ConvSearch-R1: Enhancing Query Reformulation for Conversational Search with Reasoning via Reinforcement Learning

Changtai Zhu[1*], Siyin Wang[1], Ruijun Feng[3], Kai Song[2], Xipeng Qiu[1†]

[1]Fudan University, [2]ByteDance Inc, [3]University of New South Wales

ctzhu23@m.fudan.edu.cn, {siyinwang20, xpqiu}@fudan.edu.cn, songkai.sk.batman@bytedance.com, ruijun.feng@unsw.edu.au

Abstract

Conversational search systems require effective handling of context-dependent queries that often contain ambiguity, omission, and coreference. Conversational Query Reformulation (CQR) addresses this challenge by transforming these queries into self-contained forms suitable for off-the-shelf retrievers. However, existing CQR approaches suffer from two critical constraints: high dependency on costly external supervision from human annotations or large language models, and insufficient alignment between the rewriting model and downstream retrievers. We present ConvSearch-R1, the first self-driven framework that completely eliminates dependency on external rewrite supervision by leveraging reinforcement learning to optimize reformulation directly through retrieval signals. Our novel two-stage approach combines Self-Driven Policy Warm-Up to address the cold-start problem through retrieval-guided self-distillation, followed by Retrieval-Guided Reinforcement Learning with a specially designed rank-incentive reward shaping mechanism that addresses the sparsity issue in conventional retrieval metrics. Extensive experiments on TopiOCQA and QReCC datasets demonstrate that ConvSearch-R1 significantly outperforms previous state-of-the-art methods, achieving over 10% improvement on the challenging TopiOCQA dataset while using smaller 3B parameter models without any external supervision.

1 Introduction

Conversational search aims to fulfill users' information needs through multi-turn interactions, unlike traditional information retrieval systems that only consider single-turn, keyword-based queries (Joshi et al., 2017; Kwiatkowski et al., 2019). In these multi-turn scenarios, conversational queries often contain ambiguity, omission, and coreference,

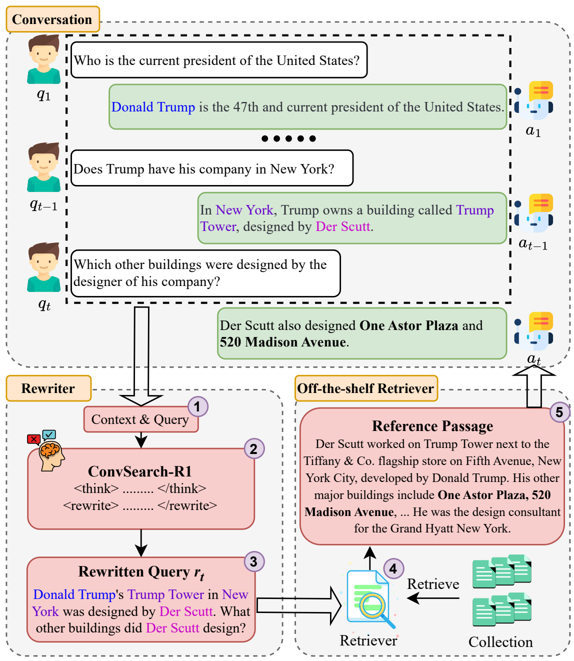


Figure 1: Illustration of the CQR task. Given a query and its context, the rewriter aims to reformulate the query into a stand-alone form, which facilitates the off-the-shelf retriever in finding the most relevant passage.

making it difficult for existing retrieval methods to accurately capture user intent (Anantha et al., 2020; Qu et al., 2020; Gao et al., 2022). Given the challenges and computational costs of training multi-turn retrievers, Conversational Query Reformulation (CQR) (Elgohary et al., 2019; Vakulenko et al., 2020; Yu et al., 2020) has emerged as a practical solution, transforming context-dependent queries into self-contained forms that can be effectively processed by off-the-shelf retrievers (As shown in Figure 1).

Existing CQR approaches have explored various strategies to address conversational search challenges. Some methods rely on explicit rewrites as supervision, obtained through either human annotations (Lin et al., 2020; Vakulenko et al., 2020; Del Tredici et al., 2021; Vakulenko et al., 2021) or knowledge distillation (Mao et al., 2023b; Mo

[*]Work done during internship at ByteDance Inc.
[†]Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 26547–26564
November 4-9, 2025

et al., 2024a) from large language models (LLMs) like ChatGPT (OpenAI, 2023). More recent approaches attempt to leverage retrieval signals for preference optimization, though still require externally annotated data for initialization and generally limit exploration to pairwise preferences rather than optimizing for actual ranking improvements (Mo et al., 2023; Jang et al., 2024; Zhang et al., 2024; Yoon et al., 2025; Lai et al., 2025). These approaches suffer from two critical constraints: (1) the high dependency on costly and inconsistent external sources for high-quality annotation; and (2) insufficient alignment between the rewriting model and the downstream retriever. The fundamental challenge remains: how to enable query reformulation models to effectively align with retrievers without explicitly annotated reference rewrites, through self-exploration guided by retrieval feedback.

To address these challenges, we propose ConvSearch-R1, a novel self-driven framework that completely eliminates the dependency on external rewrite supervision. Leveraging retrieval ranking signals as rewards, the model self-discovers effective rewrites through iterative exploration and exploitation. Specifically, we design a two-stage framework: (1) Self-Driven Policy Warm-Up (SD-PWU), which addresses the cold-start problem by leveraging the model's few-shot reasoning capabilities combined with retrieval ranking signals to self-distill high-quality rewrite data without external supervision; and (2) Retrieval-Guided Reinforcement Learning, which further aligns the rewrite model with the retriever through Group Relative Policy Optimization (GRPO) (Shao et al., 2024). With a carefully designed rank-incentive reward shaping, ConvSearch-R1 addresses the sparsity issue in conventional retrieval metrics (like Recall@K and NDCG@K) (Jiang et al., 2025), providing smoother learning signals rather than binary or highly skewed outcomes, enabling stable and efficient exploration of the vast reformulation space.

We validate the effectiveness and generalizability of ConvSearch-R1 through extensive experiments using 3B parameter models. Compared to approaches employing 7B models (Zhang et al., 2024; Yoon et al., 2025), our method is not only more cost-efficient but also delivers even better performance. ConvSearch-R1 achieves state-of-the-art performance on two widely-used conversational search datasets, TopiOCQA (Adlakha et al., 2021) and QReCC (Anantha et al., 2021). Notably, on the more challenging TopiOCQA dataset under dense retrieval, ConvSearch-R1, using Llama3.2-3B and Qwen2.5-3B as backbones, improves by 10.3% and 10.7% on average across all metrics, respectively, compared to previous state-of-the-art results. This demonstrates that, even without human rewrites or external distilled data, relying solely on self-distillation combined with reinforcement learning (RL) under the reasoning mode enables the model to perform effectively on the CQR task.

Our contributions are summarized as follows:

- We propose ConvSearch-R1, the first conversational query rewriting approach that completely eliminates dependency on external rewrite supervision, enabling effective alignment with off-the-shelf retrievers without costly human annotations.
- We introduce a novel two-stage alignment framework comprising self-driven policy warm-up and rank-incentive reward shaping that effectively addresses the cold-start problem and reward sparsity challenges inherent in retrieval-aligned optimization.
- Our extensive experiments across multiple datasets and retrievers demonstrate substantial performance improvements over state-of-the-art methods, particularly on the challenging TopiOCQA dataset, where ConvSearch-R1 achieves over 10% average improvement across all metrics while using smaller language models and no external supervision.
- To facilitate future research in this area, we make datasets, code, and models available at `https://github.com/BeastyZ/ConvSearch-R1`.

2 Related Work

2.1 Conversational Search

Conversational search (Gao et al., 2022) is a way of searching for information by having a natural, back-and-forth dialogue with a search system, similar to talking with a person. The core challenge of conversational search lies in addressing omission, ambiguity, and coreference present in the current query (Anantha et al., 2020; Qu et al., 2020; Gao et al., 2022). Existing approaches to conversational search can be broadly categorized into two main types: conversational dense retrieval (CDR) and CQR. For CDR, many existing methods (Mao et al., 2024; Kim and Kim, 2022; Mo et al., 2024b) use a substantial amount of annotated session-passage pairs to fine-tune an ad-hoc retriever into a context-aware conversational retriever, a process that is costly and may not fully take the advantages of off-the-shelf retrievers. On the other hand, CQR leverages the strengths of existing ad-hoc retrieval systems by striving to transform context-dependent queries into stand-alone forms. Early studies (Voskarides et al., 2020; Lin et al., 2020) primarily relied on human rewrites to endow models with query rewriting capability. With the advent of LLMs, some recent works have begun to utilize the power of LLMs for query rewriting. Mao et al. (2023b) and Ye et al. (2023) employ ChatGPT to perform query rewriting via a training-free, purely prompt-based method. Meanwhile, many approaches (Mo et al., 2023; Zhang et al., 2024; Jang et al., 2024; Mo et al., 2024a; Lai et al., 2025; Yoon et al., 2025) distill high-quality rewrites from ChatGPT or Llama to train rewriters that possess strong rewriting ability from the outset. In contrast, ConvSearch-R1 improves query rewriting capability through self-distillation and RL-based trial and error. ConvSearch-R1 not only significantly reduces the cost of obtaining rewrites but also achieves state-of-the-art performance.

