3</u>	-
	AdaCQR	T5-Base	74.5	-	13.8	39.2	56.6	-	19.2	45.6
	AdaRewriter	LLaMA-3.1-8B	-	48.3	13.0	-	63.0	46.5	21.6	-
	ConvSearch-R1	Qwen2.5-3B	75.1	<u>52.3</u>	<u>14.2</u>	40.3	<u>70.4</u>	42.7	15.8	44.2
	DVCQR	Qwen2.5-3B	75.6	<u>52.3</u>	14.3	<u>40.4</u>	<u>70.4</u>	<u>42.9</u>	16.0	44.0
DVCQR	Qwen3-4B	76.2	52.5	14.0	40.8	70.8	41.9	16.2	<u>44.4</u>	

Table 2: Zero-shot cross-dataset generalization results on CAst-19 and CAst-20. **Bold** and underlined numbers denote the best and second-best results, respectively.

Qwen3-4B, these advantages further expand, with particularly notable improvements under BM25, while performance under ANCE also remains robustly ahead. These findings indicate that DVCQR’s dual-view decomposition and stage-wise optimization strategy generalize well across datasets, enabling sustained alignment with sparse

and dense retriever preferences despite shifts in conversational topics and expression distributions. Additional evidence on cross-retriever generalization is provided in Appendix E.

Type	Method	TopiOCQA				QReCC			
		MRR	NDCG@3	R@10	R@100	MRR	NDCG@3	R@10	R@100
Sparse (BM25)	Qwen3-4B	13.8	12.0	26.3	46.5	30.6	28.5	46.2	67.9
	w/ Stage I	19.0	17.3	34.3	59.5	32.3	30.0	48.6	68.8
	w/ Stage I + Stage II	38.1	36.3	61.9	83.7	58.1	56.3	78.3	89.5
	w/ Stage I + Stage II + Stage III	39.5	38.0	63.7	84.1	59.4	57.6	79.4	91.5
Dense (ANCE)	Qwen3-4B	24.8	23.7	40.3	55.2	33.2	31.1	48.9	62.8
	w/ Stage I	41.9	41.0	63.0	79.7	38.0	35.5	57.1	74.0
	w/ Stage I + Stage II	48.8	48.3	69.5	85.4	48.2	46.1	68.0	81.2
	w/ Stage I + Stage II + Stage III	53.5	52.9	73.4	87.8	52.3	50.1	71.4	84.3

Table 3: Ablation results of DVCQR across training stages. **Bold** numbers indicate the best results.

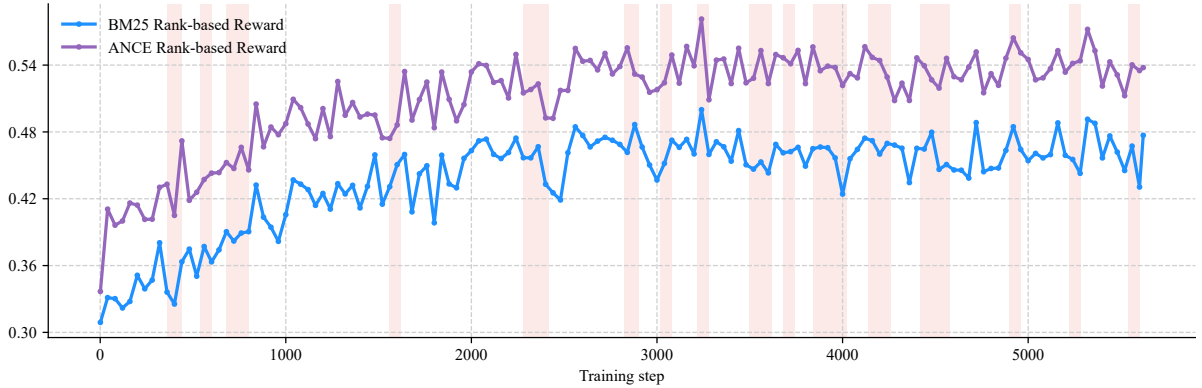


Figure 3: Reward dynamics on TopiOCQA for the single-rewrite JointRL setting. Pink regions indicate entanglement–cancellation intervals, where the BM25 and ANCE rank-based rewards change in opposite directions while the scalarized total reward exhibits minimal variation ($\rho(t) < 0.22$).

4.4 Ablation Study

Table 3 presents the stage-wise ablation results of DVCQR, revealing a clear overall trend that all three training stages are indispensable. First, the dual-view supervised warm-up in Stage I establishes a stable, task-aligned initialization for subsequent RL, leading to substantial performance improvements under both retrieval settings. Building on this foundation, Stage II serves as the primary driver of sparse-retrieval gains, producing the largest performance increase on BM25 while also delivering clear cross-view improvements on ANCE. Stage III further reinforces dense retrieval, advancing ANCE to its best final performance, and continues to yield additional benefits for BM25. These results confirm the effectiveness of a "warm-up-first, stage-wise alignment" strategy for sparse and dense preferences: Stage II dominates sparse improvements, Stage III is critical for dense improvements, and the two RL stages exhibit a clear cross-view synergy, which we analyze further in § 4.6. A detailed reward analysis is provided in Appendix F.

4.5 Diagnosing Sparse–Dense Preference Conflict in JointRL

We refer to the joint-scalarized RL setting as JointRL, where sparse and dense retriever feedback are combined into a single scalar reward and optimized within one RL stage. We analyze Single-Rewrite JointRL, where a single rewrite is optimized using a scalar reward $R(t)$ that combines sparse and dense ranking rewards, denoted as $R_s(t)$ and $R_d(t)$, together with a format reward. Let t^- denote the previous logging step. We define step-to-step reward changes as $\Delta R_s(t)$, $\Delta R_d(t)$, and $\Delta R(t)$, respectively. A local conflict is identified when $\Delta R_s(t)$ and $\Delta R_d(t)$ exhibit opposite signs, indicating competing optimization signals. To quantify the degree of cancellation, we introduce a ratio $\rho(t)$ that compares $|\Delta R(t)|$ against $|\Delta R_s(t)| + |\Delta R_d(t)|$, where a smaller $\rho(t)$ implies stronger cancellation between sparse and dense objectives. We further aggregate consecutive conflict steps with low $\rho(t)$ into entanglement-cancellation intervals to characterize sustained interference. Formal definitions and detailed equations are provided in Appendix G.

As illustrated in Figure 3, reward cancellation

in JointRL is both frequent and persistent. Across training steps 1–5620, we identify 16 entanglement-cancellation intervals spanning 1480 steps (26.3%), with 12 intervals occurring after step 2000. Within these intervals, changes in the scalar reward are markedly smaller than those of the individual sub-rewards (on average $|\Delta R| \approx 0.004$ versus $|\Delta R_s| \approx 0.014$ and $|\Delta R_d| \approx 0.016$), resulting in a low mean cancellation ratio of $\bar{\rho} \approx 0.13$. These observations indicate that antagonism between sparse and dense objectives substantially weakens the effective learning signal. This behavior helps explain the inferior offline retrieval performance of Single-Rewrite JointRL (see Appendix H for a detailed comparison between JointRL and Stage-wise RL), and directly motivates our design to decouple sparse and dense preferences both in the output space, via dual-view rewrites, and in the optimization process, via stage-wise RL.