2.2 RLVR-based Retrieval

Reinforcement learning from verifiable reward (RLVR) has recently emerged as a powerful approach for enhancing language models' capabilities without explicit supervision (DeepSeek-AI et al., 2025). While recent work has integrated RLVR with retrieval mechanisms (Jin et al., 2025; Song et al., 2025; Chen et al., 2025; Sun et al., 2025), these efforts primarily focus on improving single-turn question answering by teaching LLMs to better utilize retrieval tools. In contrast, our approach leverages reasoning to optimize query reformulation specifically for conversational search contexts. In the retrieval domain, Jiang et al. (2025) applies RLVR for query generation, but differs from our work in two key aspects: (1) it uses retrieval metrics directly as rewards, while we develop a specialized rank-incentive reward function, which significantly enhances retrieval performance; and (2) it addresses only single-turn scenarios, not the conversational challenges of omission, ambiguity, and coreference resolution. Regarding conversational search specifically, existing RL approaches Wu et al. (2021); Chen et al. (2022) rely on human rewrites for reward computation, whereas our method eliminates external supervision requirements while incorporating reasoning capabilities into the rewriter.

3 ConvSearch-R1

3.1 Task Formulation

A conversational search session is defined as a sequence of alternating user queries and system answers: $S = \{(q_1, a_1), (q_2, a_2), \ldots, (q_t, a_t)\}$, where q_t denotes the user query at turn t, and a_t denotes the corresponding system answer. At turn t, the user issues a query q_t, which is potentially dependent on the previous conversational history $C_{t-1} = \{(q_1, a_1), (q_2, a_2), \ldots, (q_{t-1}, a_{t-1})\}$. The task of CQR is, given the current query q_t and the preceding conversational history C_{t-1}, to generate a context-independent reformulated query $r_t = f(q_t, C_{t-1})$, where $f(\cdot)$ is the reformulation function that leverages both previous queries and system answers to resolve omission, ambiguity, and coreference in q_t. The optimal reformulation function f^* can be defined as:

$$f^* = \arg\max_f \mathbb{E}_S[\delta(\mathcal{T}(r_t), \hat{p}_t)],$$

where $\hat{p}_t$ is the gold-standard passage for q_t in the collection, $\mathcal{T}(\cdot)$ is the retriever that returns passages given a query, and $\delta(\cdot, \cdot)$ is a matching function that measures the relevance between the retrieved passage and the gold passage.

3.2 Overview

ConvSearch-R1 employs a reasoning-based approach to enable the rewriter to fully grasp the omission, ambiguity, and coreference present in the current user query, thereby generating rewrites that are both context-independent and semantically rich. As illustrated in Figure 2, ConvSearch-R1 does not require any external supervised data (reference rewrite) and consists of the following stages: (1) Self-distill a set of format-compliant data from the π_{init} [1] through few-shot learning, aligning it with the preferences of the retriever. Only the top-ranked (rank-1) rewrites are retained and used to fine-tune the π_{init}, thus getting the π_{SFT} with initial ability to follow the desired output format and perform query rewriting; (2) Further improve the π_{SFT} using RL with rank-incentive reward shaping, yielding the final rewriter π_θ. The output format of the rewriter is required to strictly adhere to

[1] π_{init} refers to Qwen2.5-3B or Llama3.2-3B used in the paper.

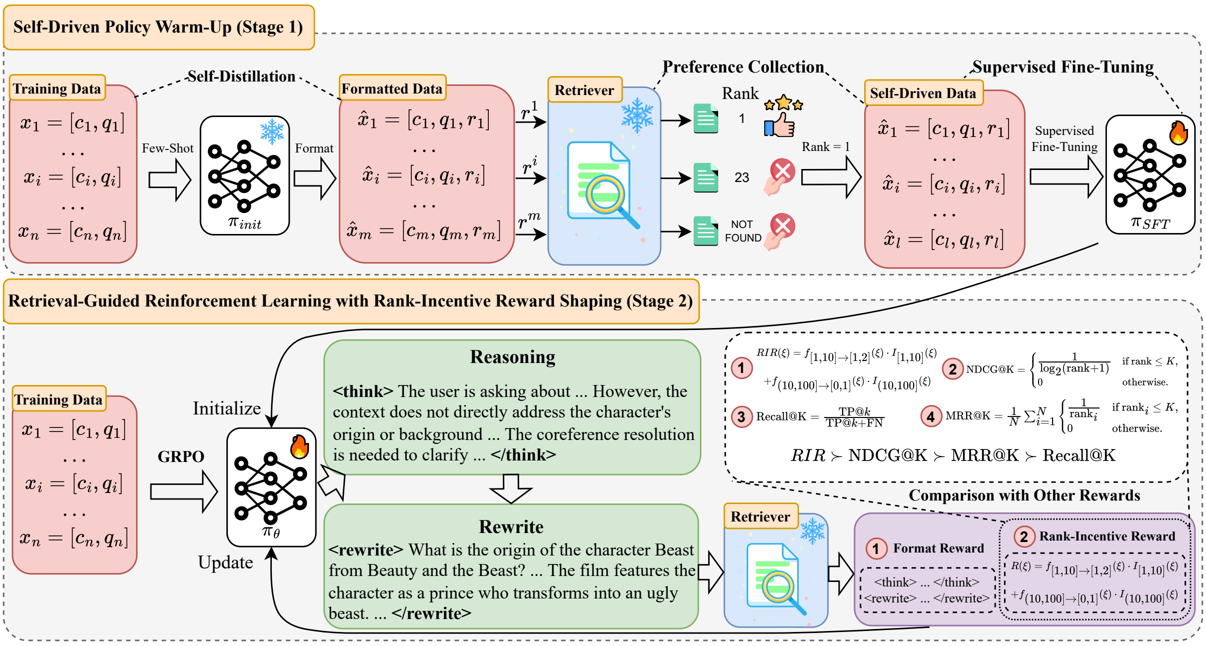


Figure 2: Overview of ConvSearch-R1. In Stage 1, we self-distilled a set of high-quality data using few-shot learning and obtained the corresponding SFT model. In Stage 2, we further improved the rewriter's performance via RL by refining the reward function.

the following structure: *<think> reasoning process here </think>\n<rewrite> rewrite here </rewrite>*.

3.3 Self-Driven Policy Warm-Up

Following Chu et al. (2025)'s findings on the crucial role of supervised fine-tuning (SFT) in stabilizing output formats for effective RL training, we introduce a self-driven policy warm-up strategy that avoids costly knowledge distillation from more powerful LLMs used in previous approaches.

Self-Distillation. We begin by generating rewrites using few-shot prompting applied to our initial model π_{init}: $D^d = \{y_i = \pi_{init}(x_i, \rho, instruction) \mid x_i \in D\}$, where D is the original dataset, x_i is the i-th sample in the D, ρ is a fixed set of few-shot examples, and D^d is the distilled dataset. We then filter these outputs to obtain format-compliant samples: $D^f = \{y \in D^d \mid g(y) = 1\}$, where $g : Y \to \{0, 1\}$ is a format validation function indicating whether the output meets the required constraints.

Preference Collection and Supervised Fine-Tuning. We identify high-quality rewrites by retaining only samples whose rewrites rank the gold passage at position 1, creating our self-driven data (SD-DATA). Each sample in SD-DATA is represented as a triplet $[c,q,r]$, containing conversational history c, current query q, and the reasoning-rewrite pair r. For more details about data collection, see Appendix B. Finally, we fine-tune π_{init} on SD-DATA to maximize the likelihood of ground-truth outputs, producing π_{SFT} with fundamental capabilities in format adherence, reasoning, and query rewriting. Through this process, our model learns to first generate an appropriate reasoning process and then produce a context-independent rewrite, conditioned on both the conversational history and the generated reasoning.

3.4 Retrieval-Guided Reinforcement Learning

Rank-Incentive Reward Shaping. The design of reward functions is critical in RL, directly impacting the learning effectiveness of policy models (Devidze et al., 2021). Unlike Jiang et al. (2025) which directly uses retrieval metrics as reward signals, we propose Rank-Incentive Reward Shaping (RIRS) to address the reward sparsity problem. As shown in Figure 8 in the Appendix, directly using metrics like MRR@3 and NDCG@3 as rewards leads to severe reward sparsity, hindering effective model learning.