4.6 Training Trajectory Analysis: Synergy Window in Stage-wise RL

Figure 4 presents the checkpoint-wise MRR trajectories on TopiOCQA for BM25 and ANCE. During Stage II, BM25 MRR increases sharply, while ANCE also shows clear improvements, despite this stage being optimized exclusively with sparse retrieval feedback. This suggests that reinforcing lexical anchors and explicit constraint expressions can simultaneously enhance dense retrievability. After transitioning to Stage III, ANCE exhibits more sustained and concentrated gains, continuing to rise steadily after a brief fluctuation at the stage boundary. In contrast, BM25 benefits synergistically during the early portion of Stage III and reaches its peak, but slightly declines at later checkpoints as ANCE continues to improve. This pattern indicates a stable post-transition *synergy window* in which both sparse and dense effectiveness improve concurrently, followed by mild preference drift as optimization increasingly emphasizes the dense objective, resulting in marginal regression on the sparse side. Complete results for all four evaluation metrics are provided in Appendix Figure 5.

4.7 Case Study

We further compare ConvSearch-R1 with DVCQR using two cases in the appendix (Table 14 & 15). In the first case, ConvSearch-R1 makes a critical coreference error in the reasoning stage, misattributing “her” to Sadie Harris, which shifts the entity focus and results in Not Found for both

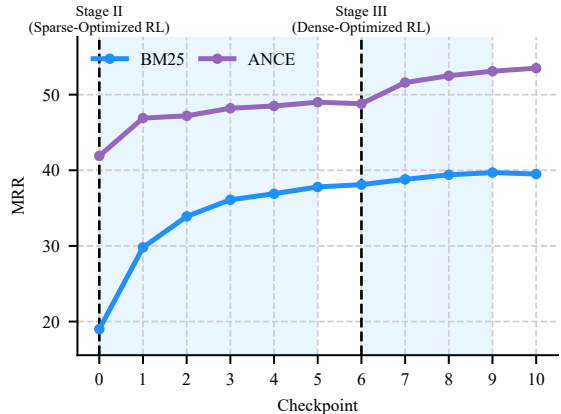


Figure 4: Checkpoint-wise MRR on TopiOCQA for BM25 and ANCE retrieval. The blue shaded region highlights the synergy window in which BM25 and ANCE improve concurrently.

BM25 and ANCE; in contrast, DVCQR resolves the coreference during reasoning, and uses the sparse rewrite to provide highly discriminative lexical anchors while keeping the dense rewrite semantically focused, enabling both retrievers to reach Rank 1. This case directly shows that DVCQR’s dual-view structured reasoning is of higher quality and better tailored to the CQR task. In the second case, a single rewrite is more likely to produce lengthy, explanatory text with unstable lexical cues, leading to a mismatch for BM25 while ANCE can still hit the gold passage; DVCQR explicitly decouples the output forms of sparse and dense, allowing both retrievers to achieve Rank 1. This case suggests that a single rewrite can easily become imbalanced between “informativeness” and “lexical anchors”, whereas DVCQR’s explicit output decoupling makes it easier to obtain stable gains under a dual-retriever setting.

5 Conclusion

We propose DVCQR, a dual-view reasoning framework for conversational query rewriting that explicitly aligns the query-form preferences of sparse and dense retrievers. During the reasoning stage, DVCQR decomposes retrieval cues into lexical anchors and semantic constraints, and jointly generates a reasoning trace, a sparse-oriented keyword rewrite, and a dense-oriented natural-language rewrite, thereby decoupling the inherent preference tension between sparse and dense retrieval. To mitigate objective conflicts during optimization, we introduce a stage-wise RL paradigm based

on retrieval feedback, employing GSPO for stable training. Extensive experiments on TopiOCQA, QReCC, and CAsT-19/20 demonstrate consistent improvements under both BM25 and ANCE settings, validating the effectiveness of our dual-view decomposition and stage-wise alignment strategy.

Limitations

Although DVCQR achieves strong retrieval effectiveness, it has several limitations. **(1) Dual-view specificity.** DVCQR is designed for hybrid retrieval settings and consistently generates both a sparse-view rewrite and a dense-view rewrite. As a result, it does not natively support producing a single unified rewrite, which may be preferable in deployments with a single retriever or strict single-query interfaces. **(2) Increased training cost.** Compared with a single-stage RL pipeline, the proposed stage-wise training strategy introduces additional RL stages and retrieval-feedback rollouts, leading to higher training time and computational overhead.

Acknowledgments

This work was supported by National Language and Character Research Base (ZDI145-168), the National Natural Science Foundation of China (62472192) and the Self-determined Research Funds of CCNU from the Colleges' Basic Research and Operation of MOE (JC2026PT-002). Authors are grateful to the anonymous reviewers for helpful comments.

References

- Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. [TopiOCQA: Open-domain conversational question answering with topic switching](#). *Transactions of the Association for Computational Linguistics*, 10:468–483.
- Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. [Open-domain question answering goes conversational via question rewriting](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 520–534, Online. Association for Computational Linguistics.
- Manoj Ghuhan Arivazhagan, Lan Liu, Peng Qi, Xinchu Chen, William Yang Wang, and Zhiheng Huang. 2023. [Hybrid hierarchical retrieval for open-domain question answering](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10680–10689, Toronto, Canada. Association for Computational Linguistics.
- Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020a. [Cast 2020: The conversational assistance track overview](#). In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).
- Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020b. [Trec cast 2019: The conversational assistance track overview](#). *arXiv preprint arXiv:2003.13624*.
- DeepSeek-AI. 2025. [Deepseek-v3.2-exp: Boosting long-context efficiency with deepseek sparse attention](#).
- Rati Devidze, Goran Radanovic, Parameswaran Kamalaruban, and Adish Singla. 2021. [Explicable reward design for reinforcement learning agents](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 20118–20131. Curran Associates, Inc.
- Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. [Can you unpack that? learning to rewrite questions-in-context](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5918–5924, Hong Kong, China. Association for Computational Linguistics.
- Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. [Knowledge distillation: A survey](#). *Int. J. Comput. Vision*, 129(6):1789–1819.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *arXiv preprint arXiv:2501.12948*.
- Yunah Jang, Kang-il Lee, Hyunkyung Bae, Hwanhee Lee, and Kyomin Jung. 2024. [IterCQR: Iterative conversational query reformulation with retrieval guidance](#). 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 8121–8138, Mexico City, Mexico. Association for Computational Linguistics.
- 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*.

- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, 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*.
- Yilong Lai, Jialong Wu, Zhenglin Wang, and Deyu Zhou. 2025a. **AdaRewriter: Unleashing the power of prompting-based conversational query reformulation via test-time adaptation**. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3889–3905, Suzhou, China. Association for Computational Linguistics.
- Yilong Lai, Jialong Wu, Congzhi Zhang, Haowen Sun, and Deyu Zhou. 2025b. **AdaCQR: Enhancing query reformulation for conversational search via sparse and dense retrieval alignment**. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7698–7720, Abu Dhabi, UAE. Association for Computational Linguistics.
- Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. *arXiv preprint arXiv:2004.01909*.
- Man Luo, Shashank Jain, Anchit Gupta, Arash Einolghozati, Barlas Oguz, Debojeet Chatterjee, Xilun Chen, Chitta Baral, and Peyman Heidari. 2023. **A study on the efficiency and generalization of light hybrid retrievers**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1617–1626, Toronto, Canada. Association for Computational Linguistics.
- Craig Macdonald and Nicola Tonellotto. 2020. Declarative experimentation in information retrieval using pyterrier. In *Proceedings of ICTIR 2020*.
- Kelong Mao, Zhicheng Dou, Bang Liu, Hongjin Qian, Fengran Mo, Xiangli Wu, Xiaohua Cheng, and Zhao Cao. 2023a. **Search-oriented conversational query editing**. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4160–4172, Toronto, Canada. Association for Computational Linguistics.
- Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. 2023b. **Large language models know your contextual search intent: A prompting framework for conversational search**. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1211–1225, Singapore. Association for Computational Linguistics.
- Fengran Mo, Abbas Ghaddar, Kelong Mao, Mehdi Rezagholizadeh, Boxing Chen, Qun Liu, and Jian-Yun Nie. 2024. **CHIQ: Contextual history enhancement for improving query rewriting in conversational search**. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2268, Miami, Florida, USA. Association for Computational Linguistics.
- Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2023. **ConvGQR: Generative query reformulation for conversational search**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4998–5012, Toronto, Canada. Association for Computational Linguistics.
- Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. **Open-retrieval conversational question answering**. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20*, page 539–548, New York, NY, USA. Association for Computing Machinery.
- Filip Radlinski and Nick Craswell. 2017. **A theoretical framework for conversational search**. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval, CHIIR '17*, page 117–126, New York, NY, USA. Association for Computing Machinery.
- Stephen Robertson and Hugo Zaragoza. 2009. **The probabilistic relevance framework: Bm25 and beyond**. *Found. Trends Inf. Retr.*, 3(4):333–389.
- 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*.
- Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021. Question rewriting for conversational question answering. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 355–363.
- Yujing Wang, Hainan Zhang, Liang Pang, Binghui Guo, Hongwei Zheng, and Zhiming Zheng. 2025. **Maferw: query rewriting with multi-aspect feedbacks for retrieval-augmented large language models**. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'25/IAAI'25/EAAI'25*. AAAI Press.
- Zequ Wu, Yi Luan, Hannah Rashkin, David Reiter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. 2022. **CONQRR: Conversational query rewriting for retrieval with reinforcement learning**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10000–10014, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations (ICLR)*.

- Fanghua Ye, Meng Fang, Shenghui Li, and Emine Yilmaz. 2023. [Enhancing conversational search: Large language model-aided informative query rewriting](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5985–6006, Singapore. Association for Computational Linguistics.
- Chanwoong Yoon, Gangwoo Kim, Byeongguk Jeon, Sungdong Kim, Yohan Jo, and Jaewoo Kang. 2025. [Ask optimal questions: Aligning large language models with retriever’s preference in conversation](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5914–5936, Albuquerque, New Mexico. Association for Computational Linguistics.
- Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. [Few-shot generative conversational query rewriting](#). In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’20*, page 1933–1936, New York, NY, USA. Association for Computing Machinery.
- Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on research and development in information retrieval*, pages 829–838.
- Tianhua Zhang, Kun Li, Hongyin Luo, Xixin Wu, James R. Glass, and Helen M. Meng. 2024. [Adaptive query rewriting: Aligning rewriters through marginal probability of conversational answers](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13444–13461, Miami, Florida, USA. Association for Computational Linguistics.
- Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2025. [Swift: a scalable lightweight infrastructure for fine-tuning](#). In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI’25/IAAI’25/EAAI’25*. AAAI Press.
- Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.
- Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyang Luo. 2024. [LlamaFactory: Unified efficient fine-tuning of 100+ language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.
- Changtai Zhu, Siyin Wang, Ruijun Feng, Kai Song, and Xipeng Qiu. 2025. [ConvSearch-r1: Enhancing query reformulation for conversational search with reasoning via reinforcement learning](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26547–26564, Suzhou, China. Association for Computational Linguistics.

A Discussion

This section positions DVCQR with respect to three representative retriever-aligned CQR paradigms: (1) reasoning-first RL with a single rewrite (ConvSearch-R1), (2) joint sparse and dense alignment with a single rewrite (AdaCQR), and (3) test-time best-of- N selection with an outcome-supervised reward model (AdaRewriter). We also discuss how DVCQR relates to broader multi-candidate and multi-query practices, as well as to conversational dense retrieval (CDR) paradigms.

A.1 DVCQR vs. ConvSearch-R1: Dual-View Factorization vs. Single-Rewrite Optimization

ConvSearch-R1 (Zhu et al., 2025) uses retrieval-feedback RL to generate an explicit reasoning trace followed by a *single* rewrite, and optimizes it under a chosen retriever setting. This design is effective when the downstream retrieval stack is anchored to one retriever family or one fixed configuration, because the output space and the reward signal are relatively coherent.

DVCQR targets hybrid deployment where sparse and dense retrievers co-exist. In such settings, a single rewrite is often pulled between (1) **form tension**, where sparse retrieval prefers compact lexical anchors while dense retrieval prefers semantically explicit natural-language constraints, and (2) **objective interference**, where sparse and dense rank-based feedback can disagree and partially cancel when combined in one scalar objective (§ 4.5). DVCQR addresses both by decoupling: (1) **output space**: it generates a sparse-view keyword rewrite S and a dense-view natural-language rewrite D grounded by a shared reasoning trace T ; (2) **optimization**: it aligns S and D in successive RL stages, reducing direct reward cancellation and stabilizing training. In this sense, DVCQR can be viewed as extending the reasoning-first RL formulation of ConvSearch-R1 with an explicit dual-view factorization and stage-wise objective scheduling, aiming to better match heterogeneous retriever preferences within a single model.

A.2 DVCQR vs. AdaCQR: Explicit View-Specific Objectives vs. Unified Single Query with Fused Evaluation

AdaCQR (Lai et al., 2025b) aims to improve cross-retriever robustness while still producing a *single* unified rewrite. A key ingredient is a fusion-style metric that evaluates a candidate query across sparse and dense retrievers, inspired by reciprocal-rank fusion, and the method leverages multi-candidate generation and a contrastive objective to align the rewriter using relative candidate orderings induced by the fused metric.

DVCQR makes a different commitment explicit. Instead of asking one query to simultaneously satisfy two distinct query-form preferences, it factorizes the output into (S, D) and aligns each view with its corresponding retriever. This explicit factorization reduces preference entanglement in the output space, and the staged training further reduces interference in the optimization process. The trade-off is that DVCQR issues two short, view-specific queries at inference time rather than one unified query, but this can remain practical and efficient when each view is concise (Appendix I). Overall, AdaCQR and DVCQR represent two complementary strategies for dual-retriever robustness: AdaCQR emphasizes a single-query solution guided by fused cross-retriever evaluation, while DVCQR emphasizes a two-query solution that disentangles query forms and objectives.

A.3 DVCQR vs. AdaRewriter: Training-Time Alignment vs. Inference-Time Selection

AdaRewriter (Lai et al., 2025a) shifts robustness to inference by sampling N rewrite candidates and selecting best-of- N using an outcome-supervised reward model trained to predict retrieval outcomes under a fusion-based proxy. This strategy can hedge against occasional generation failures, but it increases inference computation roughly linearly with N due to repeated generation and scoring.

DVCQR instead learns a *single-pass* policy that directly produces two retriever-aligned outputs (S, D) through structured reasoning and stage-wise training. The two directions are orthogonal and potentially compatible. For example, one could apply best-of- N *within each view* (select among S candidates and among D candidates) as a compute-accuracy knob, without changing DVCQR’s core claim that sparse and dense prefer different query forms and benefit from decoupled objectives.

Relation to multi-candidate and multi-query paradigms. Beyond retriever-aligned training, another common way to improve robustness is test-time scaling, e.g., generating multiple candidates and fusing or selecting them (such as CHIQ), or issuing multiple rewrites/queries via repeated LLM calls (e.g., LLM4CS-style multi-query variants). These approaches are largely orthogonal to DVCQR. DVCQR focuses on *training-time* decoupling of query forms and objectives (dual views + stage-wise alignment) to obtain a strong *single-pass* policy, whereas multi-candidate/multi-query methods primarily trade extra inference compute (more generations and often more scoring/retrieval calls) for robustness. Multi-candidate selection and additional query diversification can be seen as optional extensions on top of DVCQR, and we view them as complementary rather than necessary for the main scope of this work.