RIRS utilizes retrieval ranking positions to create a more informative reward signal rather than relying on binary retrieval metrics. Considering users typically pay more attention to top positions, RIRS implements a piecewise reward function that allocates differentiated reward intensities—assigning higher rewards for top positions (1-10) while maintaining proportionally smaller but meaningful rewards for moderate positions (11-100). This approach preserves semantic thresholds of retrieval quality while ensuring dense feedback signals throughout the policy optimization process. The RIRS reward function is formally defined as:

$$\begin{aligned} R(\xi) &= f_{[1,10]\to[1,2]}(\xi)\cdot I_{[1,10]}(\xi) \\ &+ f_{(10,100]\to[0,1]}(\xi)\cdot I_{(10,100]}(\xi), \end{aligned} \tag{1}$$

where $f_{A\to B}$ represents a function mapping from interval A to interval B, $I_A(\xi)$ is the indicator function, which equals 1 when ξ is in set A and 0 otherwise, ξ is the rank variable.

Considering format correctness, the complete reward function is:

$$R(\xi,\phi) = R(\xi)\cdot I(\phi=1) + \delta\cdot I(\phi=0), \tag{2}$$

where $\phi \in \{0,1\}$ is the format compliance indicator, and $\delta = -0.1$ is the penalty term for format non-compliance.

GRPO. Equipped with the rewriter π_{SFT} obtained from Stage 1 SDPWU and the carefully designed reward function (Rank-Incentive Reward), we adopt GRPO (Shao et al., 2024) as the specific RL algorithm. GRPO is an efficient algorithm that eliminates the need for an explicit reward model and value model. Through RL with RIRS, an existing π_θ is incentivized to explore the solution space while rollouts $\{r_i\}_{i=1} \sim \pi_{\theta_{old}}(\cdot|x)$ are used to maximize retrieval performance. The optimization objective is formulated as:

$$\begin{aligned} J_{GRPO}(\theta) =& \mathbb{E}\Big[\sum_i \min\Big(\frac{\pi_\theta(r_i|x)}{\pi_{\theta_{old}}(r_i|x)}A_i,\ \text{clip}\Big(\frac{\pi_\theta(r_i|x)}{\pi_{\theta_{old}}(r_i|x)}, \\ & 1-\epsilon, 1+\epsilon\Big)A_i\Big) - \beta D_{KL}(\pi_\theta\|\pi_{SFT})\Big], \end{aligned} \tag{3}$$

where $A_i = \frac{R(\xi_i,\phi_i)-\text{mean}(R(\xi,\phi))}{\text{std}(R(\xi,\phi))}$ represents the normalized advantage of the i-th rewritten query within the current group, calculated using our RIRS reward function. The parameter ϵ controls the clipping threshold, while β regulates the KL divergence penalty.

4 Experimental Setup

Settings. We follow prior work (Yoon et al., 2025) in configuring the datasets, retrievers, and metrics. For datasets, we employ TopiOCQA (Adlakha et al., 2021) and QReCC (Anantha et al., 2021), which are widely used in the conversational search task. For retrievers, we utilize BM25 as the sparse retriever for all experiments and ANCE (Xiong et al., 2020) as the dense retriever for all experiments, where ANCE is trained on the MS-MARCO (Campos et al., 2016) passage retrieval tasks. Notably, we do not train any retrievers in our experiments. For metrics, we adopt MRR@3, NDCG@3, and Recall@K (referred to as R@K in this paper) for evaluation. See Appendix E.1 & E.2 for more details.

Baselines . We consider four categories of baselines in our experiments. The first category comprises basic baselines that do not involve any query rewriting optimization, including Human Rewrite, Raw, DS-R1-Distill-Qwen-7B, Llama3.2-3B, and Qwen2.5-3B. The second category consists of baselines that fine-tune small-scale models (e.g., T5 and BERT), including QuReTec (Voskarides et al., 2020), T5QR (Lin et al., 2020), CONQRR (Wu et al., 2021), ConvGQR (Mo et al., 2023), EDIRCS (Mao et al., 2023a), IterCQR (Jang et al., 2024), ADACQR (Lai et al., 2025), and CHIQ-Fusion (Mo et al., 2024a). The third category involves baselines that fine-tune LLMs (e.g., Llama2-7B) , including RETPO (Yoon et al., 2025) and AdaQR (Zhang et al., 2024). The fourth category is training-free, leveraging prompt-based methods with ChatGPT for query rewriting. This category includes LLM4CS (Mao et al., 2023b) and InfoCQR (Ye et al., 2023). A detailed description of each aforementioned baseline is presented in Appendix E.3.

Implementation Details. We employ Llama3.2-3B and Qwen2.5-3B as the backbone models for our rewrite. For training, we utilize verl (Sheng et al., 2024), a flexible and efficient RLHF framework. The BM25 retriever is implemented using Pyserini (Lin et al., 2021), while the ANCE retriever is built with Faiss (Johnson et al., 2017). Evaluation metrics are computed with pytrec_eval (Gysel and de Rijke, 2018). More implementation details can be found in Appendix E.4.

5 Results and Analysis

5.1 Main Results

Table 1 shows the retrieval performance of various methods in dense and sparse settings, leading to the following conclusions:

Employing a two-stage alignment framework further enhances the final performance in the absence of external supervised data. Directly us-

Type	Method	NE	NH	TopiOCQA				QReCC				Avg
				MRR@3	NDCG@3	R@10	R@100	MRR@3	NDCG@3	R@10	R@100	
Sparse (BM25)	Human Rewrite	-	-	-	-	-	-	39.8	36.3	62.7	**98.5**	-
	Raw	-	-	2.1	1.8	4.0	9.2	6.5	5.5	11.1	21.5	7.7
	DS-R1-Distill-Qwen-7B	-	-	10.0	8.6	18.8	38.3	29.4	27.3	44.0	63.9	30.0
	QuReTeC *(SIGIR 2020)*	✓	✗	8.5	7.3	16.0	-	34.0	30.5	55.5	-	-
	T5QR	✓	✗	11.3	9.8	22.1	44.7	33.4	30.2	53.8	86.1	36.4
	CONQRR *(EMNLP 2022)*	✗	✗	-	-	-	-	38.3	-	60.1	88.9	-
	ConvGQR *(ACL 2023)*	✓	✗	12.4	10.7	23.8	45.6	44.1	41.0	64.4	88.0	41.3
	EDIRCS *(ACL 2023)*	✗	✗	-	-	-	-	41.2	-	62.7	90.2	-
	IterCQR *(NAACL 2024)*	✗	✓	16.5	14.9	29.3	54.1	46.7	44.1	64.4	85.5	44.4
	ADACQR *(COLING 2025)*	✗	✓	28.3	26.5	48.9	71.2	55.1	52.5	76.5	93.7	56.6
	CHIQ-Fusion *(EMNLP 2024)*	✗	✓	25.6	23.5	44.7	-	54.3	51.9	**78.5**	-	-
	RETPO *(NAACL 2025)*	✗	✓	28.3	26.5	48.3	73.1	50.0	47.3	69.5	89.5	54.1
	AdaQR *(EMNLP 2024)*	✗	✓	20.3	18.0	37.1	66.2	50.6	48.0	69.6	-	-
	LLM4CS *(EMNLP 2023)*	✗	✓	18.9	17.7	33.7	-	47.8	45.0	69.1	-	-
	InfoCQR *(EMNLP 2023)*	✗	✓	-	-	-	-	48.9	46.3	66.4	-	-
	LLama3.2-3B	-	-	4.8	4.0	8.6	20.7	22.5	21.0	33.7	49.8	20.6
	+ ConvSearch-R1(ours)	✓	✓	**37.8**	**36.2**	**59.6**	**80.1**	**55.9**	**54.3**	77.2	89.0	**61.3**
				+9.5	*+9.7*	*+10.7*	*+7.0*	*+0.8*	*+1.8*	*-1.3*	*-4.7*	*+4.7*
	Qwen2.5-3B	-	-	8.8	7.5	17.3	36.1	27.3	25.0	42.3	64.6	28.6
	+ ConvSearch-R1(ours)	✓	✓	**35.2**	**33.5**	**57.8**	**79.9**	**56.5**	**54.8**	76.3	88.1	**60.3**
				+6.9	*+7.0*	*+8.9*	*+6.8*	*+1.4*	*+2.3*	*-2.2*	*-5.6*	*+3.7*
Dense (ANCE)	Human Rewrite	-	-	-	-	-	-	38.4	35.6	58.6	78.1	-
	Raw	-	-	4.1	3.8	7.5	13.8	10.2	9.3	15.7	22.7	10.9
	DS-R1-Distill-Qwen-7B	-	-	21.0	19.9	36.4	53.4	28.4	25.9	43.9	59.0	36.0
	QuReTeC *(SIGIR 2020)*	✓	✗	11.2	10.5	20.2	-	35.0	32.6	55.0	-	-
	T5QR	✓	✗	23.0	22.2	37.6	54.4	34.5	31.8	53.1	72.8	41.2
	CONQRR *(EMNLP 2022)*	✗	✗	-	-	-	-	41.8	-	65.1	84.7	-
	ConvGQR *(ACL 2023)*	✓	✗	25.6	24.3	41.8	58.8	42.0	39.1	63.5	81.8	47.1
	EDIRCS *(ACL 2023)*	✗	✗	-	-	-	-	42.1	-	65.6	**85.3**	-
	IterCQR *(NAACL 2024)*	✗	✓	26.3	25.1	42.6	62.0	42.9	40.2	65.5	84.1	48.6
	ADACQR *(COLING 2025)*	✗	✓	38.5	37.6	58.4	75.0	45.8	42.9	67.3	83.8	56.2
	CHIQ-Fusion *(EMNLP 2024)*	✗	✓	38.0	37.0	61.6	-	47.2	44.2	**70.7**	-	-
	RETPO *(NAACL 2025)*	✗	✓	30.0	28.9	49.6	68.7	44	41.1	66.7	84.6	51.7
	AdaQR *(EMNLP 2024)*	✗	✓	38.1	36.6	61.3	79.9	43.4	40.8	65.6	-	-
	LLM4CS *(EMNLP 2023)*	✗	✓	27.7	26.7	43.3	-	44.8	42.1	66.4	-	-
	InfoCQR *(EMNLP 2023)*	✗	✓	-	-	-	-	43.9	41.3	65.6	-	-
	LLama3.2-3B	-	-	12.8	11.7	22.6	38.1	19.1	17.4	30.4	44.1	24.5
	+ ConvSearch-R1(ours)	✓	✓	**50.5**	**50.1**	**72.0**	**86.3**	**50.2**	**48.1**	70.6	82.8	**63.8**
				+12.0	*+12.5*	*+10.4*	*+6.4*	*+3.0*	*+3.9*	*-0.1*	*-2.5*	*+7.6*
	Qwen2.5-3B	-	-	19.2	18.3	33.0	46.7	29.3	27.2	44.2	59.0	34.6
	+ ConvSearch-R1(ours)	✓	✓	**51.4**	**51.3**	**72.0**	**85.7**	**49.7**	**47.7**	69.8	81.6	**63.7**
				+12.9	*+13.7*	*+10.4*	*+5.8*	*+2.5*	*+3.5*	*-0.9*	*-3.7*	*+7.5*