A.4 DVCQR vs. Conversational Dense Retrieval (CDR): Rewrite-Centric vs. Retriever-Centric Adaptation

Conversational Dense Retrieval (CDR) paradigms are largely orthogonal to Conversational Query Rewriting (CQR). CDR methods typically adapt the retriever itself to encode the current query together with the dialogue history, whereas DVCQR optimizes rewrites that can be paired with off-the-shelf retrievers. In this sense, CDR is *retriever-centric*, while DVCQR is *rewrite-centric*. The comparison is therefore not strictly apples-to-apples, but it still provides a useful reference for understanding the effectiveness of our dual-view formulation under a standard dense retriever setting.

Using an off-the-shelf ANCE retriever, we compare DVCQR against several representative CDR methods on TopiOCQA and QReCC in terms of MRR. The results are shown in Table 4. DVCQR achieves the best MRR on both benchmarks, outperforming prior CDR baselines by a clear margin. This result suggests that improving the query side through structured dual-view rewriting can remain highly competitive even without retriever-side conversational adaptation.

Beyond effectiveness, DVCQR retains two properties that distinguish it from retriever-centric CDR approaches. First, the two rewrites remain interpretable: the sparse view exposes lexical anchors preferred by sparse retrieval, while the dense view makes semantic constraints explicit for dense retrieval. Second, DVCQR is compatible with

Method	TopiOCQA	QReCC
Conv-ANCE	22.9	47.1
InstructoR-ANCE	25.3	43.5
Conv-SPLADE	30.7	50.0
LeCoRE	32.0	51.1
ChatRetriever	38.1	36.5
ContextualRetriever	42.2	36.8
DVCQR	53.5	52.3

Table 4: Comparison with representative conversational dense retrieval (CDR) paradigms using an off-the-shelf ANCE retriever. Evaluation metric is MRR.

both sparse and dense retrievers, rather than being tied to a specially adapted conversational retriever. More importantly, these two directions are complementary rather than exclusive. In future work, DVCQR could be combined with retriever-centric CDR methods, potentially yielding gains from both query-side dual-view specialization and retriever-side conversational modeling.

B SFT Data Collection

We use DeepSeek-V3.2-Exp (DeepSeek-AI, 2025) as the teacher model and distill data from the TopiOCQA and QReCC training sets using a pre-designed prompt (As shown in Table 12). We retain only the instances that satisfy the format requirements and rank the gold passage at 1 under both the sparse and dense retrievers as the SFT training data. Detailed statistics are reported in Table 5.

	TopiOCQA	QReCC
Sample Nums	12,987	8,985

Table 5: Statistics of the distilled data used for SFT.

C Format Reward Design

In addition to the rewards proposed in the main text, we further introduce a format reward R_f as a global pre-check for all reward computations. If the output format is incorrect, we set $R_f = 0$ and skip all subsequent reward evaluations, i.e., the total reward is set to 0. If the format is correct, we set $R_f = 1$ and compute the downstream rewards. The full reward R_{full} is defined in Eq. 8.

We implement R_f as a hard gate in our reward pipeline. In addition, for logging purposes, we register an auxiliary format-check reward in MS-Swift with a near-zero weight (e.g., 10^{-6}) so that the framework can report format compliance statistics

as a separate metric during training. This logging-only term is used for monitoring and has negligible impact on optimization.

$$R_{full} = \begin{cases} 0, & R_f = 0, \\ R^{(s)}, & R_f = 1, \text{ Stage II} \\ R^{(d)}, & R_f = 1, \text{ Stage III} \end{cases} \quad (8)$$

D Experimental Details

D.1 Datasets Details

TopiOCQA is an open-domain conversational QA dataset characterized by frequent topic switches. QReCC is an end-to-end open-domain conversational QA dataset built from TREC CASt, QuAC, and Google Natural Questions. CASt-19 is constructed using the TREC CAR paragraph collection and the MS MARCO passage collection, and CASt-20 follows a similar setup. Table 6 reports detailed statistics for the four datasets. All datasets used in this paper are supported for academic research.

Dataset	Split	#Conv.	#Turns(Qry.)	#Collection
TopiOCQA	train	3,509	45,450	25M
	test	205	2,514	
QReCC	train	10,823	29,596	54M
	test	2,775	8,124	
CASt-19	test	50	479	38M
CASt-20	test	25	208	

Table 6: Datasets statistics.

D.2 Evaluation Metrics Details

For evaluation metrics, we report MRR, NDCG@3, and Recall@K, where MRR reflects the rank of the first relevant result, NDCG@3 measures top-3 relevance with position discounting, and Recall@K is the proportions of relevant documents within the top K results.

D.3 Baselines Details

We have provided a more detailed introduction to the following baseline methods:

T5QR (Lin et al., 2020): A vanilla supervised baseline that fine-tunes a T5-base sequence-to-sequence model to rewrite context-dependent conversational queries into standalone rewrites for retrieval.

InfoCQR (Ye et al., 2023): A prompting-based method that uses an LLM to produce more informative rewrites (e.g., via a rewrite-then-edit style

prompting strategy) to improve downstream retrieval quality.

LLM4CS(Mao et al., 2023b): A prompting framework that generates multiple rewrites with an LLM and aggregates them (e.g., at the embedding level for dense retrieval, with an adapted strategy for sparse retrieval).

CHIQ-Fusion(Mo et al., 2024): A history-enhanced conversational query rewriting approach that combines complementary CHIQ variants and fuses their ranked results to improve retrieval effectiveness.

CONQRR(Wu et al., 2022): A retriever-aligned CQR baseline that directly optimizes retrieval effectiveness by training a T5-based rewriter with RL using retriever feedback.

EDIRCS(Mao et al., 2023a): An editing-style rewriting model that generates queries efficiently by combining non-autoregressive text selection with autoregressive token generation on a fine-tuned T5 backbone.

ConvGQR(Mo et al., 2023): A generative rewriting framework that boosts retrieval by coupling a query rewriting model with an additional query expansion component (e.g., generating extra keywords/answer-like cues).

IterCQR(Jang et al., 2024): A retrieval-guided iterative rewriting method that reduces reliance on human rewrites by repeatedly improving a T5-based rewriter using retrieval-driven objectives.

AdaQR(Zhang et al., 2024): A retriever-aligned query rewriting approach that aligns the rewriter using answer-related signals (via marginal probability of conversational answers) to better match downstream retrieval.

RETPO(Yoon et al., 2025): A preference-optimization baseline that generates multiple rewrites, derives pairwise preferences from retriever feedback, and trains an LLM rewriter with direct preference optimization.

ConvSearch-R1(Zhu et al., 2025): A reasoning-first RL framework that generates intermediate reasoning before producing the final rewrite, and learns from retriever ranking feedback via reward shaping.

AdaCQR(Lai et al., 2025b): A two-stage alignment framework that jointly targets sparse and dense retrieval by aligning the rewriting model (rewriter) to retrievers from term-matching and semantic perspectives, including contrastive objectives.

AdaRewriter(Lai et al., 2025a): A prompting-

based rewriting framework that performs test-time adaptation by learning a lightweight outcome-supervised reward model to score candidates and select the best-of-N rewrite during inference.

D.4 Implementation Details

For model selection, the backbone versions we use are Qwen2.5-3B (Qwen/Qwen2.5-3B-Instruct) and Qwen3-4B (Qwen/Qwen3-4B-Instruct-2507). We use ANCE (sentence-transformers/msmarco-roberta-base-ance-firstp) as the dense retriever.

For training, during the SFT stage, we use a learning rate of 1×10^{-5} , a maximum sequence length of 3072, and 2 training epochs. In the two-stage RL, we set the maximum prompt length to 2048 and the maximum response length to 1024, use a learning rate of 3×10^{-6} , set β to 0.001, and sample $num_generations = 6$ responses with $temperature = 0.7$, $top_p = 0.8$, and $top_k = 20$. We train for 4 epochs on TopiOCQA and 5 epochs on QReCC, respectively. In addition, the reward weights in both RL stages are set to $w_1 = 0.9$, and $w_2 = 0.1$. In Stage III, ANCE is used as the encoder to compute the semantic similarity reward. All experiments ran on $4 \times$ NVIDIA RTX PRO 6000 Blackwell GPUs.

For evaluation, following prior work (Zhang et al., 2024), we set BM25 hyperparameters to $k_1 = 0.9$ and $b = 0.4$ for TopiOCQA, and $k_1 = 0.82$ and $b = 0.68$ for QReCC.

E Cross-Retriever Generalization

Table 7 reports cross-retriever generalization results to examine whether the rewriter overfits to a specific retriever configuration. Keeping the training and inference pipeline unchanged, we replace the sparse retriever BM25 with DPH and the dense retriever ANCE with BGE (BAAI/bge-small-en-v1.5). The results show that DVCQR consistently outperforms ConvSearch-R1 across all four retrieval settings, and its advantage remains substantial after swapping retrievers. For example, on DPH, DVCQR (Qwen2.5-3B) improves over ConvSearch-R1 by +2.8 MRR and +3.7 R@10; on BGE, it gains +2.6 MRR and +1.9 R@100. The improvements are even larger with the stronger backbone Qwen3-4B. These results suggest that the dual-view outputs explicitly disentangle “lexical anchors” and “semantic constraints,” enabling S and D to better match the query preferences of different retriever families, thereby improving

Type	Method	Backbone	TopiOCQA			
			MRR	NDCG@3	R@10	R@100
Sparse (BM25)	ConvSearch-R1	Qwen2.5-3B	35.2	33.5	57.8	79.9
	DVCQR	Qwen2.5-3B	37.4	35.7	60.3	82.8
	DVCQR	Qwen3-4B	39.5	38.0	63.7	84.1
Sparse (DPH)	ConvSearch-R1	Qwen2.5-3B	34.0	32.6	55.9	78.2
	DVCQR	Qwen2.5-3B	36.8	34.9	59.6	81.7
	DVCQR	Qwen3-4B	39.0	37.6	63.4	83.5
Dense (ANCE)	ConvSearch-R1	Qwen2.5-3B	51.4	51.3	72.0	85.7
	DVCQR	Qwen2.5-3B	52.5	52.1	72.7	86.6
	DVCQR	Qwen3-4B	53.5	52.9	73.4	87.8
Dense (BGE)	ConvSearch-R1	Qwen2.5-3B	49.4	48.9	73.0	89.2
	DVCQR	Qwen2.5-3B	52.0	50.8	73.9	91.1
	DVCQR	Qwen3-4B	53.2	52.2	74.8	92.7

Table 7: Cross-retriever generalization results. DPH takes S as input and BGE takes D . The BM25 and ANCE results are the same as those in the main experiments.

Type	Reward	TopiOCQA			
		MRR	NDCG@3	R@10	R@100
Sparse (BM25)	$R^{(s)}$	56.2	57.6	76.6	88.1
	$R_{\text{rank}}(r_s)$	55.7	56.9	76.6	88.1
Dense (ANCE)	$R^{(d)}$	61.5	61.2	77.4	90.5
	$R_{\text{rank}}(r_d)$	60.8	60.4	77.4	90.5

Table 8: Reward analysis on TopiOCQA. $R^{(s)}$ denotes the total reward in Stage II, while $R_{\text{rank}}(r_s)$ is the Stage II BM25 rank-based main reward, i.e., the total reward without the keyword-coverage reward R_k . $R^{(d)}$ denotes the total reward in Stage III, while $R_{\text{rank}}(r_d)$ is the Stage III ANCE rank-based main reward, i.e., the total reward without the semantic similarity reward R_s .

cross-retriever robustness and transferability.

F Reward Analysis

Due to limited compute resources, we did not conduct a strict reward ablation study; instead, we performed a lightweight inference-based verification. Using DVCQR (Qwen3-4B), we randomly sampled 165 examples from the TopiOCQA test set, generated 6 candidate outputs per input, and selected candidates according to two criteria. The first uses the total reward (Stage II: $R^{(s)} = 0.9 \cdot R_{\text{rank}}(r_s) + 0.1 \cdot R_k$; Stage III: $R^{(d)} = 0.9 \cdot R_{\text{rank}}(r_d) + 0.1 \cdot R_s$). The second uses only the rank-based main reward (Stage II: $R_{\text{rank}}(r_s)$; Stage III: $R_{\text{rank}}(r_d)$). As shown in Table 8, both the main and auxiliary rewards are beneficial: the total reward yields consistent but modest gains on both BM25 and ANCE. Specifically, BM25 MRR/NDCG@3 increase by +0.5/+0.7, and ANCE MRR/NDCG@3 increase by +0.7/+0.8, while R@10 and R@100 remain essentially un-

changed. This suggests that once the final model already has a strong ability to push the gold passage into the top-100, the rank-based main reward accounts for most of the gains, whereas the auxiliary terms act more like a fine-grained preference constraint. Meanwhile, during training, the auxiliary reward can provide a subtle learning signal when the main ranking reward is zero (i.e., rank > 100), making training more stable.

G Diagnosing Sparse–Dense Preference Conflict in JointRL

We conduct jointRL under the single-rewrite setting, where the ranking rewards from the sparse and dense retrievers are simultaneously used for policy optimization. Let $R_s(t)$ and $R_d(t)$ denote the sparse and dense ranking rewards at logging step t , respectively, and let $R(t)$ be the scalarized total reward used for optimization:

$$R(t) = \lambda R_s(t) + (1 - \lambda) R_d(t) + R_f, \quad (9)$$

where $\lambda \in (0, 1)$ is a mixing weight and R_f denotes format reward.

Local conflict and cancellation metric We focus on local changes between two adjacent logging points and define

$$\begin{aligned}\Delta R_s(t) &= R_s(t) - R_s(t^-), \\ \Delta R_d(t) &= R_d(t) - R_d(t^-), \\ \Delta R(t) &= R(t) - R(t^-),\end{aligned}\quad (10)$$

where t^- denotes the most recent logging point before t . If $\Delta R_s(t) \cdot \Delta R_d(t) < 0$, the sparse and dense rewards move in opposite directions within this local interval, indicating a potential preference conflict. To quantify the extent to which these opposite changes are canceled out in the scalarized objective, we define the cancellation ratio as

$$\rho(t) = \frac{|\Delta R(t)|}{|\Delta R_s(t)| + |\Delta R_d(t)| + \epsilon}, \quad (11)$$

where ϵ is a numerical-stability constant. Intuitively, when $|\Delta R_s(t)|$ and $|\Delta R_d(t)|$ are large but cancel each other in $R(t)$, $|\Delta R(t)|$ becomes relatively smaller, leading to a lower value of $\rho(t)$.

Entanglement-cancellation intervals We merge consecutive logging points that simultaneously satisfy $\Delta R_s(t) \cdot \Delta R_d(t) < 0$ and $\rho(t)$ being below a threshold into an entanglement-cancellation interval, which characterizes the sustained occurrence of conflicts along the training trajectory.