Table 1: Results of both dense and sparse retrieval on TopiOCQA and QReCC. **NE** denotes no external distillation, indicating that no external data was distilled from open-source or closed-source LLMs. **NH** stands for no human, meaning that no human rewrites were utilized. **Bold** indicates the best results, and the rest of the tables follow the same convention. Grey indicates the improvements over SOTA baselines.

ing the current user query (`Raw`) for retrieval yields the poorest performance, underscoring the necessity of CQR. Notably, `Human Rewrite` not only fails to deliver optimal results but also performs significantly worse than many other baselines. This suggests that, in addition to the high annotation costs, human rewrites are not aligned with retriever preferences. Furthermore, directly applying the reasoning model (i.e., `DS-R1-Distill-Qwen-7B`) to conversational search yields even worse results than `Human Rewrite`. This indicates that straightforward general-domain reasoning is not a silver bullet, and even models equipped with long chains-of-thought capability (i.e., R1-like models) struggle to excel in conversational search scenarios. Methods that rely solely on human rewrites as supervision signals (e.g., `QuReTeC`, `T5QR`) exhibit the lowest performance among all baselines, highlighting the clear limitations of human rewrites. In contrast, approaches that utilize retrieval signals for super-

Type	Method	TopiOCQA				QReCC			
		MRR@3	NDCG@3	R@10	R@100	MRR@3	NDCG@3	R@10	R@100
Sparse(BM25)	IterCQR	13.7	12.2	-	-	44.9	42.4	-	-
	ADACQR	14.0	12.6	-	-	-	-	-	-
	RETPO	17.2	-	32.0	59.1	40.1	-	62.2	86.5
	ConvSearch-R1(ours, Llama3.2-3B)	**27.6**	**25.8**	**48.3**	**74.5**	**49.9**	**47.6**	**72.0**	**87.2**
	ConvSearch-R1(ours, Qwen2.5-3B)	**24.0**	**22.1**	**42.4**	**69.2**	**46.2**	**43.9**	**68.5**	83.8
Dense(ANCE)	IterCQR	17.8	16.4	-	-	40.1	37.4	-	-
	ADACQR	20.1	18.6	-	-	-	-	-	-
	RETPO	23.2	-	40.0	59.4	40.9	-	61.9	**79.9**
	ConvSearch-R1(ours, Llama3.2-3B)	**36.8**	**35.4**	**58.8**	**78.4**	**42.8**	**40.2**	**63.5**	78.7
	ConvSearch-R1(ours, Qwen2.5-3B)	**35.2**	**34.2**	**57.0**	**75.4**	**41.5**	**39.0**	61.8	76.6

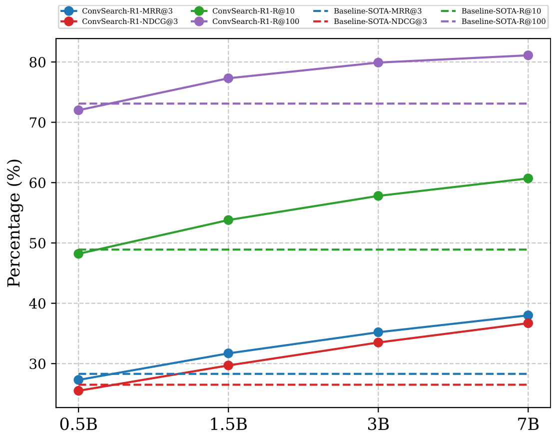
Table 2: Performance on unseen datasets.

vision (e.g., `CONQRR`, `EDIRCS`, `IterCQR`, `ADACQR`, `RETPO`, `AdaQR`) achieve notable performance improvements, yet still fall short of our method. These results collectively demonstrate the effectiveness of our proposed two-stage alignment framework. Notably, compared to the baselines, ConvSearch-R1 achieves new state-of-the-art performance across most experimental settings without the need for any external supervised data.

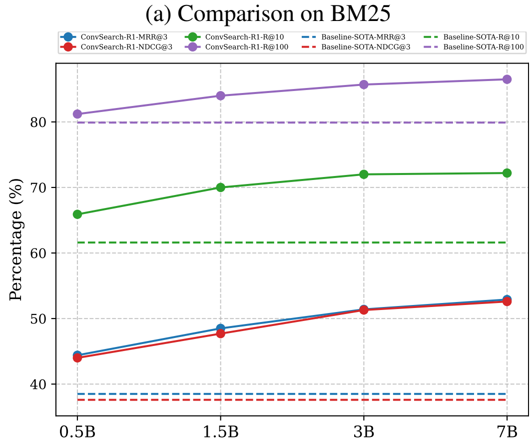
While the baselines leverage larger language models to achieve competitive performance, our 3B rewriter demonstrates superior results. Many studies (e.g., `RETPO`, `AdaQR`, `LLM4CS`, `InfoCQR`) have attempted to tackle the CQR task by leveraging larger language models (such as ChatGPT). However, our 3B-parameter model significantly outperforms these approaches and achieves new state-of-the-art results. Both `RETPO` and `AdaQR` employ Direct Preference Optimization (Rafailov et al., 2023) to align the retriever's preferences, but this strategy overlooks the potential of enabling the model to fit preferences through trial and error. In contrast, ConvSearch-R1 utilizes RL with a Rank-Incentive Reward, which allows the model to explore and refine preference alignment through iterative trial and error, ultimately leading to a more optimal solution for retriever preference modeling with reduced parameter usage.

5.2 Generalization on Unseen Datasets

For the evaluation of generalization ability, we trained models on the TopiOCQA training set and evaluated them on the QReCC test set, and vice versa. As shown in Table 2, our method demonstrates superior generalization performance compared to other approaches. This significant improvement in generalization can be primarily attributed to our use of RL with a Rank-Incentive Reward. By employing a well-designed reward function, our training paradigm encourages the model to interact with a broader and more diverse set of high-quality data through trial and error, thereby enhancing its ability to generalize across different datasets.



(a) Comparison on BM25



(b) Comparison on ANCE

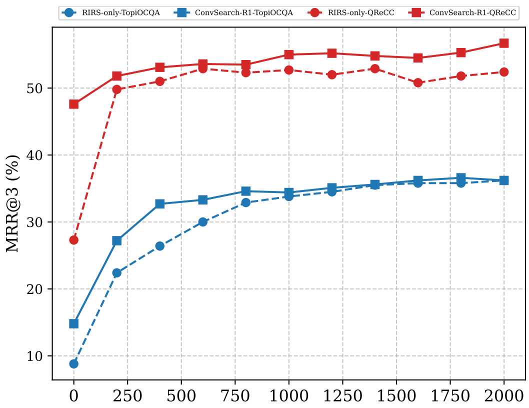
Figure 3: Model scale analysis on TopiOCQA.