H JointRL vs. Stage-wise RL

In Table 9, we compare the offline retrieval performance of four training strategies on TopiOCQA. Under Single-Rewrite JointRL, which applies a joint scalarized reward from both retrievers to a single-rewrite output, the MRR of BM25/ANCE is only 21.1/33.1, substantially worse than ConvSearch-R1 (35.2/51.4), which optimizes a single-rewrite model using a scalar reward from a single retriever. This suggests that, when sparse and dense preferences are forced to share the same output space and the same scalar objective, they are more likely to constrain each other, making it difficult to simultaneously form stable lexical anchors and sufficient semantic constraints. After switching the output to a dual-view format, Dual-View-Rewrite JointRL improves markedly to 34.4/47.0 MRR on BM25/ANCE, indicating that decoupling the output form into two views can

indeed mitigate part of the conflict; however, it still lags clearly behind the stage-wise DVCQR (Qwen2.5-3B: 37.4/52.5), with a more pronounced gap on dense-side top-ranked metrics, implying that even if the outputs are decoupled, as long as the two signals are still compressed into a single scalar reward and optimized synchronously within the same stage, training can still suffer from objective interference and signal cancellation. Ultimately, DVCQR’s stage-wise RL achieves the best results under both BM25 and ANCE, and further enlarges the margin with the stronger Qwen3-4B backbone (39.5/53.5), more directly supporting our conclusion that decoupling is needed at both the output space and the optimization process, and that stage-wise reinforcement is crucial to reliably unlock the full potential of the dual-view structure.

I Efficiency Analysis

We conduct efficiency experiments on the TopiOCQA test set. Using the Qwen2.5-3B tokenizer, we count the number of tokens in the actual query text issued to each retriever, and we measure the wall-clock latency of output generation, BM25 retrieval, and ANCE retrieval. Latency is measured for both methods under the same implementation and hardware environment. We additionally measure output latency, defined as the wall-clock time from generation start to output completion. A key difference is that ConvSearch-R1 generates a single rewrite per query, which is then issued to both BM25 and ANCE, whereas DVCQR generates two view-specific rewrites and issues the sparse-view rewrite to BM25 and the dense-view rewrite to ANCE. As shown in Table 10, ConvSearch-R1 produces substantially longer rewrites (267.3 tokens on average), resulting in 671,788 tokens issued to BM25 and the same 671,788 tokens issued again to ANCE over the test set. In contrast, DVCQR generates a compact sparse-view rewrite (70.9 tokens on average) and a concise dense-view rewrite (87.5 tokens on average), reducing the issued token budget by 73.5% for BM25 and 67.3% for ANCE. Considering both retrievers together (i.e., total issued tokens across the two retrieval calls per query), DVCQR reduces the overall token budget from 1,343,576 to 397,832 (70.4% fewer tokens).

In addition to the reduction in rewrite length, DVCQR also substantially improves generation efficiency. As shown in Table 10, the average output latency drops from 966.7 ms to 325.6 ms per query,

Type	Method	Backbone	TopiOCQA			
			MRR	NDCG@3	R@10	R@100
Sparse (BM25)	Single-Rewrite JointRL	Qwen2.5-3B	21.1	19.0	38.6	63.9
	Dual-View-Rewrite JointRL	Qwen2.5-3B	34.4	32.3	57.0	80.6
	ConvSearch-R1	Qwen2.5-3B	35.2	33.5	57.8	79.9
	DVCQR	Qwen2.5-3B	37.4	35.7	60.3	82.8
	DVCQR	Qwen3-4B	39.5	38.0	63.7	84.1
	Dense (ANCE)	Single-Rewrite JointRL	Qwen2.5-3B	33.1	31.7	52.3
Dual-View-Rewrite JointRL		Qwen2.5-3B	47.0	46.2	67.2	83.9
ConvSearch-R1		Qwen2.5-3B	51.4	51.3	72.0	85.7
DVCQR		Qwen2.5-3B	52.5	52.1	72.7	86.6
DVCQR		Qwen3-4B	53.5	52.9	73.4	87.8

Table 9: Comparison results between JointRL and Stage-wise RL. ConvSearch-R1 produces a single rewrite and uses RL to align it with the preference of a single retriever. DVCQR generates dual-view rewrites and employs stage-wise RL to align with the preferences of two types of retrievers.

Method	Rewrite Tokens				Output Latency		BM25 Latency		ANCE Latency	
	Sparse Total	Total Sparse Avg	Dense Total	Total Dense Avg	Total (s)	Avg (ms)	Total (s)	Avg (ms)	Total (s)	Avg (ms)
ConvSearch-R1	671,788	267.3	671,788	267.3	2,430.1	966.7	6,329.8	2,517.9	1,481.0	589.2
DVCQR	178,002	70.9	219,830	87.5	818.5	325.6	1,772.4	705.0	1,436.0	571.3

Table 10: Efficiency comparison on the TopiOCQA test set. Rewrite length is measured as the number of tokens in the retriever-issued query text, counted by the Qwen2.5-3B tokenizer (reported as total tokens over the test set and average tokens per query). Output latency is measured as wall-clock time from generation start to output completion, reported as total seconds over the test set and average milliseconds per query. Retrieval latency is measured as wall-clock time for running BM25 or ANCE retrieval, reported as total seconds over the test set and average milliseconds per query.

corresponding to an approximately $2.97\times$ speedup, while the total output latency over the test set decreases from 2430.1 s to 818.5 s. Combined with the rewrite length statistics above, these results indicate that although DVCQR additionally produces reasoning and two specialized rewrites, its single-pass generation of dual-view rewrites remains significantly more efficient than ConvSearch-R1.

This reduction translates into a large latency improvement for sparse retrieval. BM25 latency drops from 2,517.9 ms to 705.0 ms per query ($3.57\times$ faster), which is consistent with the fact that sparse retrieval must process and score more term-level matching signals when the query contains many clauses or expanded terms. In contrast, ANCE latency changes only marginally (589.2 ms vs. 571.3 ms), suggesting that dense retrieval time is largely dominated by query encoding and nearest-neighbor search over dense vectors, where moderate differences in rewrite length have a smaller impact.

Dual-View Structured Reasoning Prompt

Step1. Generate Sparse Rewrite

1. Topic, Target and Relation Identification (highest priority).

Carefully read the Context and Query. Clarify references and normalize entity names. Determine one clear main entity and one clear intent (subject plus relation, for example definition, date, place, function, cast, economy, comparison). If unsure, choose the most defensible interpretation explicitly supported by the inputs; do not invent facts. Do not change the main entity or the main aspect.

2. Answer-aware Term Harvest and Anchor Selection.

Using the available information (Context, Query, and your own world knowledge) as hints, privately form an answer to the Query. From that answer and the inputs, collect high-information anchors:

- Canonical entity names and any short disambiguators (for example film, island, hormone).
- Section-relevant words that indicate the correct aspect (for example release, introduction, economy, storm tides, whaling, cast).
- Distinctive nouns, numerals, dates, locations, titles that are clearly supported by your understanding of the topic and the conversation.
- Domain or type words (for example war film, island municipality, hormone, marine organisms).

For these anchors, you may add useful Sparse-oriented variants: common aliases or abbreviations, US or UK spellings, and hyphen or no-hyphen forms (for example covid-19 covid 19). If index stemming is unknown, you may include singular or plural or derivational variants (for example policy policies, regulation regulations). Do NOT introduce a different main entity or unsupported key facts.