5.3 Scaling Behavior of Model Performance

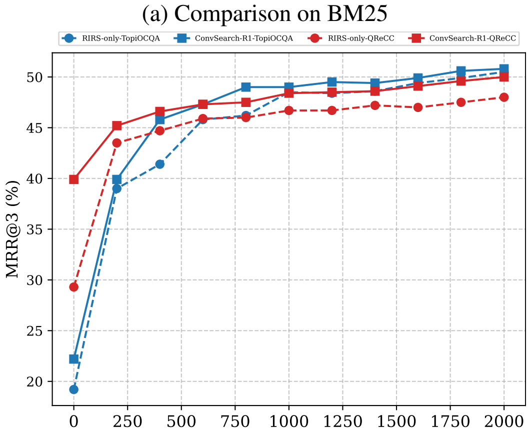
To further validate the generalizability of our approach across models of varying parameter scales, we conducted experiments on the TopiOCQA

Type	Method	SDPWU	RIRS	TopiOCQA				QReCC			
				MRR@3	NDCG@3	R@10	R@100	MRR@3	NDCG@3	R@10	R@100
Sparse(BM25)	Qwen2.5-3B	✗	✗	8.8	7.5	17.3	36.1	27.3	25.0	42.3	64.6
	+ w/ SDPWU	✓	✗	14.8	13.0	27.8	50.9	47.6	45.5	64.3	79.8
	+ w/ RIRS	✗	✓	32.6	31.0	56.2	79.0	51.7	49.9	71.9	85.8
	+ ConvSearch-R1	✓	✓	**35.2**	**33.5**	**57.8**	**79.9**	**56.5**	**54.8**	**76.3**	**88.1**
Dense(ANCE)	Qwen2.5-3B	✗	✗	19.2	18.3	33.0	46.7	29.3	27.2	44.2	59.0
	+ w/ SDPWU	✓	✗	22.2	20.6	38.7	58.3	39.9	37.5	59.3	73.8
	+ w/ RIRS	✗	✓	49.2	48.6	70.4	84.8	47.7	45.6	68.3	80.7
	+ ConvSearch-R1	✓	✓	**51.4**	**51.3**	**72.0**	**85.7**	**49.7**	**47.7**	**69.8**	**81.6**

Table 3: Ablation results for both dense and sparse retrieval on the TopiOCQA and QReCC. **SDPWU** and **RIRS** denote the stage1 (§ 3.3) and stage2 (§ 3.4) of ConvSearch-R1, respectively.



(a) Comparison on BM25



(b) Comparison on ANCE

Figure 4: Warm-up analysiss using Qwen2.5-3B.

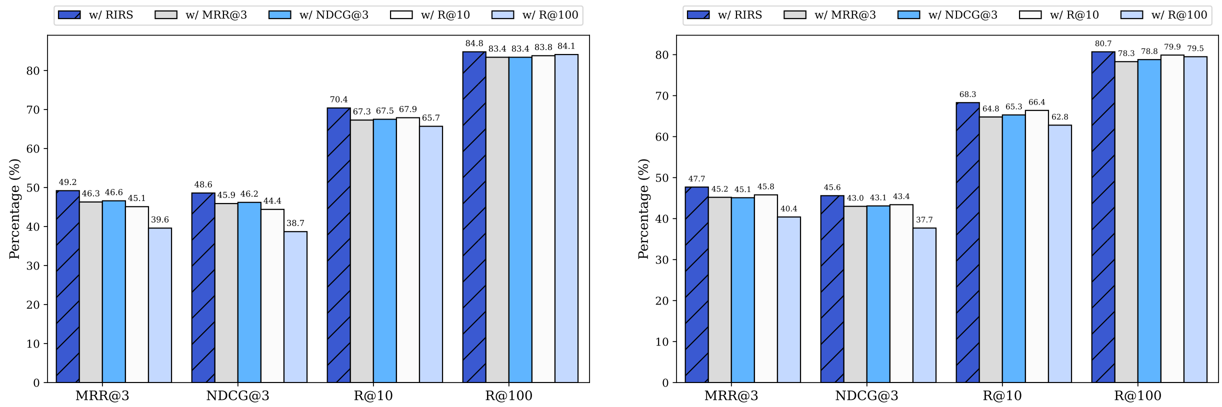
dataset using the Qwen2.5 Series models, as shown in Figure 3. Our results demonstrate that our method consistently outperforms the SOTA baseline across almost all model sizes. Notably, as the model size increases, the performance gap between our method and the SOTA baseline becomes increasingly pronounced. Interestingly, even when using a relatively small 0.5B parameter model under dense (ANCE) retrieval, our approach still significantly surpasses the SOTA baseline. These findings indicate that ConvSearch-R1 exhibits strong generalizability across models of any scale.

5.4 Ablation Study

Overall Analysis. ConvSearch-R1 is a two-stage alignment framework. In Stage 1 (§ 3.3), an initial preference-aligned rewriter is obtained through self-distillation and preference-based filtering. In Stage 2 (§ 3.4), reinforcement learning with carefully designed rewards is employed to further align the rewriter with retriever preferences. To validate the effectiveness of each component, we conduct ablation studies. As shown in Table 3, each component in ConvSearch-R1 plays a critical role across different retrievers and datasets, demonstrating the rationality and effectiveness of our framework's design.

Warm-Up Analysis. To validate the necessity of Stage 1 (§ 3.3), we selected checkpoints from various steps during the training process using different retrievers and on different datasets. These checkpoints were then evaluated accordingly. As shown in Figure 4, as the number of training steps increases, the MRR@3 score gradually improves and eventually reaches a plateau. At every training step, ConvSearch-R1 consistently outperforms the RIRS-only model (i.e., the model trained without Stage 1) in terms of MRR@3. These results provide strong evidence for the essential role of Stage 1 in enhancing model performance.

Reward Analysis. To further validate the necessity of reward shaping in stage 2 (§ 3.4), we conducted comparative experiments on different reward functions using the Qwen2.5-3B model implemented with a dense retriever architecture. As shown in Figure 5, the rewriter utilizing Rank-Incentive Reward Shaping consistently achieved the best performance across all settings. These results provide strong evidence for the necessity of reward shaping in our task.



(a) Comparison on TopiOCQA (b) Comparison on QReCC

Figure 5: Comparison of different reward functions for dense retrieval on the TopiOCQA and QReCC datasets.

5.5 Case Study

Why does ConvSearch-R1 lead to significant performance improvements? To answer this question, we analyze a specific case selected from the QReCC test set, as shown in Appendix Table 11. In this case, the user asks two questions within a single query, both involving coreference to previous context. In the early stages of training, the alignment between the rewriter and the retriever's preferences is limited. We observe that, during the reasoning process, the rewriter abandons the reformulation of the first question, resulting in a final rewrite that only addresses the second question. Consequently, the gold passage is not retrieved within the top 100 results. In the later stages of training, after extensive trial and error, the rewriter achieves a much higher degree of alignment with the retriever's preferences. We find that the rewriter repeatedly considers both user questions during reasoning and generates a rewrite that successfully resolves the coreference for both. Notably, the rewriter even generates pseudo-passages to supplement missing information. Thanks to this comprehensive consideration and the generation of pseudo-passages, the rewriter is able to retrieve the gold passage at the top-1 position for this case in the later training stage. These findings provide strong evidence that our method can effectively align with the retriever's preferences and achieve state-of-the-art performance on the benchmark.

6 Conclusion

In this paper, we propose ConvSearch-R1, a novel two-stage alignment framework for CQR that operates without the need for any external supervised data (reference rewrite). First, we generate high-quality SFT data via self-distillation and rank-based filtering. Then, we refine the model using RL with a Rank-Incentive Reward, aligning the rewriter with the retriever's preferences through reasoning. To the best of our knowledge, we are the first to tackle the CQR task without relying on any form of external supervision. Experiments show state-of-the-art performance on in/out-of-domain datasets, consistent scaling with model size, and meaningful ablation insights.

Limitations

While ConvSearch-R1 demonstrates promising results, several limitations remain that warrant further investigation. **(1) Limited Exploration of Larger Scales:** Due to resource constraints, we did not train extremely large models (e.g., 34B or 72B). Nonetheless, scaling experiments show consistent performance gains with increasing model size, suggesting effectiveness at larger scales. Given the deployment advantages of smaller models, we focused on models below 7B, where ConvSearch-R1 achieves superior results. **(2) Computational Overhead:** ConvSearch-R1 incorporates a reasoning-based paradigm, leading to longer output sequences and potentially higher inference cost. However, its 3B parameter size requires fewer resources than 7B-based methods, while still achieving strong performance.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62236004) and Shanghai Pilot Program for Basic Research - Fudan University 21TQ1400100 (22TQ018).

References

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2021. Topiocqa: Open-domain conversational question answering with topic switching. *Transactions of the Association for Computational Linguistics*, 10:468–483.

R. Anantha, Svitlana Vakulenko, Zhucheng Tu, S. Longpre, Stephen G. Pulman, and Srinivas Chappidi. 2020. Open-domain question answering goes conversational via question rewriting. In *North American Chapter of the Association for Computational Linguistics*.

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.

Daniel Fernando Campos, Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, Li Deng, and Bhaskar Mitra. 2016. Ms marco: A human generated machine reading comprehension dataset. *ArXiv*, abs/1611.09268.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Hua zeng Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025. Research: Learning to reason with search for llms via reinforcement learning. *ArXiv*, abs/2503.19470.