3. Sparse Keyword Construction and Hygiene.

Compose a compact bag-of-words keyword string using only the selected anchors:

- Start from the canonical main entity name or names.
- Add relation or type words and critical constraints (year, location, domain, aspect) that are needed to retrieve the correct passage.
- Remove function words and fillers.

Constraints:

- Deduplicate tokens; prefer lowercase except for acronyms or proper names.
- Aim for 8 to 15 information tokens (at most about 20).
- Allowed characters: letters, digits, spaces, hyphen -.
- Disallowed: quotes, parentheses, commas, slashes, colons, boolean operators such as AND OR NOT, and any field syntax.
- Do NOT mix unrelated entities or broaden to a different topic.

4. Final Text for Sparse.

Output the sparse-retriever-style keyword string enclosed within <sparse> and </sparse>.

Step2. Generate Dense Rewrite

1. Natural-language Rewrite Planning.

Using the same main entity and intent determined in Step1, and the internal answer formed there, decide how to express the Query as one self-contained natural-language text. The Dense rewrite should:

- Be a fluent English question or statement with 1 to 3 sentences.
- Explicitly mention the canonical entity name (and at most one common abbreviation).
- Clearly state the relation and include any essential constraints such as time, location, domain, type, section aspect.
- Stay focused on the same semantic center as the sparse rewrite; do not broaden to unrelated aspects or topics.

2. Dense Rewrite Construction.

Write the Dense rewrite as a short, well-formed snippet that a user might naturally type into a search engine:

- Avoid unresolved pronouns like it, he, this place; replace them with explicit entities or descriptions.
- Keep the rewrite concise and information-dense, without extra explanation or multiple rephrasings.
- Use only plain English text (no bullet points, no lists, no special syntax).
- Keep the length reasonable (for example, roughly 1 to 3 sentences, not a long multi-paragraph essay).

3. Rewrite Answer Generation.

Based on the same understanding and the available information, produce a concise final answer that is consistent with the dense rewrite:

- The answer should be factually plausible and aligned with the Context and your world knowledge; do not contradict the earlier reasoning or change the main entity.
- You may use general world knowledge, but you must not introduce obviously incorrect or contradictory details.
- The answer should be less than 3 sentences.

4. Final Text for Dense.

Concatenate the dense rewrite and the answer you generated into a single coherent text, and enclose this text within <dense> and </dense>.

Step3. Wrap your reasoning inside <thinking>

Write your full reasoning process in English and wrap it inside <thinking> and </thinking>.

Step4. If everything is correct, you can output it.

Table 11: Dual-view structured reasoning prompt.

Prompt for Data Collection

You are a conversational query rewriter for the English dataset, optimized for a Sparse retriever and a Dense retriever. You are given a Query and its Context. Your task is to rewrite the Query to make it suitable for the two retrievers. A suitable rewrite means that the rewrite should be sufficient for the retriever to rank the single most relevant passage in the top position without relying on the Context. You MUST give a sparse rewrite for Sparse retrieval and a dense rewrite for Dense retrieval. All outputs in <thinking>, <sparse>, and <dense> MUST be in English.

Your output MUST be:

<thinking>your reasoning here</thinking>

<sparse>your sparse rewrite here</sparse>

<dense>your dense rewrite here</dense>

and NOTHING ELSE.

The only available information is the query, the context, and your world knowledge.

You MUST keep the main entity and section/aspect of the question consistent with the inputs.

1. Do NOT switch to a different main entity.
2. Do NOT drift from a specific aspect/section to a very broad, generic topic.
3. Do NOT invent clearly implausible awards, festivals, events, places or dates. If you are uncertain, prefer safer, more conservative wording rather than fabricating specific facts.

Here is one example for your reference.

####Example Begin####

{example}

####Example End####

Now it's your task, and refer to the following thinking steps to complete the task.

####Context Begin####

{context}

####Context End####

Query: {query}

Let's think step-by-step (your reasoning).

{dual-view_structured_reasoning_prompt}

Table 12: Prompt for data collection.

Prompt for Training and Inference

You are a conversational query rewriter for the English dataset, optimized for a Sparse retriever and a Dense retriever. You are given a Query and its Context. Your task is to rewrite the Query to make it suitable for the two retrievers. A suitable rewrite means that the rewrite should be sufficient for the retriever to rank the single most relevant passage in the top position without relying on the Context. You MUST give a sparse rewrite for Sparse retrieval and a dense rewrite for Dense retrieval. All outputs in <thinking>, <sparse>, and <dense> MUST be in English.

Your output MUST be:

<thinking>your reasoning here</thinking>

<sparse>your sparse rewrite here</sparse>

<dense>your dense rewrite here</dense>

and NOTHING ELSE.

The only available information is the query, the context, and your world knowledge.

You MUST keep the main entity and section/aspect of the question consistent with the inputs.

1. Do NOT switch to a different main entity.
2. Do NOT drift from a specific aspect/section to a very broad, generic topic.
3. Do NOT invent clearly implausible awards, festivals, events, places or dates. If you are uncertain, prefer safer, more conservative wording rather than fabricating specific facts.

Now it's your task, and refer to the following thinking steps to complete the task.

####Context Begin####

{context}

####Context End####

Query: {query}

Let's think step-by-step (your reasoning).

{dual-view_structured_reasoning_prompt}

Table 13: Prompt for training and inference.

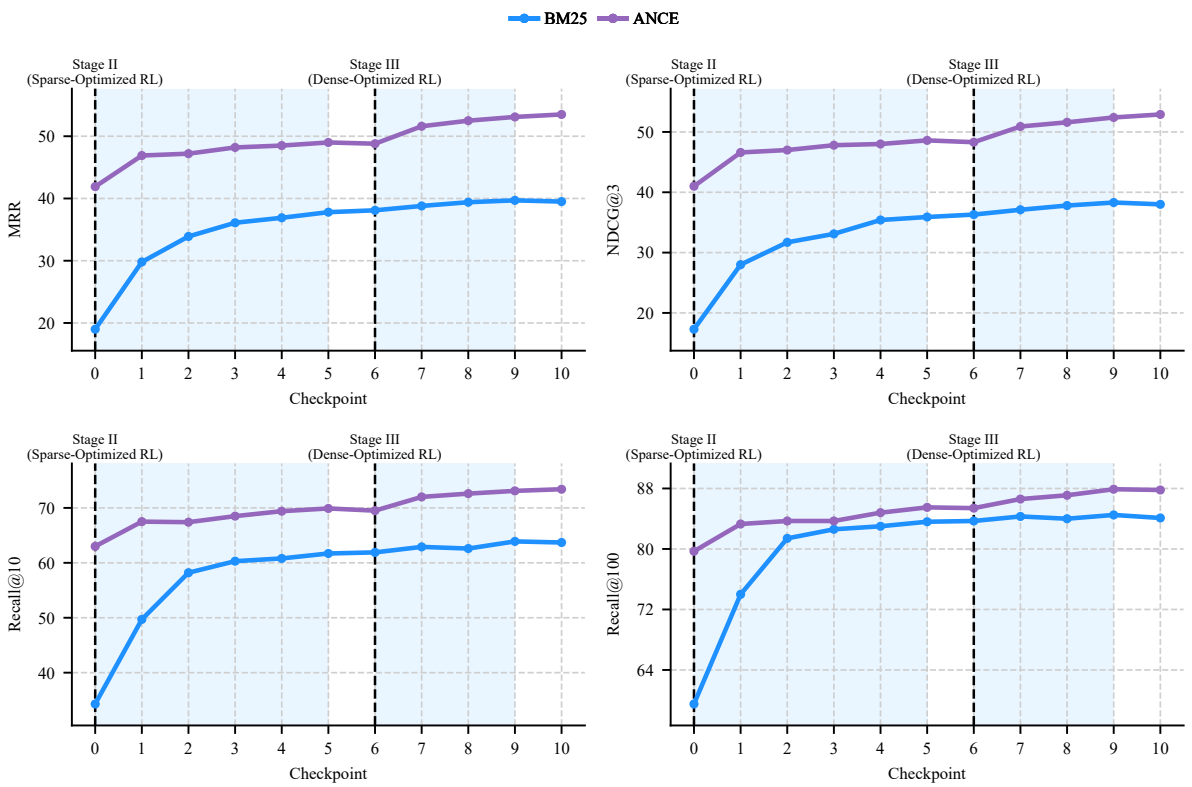


Figure 5: Checkpoint training trajectories on TopiOCQA. Each checkpoint is evaluated on both BM25 and ANCE retrieval metrics; dashed lines indicate the starting points of Stage II and Stage III, and the blue shaded region highlights the window where both sparse and dense metrics rise simultaneously.