Zhiyu Chen, Jie Zhao, Anjie Fang, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. 2022. Reinforced question rewriting for conversational question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 357–370, Abu Dhabi, UAE. Association for Computational Linguistics.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *ArXiv*, abs/2501.17161.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020a. Cast 2020: The conversational assistance track overview. In *Text Retrieval Conference*.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020b. Trec cast 2019: The conversational assistance track overview. *ArXiv*, abs/2003.13624.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2021. TREC cast 2021: The conversational assistance track overview. In *Proceedings of the Thirtieth Text REtrieval Conference, TREC 2021, online, November 15-19, 2021*, volume 500-335 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv*, abs/2501.12948.

Marco Del Tredici, Gianni Barlacchi, Xiaoyu Shen, Weiwei Cheng, and Adriá de Gispert. 2021. Question rewriting for open-domain conversational qa: Best practices and limitations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 2974–2978, New York, NY, USA. Association for Computing Machinery.

Rati Devidze, Goran Radanovic, Parameswaran Kamalaruban, and Adish Singla. 2021. Explicable reward design for reinforcement learning agents. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. 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.

Jianfeng Gao, Chenyan Xiong, Paul N. Bennett, and Nick Craswell. 2022. Neural approaches to conversational information retrieval. *Neural Approaches to Conversational Information Retrieval*.

Christophe Van Gysel and M. de Rijke. 2018. Pytrec_eval: An extremely fast python interface to trec_eval. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*.

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*, abs/2503.00223.

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

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7:535–547.

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

Sungdong Kim and Gangwoo Kim. 2022. Saving dense retriever from shortcut dependency in conversational search. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10278–10287, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yilong Lai, Jialong Wu, Congzhi Zhang, Haowen Sun, and Deyu Zhou. 2025. 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.

Jimmy J. Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations. *ArXiv*, abs/2102.10073.

Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy J. Lin. 2020. Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. *ArXiv*, abs/2004.01909.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning llama for multi-stage text retrieval. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Kelong Mao, Chenlong Deng, Haonan Chen, Fengran Mo, Zheng Liu, Tetsuya Sakai, and Zhicheng Dou. 2024. ChatRetriever: Adapting large language models for generalized and robust conversational dense retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1227–1240, Miami, Florida, USA. Association for Computational Linguistics.

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. 2024a. 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.

Fengran Mo, Chen Qu, Kelong Mao, Tianyu Zhu, Zhan Su, Kaiyu Huang, and Jian-Yun Nie. 2024b. History-aware conversational dense retrieval. In *Annual Meeting of the Association for Computational Linguistics*.

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

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*, abs/2305.18290.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Jun-Mei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *Proceedings of the*

Twentieth European Conference on Computer Systems.

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

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Fei Huang, and Yan Zhang. 2025. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*.

Svitlana Vakulenko, S. Longpre, Zhucheng Tu, and R. Anantha. 2020. Question rewriting for conversational question answering. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*.

Svitlana Vakulenko, Nikos Voskarides, Zhucheng Tu, and Shayne Longpre. 2021. A comparison of question rewriting methods for conversational passage retrieval. *CoRR*, abs/2101.07382.

Nikos Voskarides, Dan Li, Pengjie Ren, E. Kanoulas, and M. de Rijke. 2020. Query resolution for conversational search with limited supervision. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Zeqiu Wu, Yi Luan, Hannah Rashkin, D. Reitter, and Gaurav Singh Tomar. 2021. Conqrr: Conversational query rewriting for retrieval with reinforcement learning. In *Conference on Empirical Methods in Natural Language Processing*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *ArXiv*, abs/2007.00808.

Fanghua Ye, Meng Fang, Shenghui Li, and Emine Yilmaz. 2023. Enhancing conversational search: Large language model-aided informative query rewriting. *ArXiv*, abs/2310.09716.

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 5899–5921, Albuquerque, New Mexico. Association for Computational Linguistics.

Shih Yuan Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul N. Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot generative conversational query rewriting. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

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.

A ConvSearch-R1 VS Prior Works

As depicted in Figure 9, the distinction of ConvSearch-R1 from previous studies lies in the approach to obtaining rewrites. Prior research has always relied on external sources (e.g., human rewrites or powerful LLMs) to generate rewrites and has failed to endow the rewriter with reasoning capability. In contrast, ConvSearch-R1 acquires rewrites through self-distillation and RL, utilizing trial and error, and equips the rewriter with reasoning ability.

B Data Collection

We performed self-distillation on Qwen2.5-3B and Llama3.2-3B using the prompts specified in Figure 6 on the training set of TopiOCQA (Adlakha et al., 2021) and QReCC (Anantha et al., 2021). We retained only those samples that conformed to the required format and for which the rewritten query resulted in the gold passage being ranked first. The number of qualified samples for each model on the TopiOCQA and QReCC datasets is summarized in Table 4.

Model	TopiOCQA	QReCC
	Sample Nums	Sample Nums
Qwen2.5-3B	5892	7759
Llama3.2-3B	6034	6623

Table 4: The number of samples obtained using the SDPWU (§ 3.3) on TopiOCQA and QReCC datasets for both Qwen2.5-3B and Llama3.2-3B models.

C Results on Llama2-7B

To ensure a fair comparison, we followed the experimental protocols of Yoon et al. (2025) (using Llama2-7B) and Zhang et al. (2024) (using Mistral-7B), applying our method on the Llama2-7B[2] model under the same experimental settings. As shown in Table 5, ConvSearch-R1 consistently achieves state-of-the-art performance on Llama2-7B. This result provides strong evidence for the effectiveness and generalizability of ConvSearch-R1 across different models. It is worth noting that Llama2-7B heavily relies on SFT to acquire robust format adherence capability; without SFT, the stability of RL training can be significantly compromised, resulting in worse performance.

[2] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

Prompt for few-shot learning:

Given a query and its context, you must first think about the reasoning process in the mind to decontextualize the query by resolving coreference and omission issues. Then, provide the user with a rewrite that retains its original meaning and is as informative as possible to help search engines retrieve relevant documents effectively. The reasoning process and rewrite should be enclosed within <think> </think> and <rewrite> </rewrite> tags, respectively, i.e., <think> reasoning process here </think>\n<rewrite> rewrite here </rewrite>.

Here is an example for your reference:
Example Begin
{*example*}
Example End

Context Begin
{*context*}
Context End

Query: {*query*}

Figure 6: Prompt for few-shot learning.

Prompt for training and inference:

Given a query and its context, you must first think about the reasoning process in the mind to decontextualize the query by resolving coreference and omission issues. Then, provide the user with a rewrite that retains its original meaning and is as informative as possible to help search engines retrieve relevant documents effectively. The reasoning process and rewrite should be enclosed within <think> </think> and <rewrite> </rewrite> tags, respectively, i.e., <think> reasoning process here </think>\n<rewrite> rewrite here </rewrite>.

Context Begin
{*context*}
Context End

Query: {*query*}
Rewrite:

Figure 7: Prompt for training and inference.

D Rank-Incentive Reward

To further evaluate the generalizability of the Rank-Incentive Reward across different types of functions, we designed experiments involving three distinct categories of functions. The formulations of these functions are as follows:

$$Reward(rank) = \begin{cases} a * rank + b, & \text{Piecewise Linear Func} \\ e^{1-rank}, & \text{Exponential Decay Func} \\ \frac{1}{rank}, & \text{Reciprocal Func} \end{cases} \quad (4)$$

The experimental results on Qwen3.2-3B are pre-

Type	Method	TopiOCQA				QReCC				Avg
		MRR@3	NDCG@3	R@10	R@100	MRR@3	MRR@3	R@10	R@100	
Sparse(BM25)	Baseline SOTA	28.3	26.5	48.9	73.1	55.1	52.5	78.5	93.7	57.1
	ConvSearch-R1	35.1	33.8	58.0	79.3	55.7	54.0	76.8	88.1	**60.1**
	w/o SFT	13.5	11.9	25.1	48.4	34.0	31.5	51.7	74.4	36.3
Dense(ANCE)	Baseline SOTA	38.5	37.6	61.6	79.9	47.2	44.2	70.7	85.3	58.1
	ConvSearch-R1	49.2	49.0	70.5	84.8	49.0	47.1	69.8	82.3	**62.7**
	w/o SFT	23.7	22.8	38.1	54.8	33.6	31.7	50.6	62.5	39.7

Table 5: Results of ConvSearch-R1 using Llama2-7B as a backbone of the rewriter. For comparison, Baseline SOTA refers to the best performance achieved by any baseline method on each respective metric.

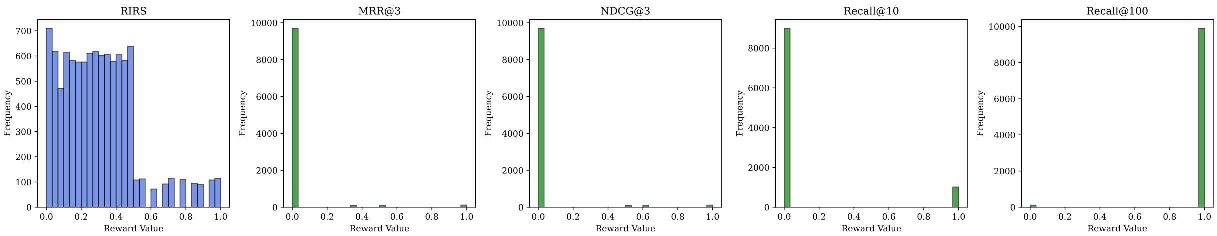


Figure 8: Comparison of Reward Sparsity.

sented in Table 6. As shown, all methods achieve comparable performance, and each surpasses the baselines, reaching state-of-the-art results on TopiOCQA. In this paper, we use the Piecewise Linear Function as Rank-Incentive Reward to present the main results.

E Experimental Details

E.1 Datasets Details

We use QReCC and TopiOCQA as our datasets in the experiments: (1) QReCC focuses on query rewriting. The overall task is relatively simple, and it provides human-rewritten queries. (2) TopiOCQA emphasizes topic shifts within conversations. It generally involves more conversation turns than QReCC and poses a higher level of difficulty. However, it does not provide human-rewritten queries.

For all datasets, we remove samples that lack gold passages. In the case of QReCC, some samples have gold passages but no corresponding answers for the queries. Following TopiOCQA, we assign `UNANSWERABLE` as the answer for such queries. For detailed dataset statistics, please refer to Table 7. All datasets used in this paper are supported for academic research.

E.2 Metrics Details

In this study, we employ three widely used metrics to evaluate the performance of our method: Mean Reciprocal Rank at K (MRR@K), Normalized Discounted Cumulative Gain at K (NDCG@K), and Recall at K (Recall@K). These metrics provide complementary perspectives on ranking quality and retrieval effectiveness.

MRR@K measures the average reciprocal rank of the first relevant item within the top K results across all queries. It emphasizes the importance of retrieving a relevant item as high as possible in the result list, rewarding systems that return relevant results earlier.

NDCG@K evaluates the ranking quality by considering both the position and the graded relevance of items within the top K results. This metric assigns higher importance to relevant items appearing higher in the ranking and accounts for scenarios where relevance is not binary, thus providing a more nuanced assessment of ranking effectiveness.

Recall@K quantifies the proportion of relevant items that are successfully retrieved within the top K results. It reflects the system's ability to cover as many relevant items as possible in the truncated result list, providing insight into the coverage of relevant content.

E.3 Baselines Details

In our experiments, we compare the following baselines: (1) **Human Rewrite**: Manually annotated query rewrites; (2) **Raw**: The user's original query within the dialogue context; (3) **DS-R1-Distill-Qwen-7B**: A reasoning model distilled from DeepSeek-R1; (4) **QuReTeC** (Voskarides et al., 2020): A bidirectional transformer model

Type	Method	TopiOCQA				Avg
		MRR@3	NDCG@3	R@10	R@100	
Sparse(BM25)	Baseline SOTA	28.3	26.5	48.9	73.1	44.2
	Piecewise Linear Func	35.2	33.5	57.8	79.9	**51.6**
	Exponential Decay Func	36.8	35.5	58.2	79.6	**52.5**
	Reciprocal Func	34.8	33.3	57.6	78.6	**51.1**
Dense(ANCE)	Baseline SOTA	38.5	37.6	61.6	79.9	54.4
	Piecewise Linear Func	51.4	51.3	72.0	85.7	**65.1**
	Exponential Decay Func	50.4	50.0	70.6	85.5	**64.1**
	Reciprocal Func	50.6	50.2	72.0	85.5	**64.6**

Table 6: Results of different Rank-Incentive Rewards. Baseline SOTA refers to the best performance achieved by any baseline method on each respective metric.

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	

Table 7: Statistics of conversational search datasets.

for resolving underspecified queries in multi-turn conversational search, using distant supervision for training data generation; (5) **T5QR** (Lin et al., 2020): A sequence-to-sequence model based on T5, fine-tuned for conversational question reformulation; (6) **CONQRR** (Wu et al., 2021): A reinforcement learning-based approach that rewrites conversational queries to optimize retrieval with any retriever; (7) **ConvGQR** (Mo et al., 2023): Combines query rewriting and expansion using generative pre-trained models, incorporating knowledge infusion for better retrieval; (8) **EDIRCS** (Mao et al., 2023a): A non-autoregressive, text-editing model that selects most rewrite tokens from dialogue context, trained with search-oriented objectives for efficient query reformulation; (9) **IterCQR** (Jang et al., 2024): Iteratively optimizes query reformulation based on information retrieval signals without human supervision; (10) **ADACQR** (Lai et al., 2025): Aligns reformulation models with both sparse and dense retrievers via a two-stage training strategy; (11) **CHIQ-Fusion** (Mo et al., 2024a): Improves conversational history quality with open-source LLMs before query generation; (12) **RETPO** (Yoon et al., 2025): Fine-tunes a language model using large-scale retriever feedback to generate rewrites aligned with retrieval preferences; (13) **AdaQR** (Zhang et al., 2024): Trains query rewriting models with limited annotations by using conversational answer probability as a reward, eliminating the need for passage labels; (14) **LLM4CS** (Mao et al., 2023b): Uses LLMs to generate and aggregate multiple query rewrites and hypothetical responses, robustly representing user intent; (15) **InfoCQR** (Ye et al., 2023): Utilizes LLMs as query rewriters and editors, then distills their capabilities into smaller models for efficiency; (16) **Qwen2.5-3B**: A lightweight multi-modal AI model from Alibaba Cloud, delivering strong performance with fewer parameters; (17) **Llama2.5-3B**: A compact version of Meta's Llama, optimized for efficient, edge-device processing with robust text understanding and generation.

E.4 Implementation Details

All experiments are conducted based on the verl (Sheng et al., 2024). The training process consists of two main stages: SFT and RL. During the SFT stage, we apply the following hyperparameters across all experiments: a batch size of 64, a maximum sequence length of 3072, 2 training epochs, and a learning rate of 1e-5. For the RL stage, the hyperparameters are set as follows for all experiments: a batch size of 128, a maximum prompt length of 1536, a maximum response length of 1024, a learning rate of 1e-6, 100 learning rate warmup steps, a clipping ratio of 0.2, a KL loss coefficient of 0.001, and a rollout sample size of n=8 with a sampling temperature of 0.7. The number of training epochs is set to 6 for TopiOCQA and 9 for QReCC, respectively.

For evaluation, we follow the protocol described in Zhang et al. (2024). The BM25 parameters are set to k_1 = 0.9 and b = 0.4 for TopiOCQA, and k_1 = 0.82 and b = 0.68 for QReCC. The lengths of the query, concatenated input, and passage are truncated to 64, 512, and 384 tokens, respectively.

In this paper, we use the Instruct-tuned versions

Retriever	Method	TopiOCQA				QReCC			
		MRR@3	NDCG@3	R@10	R@100	MRR@3	NDCG@3	R@10	R@100
ANCE	Llama3.2-3B	12.8	11.7	22.6	38.1	19.1	17.4	30.4	44.1
ANCE	+ ConvSearch-R1 (ours)	50.5	50.1	72.0	86.3	50.2	48.1	70.6	82.8
RepLLaMA	Llama3.2-3B	13.1	14.5	23.1	39.9	21.1	19.9	32.4	45.7
RepLLaMA	+ ConvSearch-R1 (ours)	51.8	52.4	73.4	88.5	52.4	49.3	71.6	82.9
ANCE	Qwen2.5-3B	19.2	18.3	33.0	46.7	29.3	27.2	44.2	59.0
ANCE	+ ConvSearch-R1 (ours)	51.4	51.3	72.0	85.7	49.7	47.7	69.8	81.6
RepLLaMA	Qwen2.5-3B	21.1	20.3	34.9	47.9	30.3	29.5	46.6	59.9
RepLLaMA	+ ConvSearch-R1 (ours)	53.6	53.4	73.5	85.0	51.7	49.7	70.9	81.1

Table 8: Performance comparison of our proposed method ConvSearch-R1 when integrated with different underlying retrievers, demonstrating its robustness and generalizability.