Conversation

Q1: Who does melissa george play in greys anatomy?

A1: Sadie Harris.

Q2: Where was she born?

A2: Perth, Western Australia.

Query: Which was her debut movie?

ConvSearch-R1

<thinking> The user is asking about the first movie that the actress Melissa George's character, Sadie Harris, appeared in. The provided context focuses on Sadie Harris' birthplace and does not contain any information about her debut movie or any of her filmography. However, **since the context mentions Sadie Harris, we can infer that the query is likely about an actress named Sadie Harris.** From previous queries, it's clear that Sadie Harris is a person, and we should focus on finding a movie debut for Sadie Harris. Additionally, Sadie Harris is known to be an Australian actress, and her early film appearances are typically well-documented. However, we would need specific context about her to determine her first movie role accurately. **</thinking>**
<rewrite> Which was Sadie Harris' debut movie? Sadie Harris is an Australian actress born in Perth, Western Australia. While the provided context does not include information about her first film, some of her early appearances include the 2003 Australian comedy film "The Dilemma" and her television debut. Her first notable acting role was in the 2001 TV film "Home and Away" and the 2003 film "The Dilemma." Sadie Harris made her feature film debut with "The Dilemma" in 2003. She was also in the 2005 film "Shine a Light" and appeared in the 2006 film "My Big Fat Greek Wedding 2." Her television debut was on the Australian television drama series "Home and Away," which she started in 2001. Sadie Harris has also appeared in several other movies such as the 2018 film "The Man Who Cried" and the 2016 film "Goodbye Christopher Robin," where she starred alongside Helena Bonham Carter. Sadie Harris is also known for her role in the popular television series "Glee," where she played the character Sadie Cohen. Sadie Harris has continued to build her acting career with various films and television shows since her early appearances. **</rewrite>** (BM25: Not Found, ANCE: Not Found)

DVCQR

<thinking> The user asks "which was her debut movie?" after previous questions about Melissa George's role in Grey's Anatomy and her birthplace. **The pronoun "her" clearly refers to Melissa George. The intent is to find her first feature film appearance.** From the available information, Melissa George made her feature film debut with a supporting role in 1998 with Dark City. She is an Australian and American actress known for this early film role before gaining wider recognition in television and later films. **For the sparse rewrite,** I need keywords like "melissa george", "debut movie", "first film", "melissa george debut", "dark city", "1998", "melissa george film debut", "australian actress debut", "early career", "feature film debut", "melissa george early life", "acting debut", "british australian actress", "young people's pictures". **For the dense rewrite,** I'll create a natural question about her debut film and provide the answer with relevant details about her early career and the specific debut film. **</thinking>**
<sparse> melissa george debut movie first film melissa george debut dark city 1998 melissa george film debut early career melissa george acting debut australian actress debut feature film debut melissa george early life british australian actress young people's pictures debut film melissa george early films **</sparse>** (BM25: Rank 1)
<dense> What was Melissa George's debut movie? Melissa George made her feature film debut in Dark City (1998), directed by Alex Proyas. She later gained wider recognition through television roles (including Grey's Anatomy) and film work such as The Amityville Horror (2005). **</dense>** (ANCE: Rank 1)

Table 14: Case study on TopiOCQA-dev_37_3. Text highlighted in red indicates critical errors, Text highlighted in cyan indicates critical correctness.

Conversation

Q1: When did the movie "the thing" come out?

A1: 1982.

Query: What is the movie about?

ConvSearch-R1

<thinking> The user is asking about the plot or summary of the movie "The Thing". The context provides a release year for the movie but does not contain a detailed description of its plot or summary. From previous information, it is known that the movie is about "The Thing," which is likely referring to the 1982 sci-fi horror film, and it is necessary to find out what the movie is about based on the context provided. Additionally, the plot summary would include details about the movie's main characters and the central conflict. </thinking>

<rewrite> **What is the movie about?** "The Thing" is a 1982 science fiction horror film directed by John Carpenter. The movie tells the story of a group of explorers who land on an alien planet and encounter an alien organism that can disguise itself as a human. The organism's goal is to infect and take over the human body, leading to a series of suspenseful and terrifying events. The plot revolves around a team of scientists who investigate a mysterious creature that causes unexplained deaths and mutations. The movie features a cast of characters who struggle against the alien menace and the challenges of survival in a hostile environment. The film is set in Antarctica, where the crew of the US Coast Guard ship USS Andromeda discovers the alien organism while searching for a crashed alien spacecraft. It stars Kurt Russell, Willem Dafoe, and John Savage, among others. The movie is known for its blend of science fiction, horror, and psychological tension, making it a classic in the genre. The main characters, including the protagonist Chris Kelvin, are forced to confront the unknown while trying to survive the alien threat. "The Thing" is a story of human resilience, the dangers of alien life, and the struggle between living and being consumed by the unknown. The movie's plot is centered around the idea of an alien organism that can mimic human appearance and senses, leading to a series of murders and the discovery of a secret that could change the course of human history. </rewrite>

(BM25: Not Found) (ANCE: Rank 1)

DVCQR

<thinking> The user asks "what is the movie the thing about?" after learning it was released in 1982. **This refers to the American science fiction horror film "The Thing."** **The query seeks a plot summary.** The film stars Kurt Russell, directed by John Carpenter, and follows scientists in Antarctica who discover a mysterious alien organism that can imitate human appearance. **For the sparse rewrite,** I need keywords like "the thing 1982 film", "the thing movie plot", "the thing about", "american science fiction horror film", "john carpenter", "kurt russell", "the thing plot summary", "antarctic alien invasion", "thing movie storyline", "science fiction horror film 1982", "anthropomorphic alien organism", "parasitic species", "the thing film plot". **For the dense rewrite,** I'll create a natural question about the film's plot and provide a concise summary based on the available information. </thinking>

<sparse> **the thing 1982 film the thing movie plot** the thing about american science fiction horror film the thing plot summary john carpenter kurt russell the thing 1982 antarctic alien invasion thing movie storyline science fiction horror film 1982 anthropomorphic alien organism parasitic species the thing film plot </sparse> (BM25: Rank 1)

<dense> **What is the 1982 film The Thing about?** The Thing is a 1982 American science fiction horror film directed by John Carpenter starring Kurt Russell. It follows a group of researchers in Antarctica who encounter a parasitic alien organism that can mimic human appearance, leading to paranoia and distrust as they try to survive. </dense> (ANCE: Rank 1)

Table 15: Case study on TopiOCQA-dev_43_2. Text highlighted in red indicates critical errors, Text highlighted in cyan indicates critical correctness.