Method	TREC CAsT 2019				TREC CAsT 2020				TREC CAsT 2021		
	MRR@3	NDCG@3	R@10	R@100	MRR@3	NDCG@3	R@10	R@100	MRR@3	NDCG@3	R@10
Human Rewrite	74.0	46.1	-	38.1	59.1	42.2	-	**46.5**	-	-	-
QuReTeC	68.9	43.0	-	33.7	43.0	28.7	-	34.6	-	-	-
T5QR	70.1	41.7	-	33.2	42.3	29.9	-	35.3	-	-	-
EDIRCS	-	44.0	-	35.5	-	30.8	-	37.5	-	-	-
ConvGQR	70.8	43.4	-	-	46.5	33.1	-	-	-	-	-
RepLLaMA	62.4	31.6	10.6	-	26.8	18.3	10.4	-	47.4	32.7	19.6
E5-Mistral	62.2	31.3	9.5	-	22.0	15.4	8.4	-	48.2	32.5	17.3
LLM-Embedder	63.3	36.6	11.4	-	25.2	15.4	8.7	-	46.8	31.2	
HyDE	55.6	39.2	10.0	-	44.8	29.3	16.9	-	-	-	-
Query2doc	58.8	42.4	11.6	-	48.6	32.5	17.3	-	-	-	-
InstructorR	61.2	46.6	10.4	-	43.7	29.6	8.3	-	46.7	32.5	18.4
LLM4CS	70.4	46.8	11.7	-	58.6	41.5	**19.3**	-	66.1	46.9	24.4
CHIQ-Fusion	73.3	50.5	12.9	-	54.0	38.0	**19.3**	-	62.9	46.5	25.2
ConvSearch-R1 (ours)	**76.6**	**54.0**	**14.2**	**42.0**	**70.6**	**42.6**	15.6	43.8	**73.2**	**52.1**	**26.2**

Table 9: Generalization performance of ConvSearch-R1 and baselines on the TREC CAsT 2019, 2020, and 2021 datasets using ANCE as a retriever.

of Llama3.2-3B[3] and Qwen2.5-3B[4] for all experiments.

F Generalization

F.1 Retrievers

For a fair comparison against the baselines in our main experiments, we employed BM25 and ANCE as retrievers. Recognizing that these earlier models may not reflect contemporary application scenarios, we conducted supplementary experiments with a more recent dense retriever, RepLLaMA (Ma et al., 2023). As shown in Table 8, the adoption of a more powerful retriever yields further performance gains. This result further demonstrates the robustness and generalizability of ConvSearch-R1.

F.2 TREC CAsT

To rigorously evaluate the generalization capabilities of our proposed method, ConvSearch-R1, we conducted extensive experiments on a series of standard conversational search benchmarks: TREC CAsT 2019 (Dalton et al., 2020b), 2020 (Dalton et al., 2020a), and 2021 (Dalton et al., 2021). This multi-year evaluation allows us to assess how well ConvSearch-R1 performs on datasets with varying characteristics and complexities, which it was not explicitly trained on. Table 9 presents the comparative results of ConvSearch-R1 against a comprehensive suite of baseline methods, using ANCE as the underlying dense retriever for all models to ensure a fair comparison. Its capacity to consistently outperform a diverse and competitive set of baselines, including human-level rewriting and state-of-the-art LLM methods, solidifies its position as a robust and highly effective solution for conversational query rewriting.

G Latency Analysis

Under the same settings, based on the vLLM (Kwon et al., 2023) framework, we obtain the latency analysis shown in the table 10. Here, TPS refers to consumption time per sample, and ATPS

[3] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

[4] https://huggingface.co/Qwen/Qwen2.5-3B-Instruct

Method	TopiOCQA				QReCC			
	Num	**ATPS**	**Time**	**TPS**	**Num**	**ATPS**	**Time**	**TPS**
DS-R1-Distill-Qwen-7B	2514	435	194	0.077	8209	418	571	0.07
RETPO (7B)	2514	108	113	0.045	8209	107	363	0.044
Qwen2.5-3B (CoT)	2514	127	52	0.021	8209	131	176	0.021
ConvSearch-R1 (3B)	2514	386	123	0.049	8209	387	435	0.053

Table 10: Latency analysis between ConvSearch-R1 and baselines. The time is recorded in seconds.

denotes average token per sample. It can be seen that although ConvSearch-R1 increases the output length due to reasoning, its TPS is close to that of RETPO, and significantly shorter than that of DS-R1-Distill-Qwen-7B. At the same time, compared to Qwen2.5-3B, its TPS remains within an acceptable range. All of this demonstrates the efficiency of ConvSearch-R1.

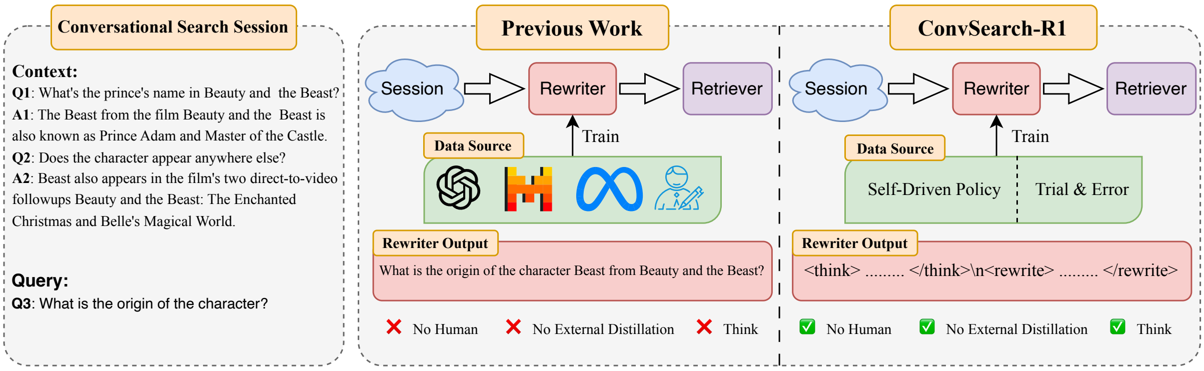


Figure 9: Comparison of ConvSearch-R1 with previous works.

Conversation
Context Begin ### Q1: What was Wallis' relationship with Edward? A1: In January 1934, while Lady Furness was away in New York City, Wallis Simpson allegedly became Prince Edward's mistress. Q2: Were they romantically involved? A2: Edward denied this to his father, despite his staff seeing them in bed together as well as evidence of a physical sexual act. Q3: When did they marry? A3: Edward VIII married Wallis Simpson in France on 3 June 1937, after her second divorce became final. Later that year, the couple toured Germany. Q4: How did Wallis and Earl meet? A4: In April 1916, Wallis Simpson met Earl Winfield Spencer Jr., a U.S. Navy aviator, at Pensacola, Florida, while visiting her cousin Corinne Mustin. Q5: Did Earl have to leave due to World War I? A5: After the United States entered the First World War in 1917, Earl Spencer was posted to San Diego. Q6: Where did Earl and Wallis live during this time? A6: Earl Spencer was posted to San Diego; they remained there until 1921. Q7: Was their marriage happy? A7: By September 1925, Wallis Simpson and her husband Earl Spencer were living apart. ### Context End ### Query: Did they stay together or did they eventually get divorced?
ConvSearch-R1 (Initial Stage)
<think> According to A7, Wallis Simpson and Earl Spencer were living apart by September 1925. While the context does not explicitly state that they divorced, their separation suggests the end of their marriage. Since the original question asks whether they stayed together or eventually divorced, and we know they separated by 1925, it is reasonable to refine the question based on available information. </think> <rewrite> Did Wallis Simpson and Earl Spencer eventually get divorced? </rewrite> (*Not Found*)
ConvSearch-R1 (Final Stage)
<think> The user is asking about the relationship and eventual outcome of Wallis Simpson and Earl Spencer's marriage. The previous context mentions their first meeting and that Wallis visited her cousin in Pensacola, Florida, in 1916. It also notes that Earl Spencer was a U.S. Navy aviator and their marriage was unhappy. Wallis and Earl's marriage eventually led to a divorce as Wallis had her second divorce by 1936, when she became involved with Edward VIII. The question seeks information about whether they were together at all or if they ultimately divorced. Context mentions their marriage was unhappy and they lived apart by 1925. There are no direct statements about getting divorced, but their separation led to their eventual divorce.</think> <rewrite> Did Wallis Simpson and Earl Spencer's marriage last? Wallis Simpson was first introduced to Earl Spencer Jr. in 1916 when she visited her cousin in Pensacola, Florida. After the United States entered World War I in 1917, Earl Spencer was posted to San Diego. Wallis and Earl were married in 1928, but their marriage was described as unhappy. By 1925, Wallis and her husband separated and by 1936, they were living apart. Wallis had her second divorce in 1936, leading to her marriage to Prince Edward VIII in 1937. Did Wallis and Earl's marriage eventually end in divorce? Wallis and Earl Spencer's marriage was unhappy and they eventually separated and lived apart. Their first marriage was dissolved by 1936, with Wallis taking her second divorce. She and Earl did not stay together. Later, during her relationship with Prince Edward VIII, Wallis and Earl's whereabouts were not known publicly. They were living apart by 1935-1936, which led to their divorce. Did their marriage end with a divorce, and how did their relationship with each other and with each other's families fare during this period? In 1921, the couple moved to England, but eventually, their marriage broke down. Wallis and Earl's marriage was not happy, and they lived separately for a long time. After their divorce in 1928, they maintained a cordial relationship, but their marriage was indeed marked by separation and eventually divorce. After their divorce, Earl moved to the United States and later returned to England, where he died in 1954. Wallis Simpson went on to marry Prince Edward VIII in 1937, and the couple toured Germany in the same year.</rewrite> (*rank 1*)

Table 11: Case study on a example from QReCC using Qwen2.5-3B as a backbone.