Compete to Complete: Co-opetition Adversarial Learning for Retrieval-Augmented Generation

Xin Liu[1,2,3], Yu-An Liu[1,2,3], Ruqing Zhang[1,2,3]*, Yixing Fan[1,2,3], Lixin Su[4], Jiafeng Guo[1,2,3], Xueqi Cheng[1,2,3]
[1]State Key Laboratory of AI Safety
[2]Institute of Computing Technology, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences [4]Baidu Inc.
{liuxin25s,liuyuan21b,zhangruqing,fanyixing,guojiafeng,cxq}@ict.ac.cn sulixinict@gmail.com

Abstract

Retrieval-augmented generation (RAG) has emerged as a promising paradigm for mitigating hallucinations in large language models (LLMs). However, the intrinsic heterogeneity between the retriever and the generator often leads to a mismatch between retrieved evidence and generation needs, hindering effective coordination. We argue that competition between discriminative retrieval and generative modeling can more effectively expose their mutual weaknesses and induce deeper interaction. Motivated by this insight, we propose CARL (Co-opetition AdveRsarial Learning), a framework that formulates retriever–generator training in RAG as a minimax game. In this game, the retriever is optimized to retrieve both useful and adversarially useless documents to challenge the generator, while the generator learns to identify useful evidence and remain robust to misleading retrievals to produce accurate answers. Experiments on seven benchmark datasets demonstrate that CARL consistently improves RAG performance, validating the effectiveness of adversarial co-opetition in enhancing retriever–generator synergy[1].

1 Introduction

To mitigate hallucination in large language models (LLMs) (Huang et al., 2025; Brown et al., 2020; Touvron et al., 2023), retrieval-augmented generation (RAG) has emerged as a promising paradigm. By grounding generation in externally retrieved evidence, RAG substantially improves factuality and relevance (Lewis et al., 2020).

Heterogeneity of RAG systems. A typical RAG system consists of a retriever and a generator. The retriever selects relevant documents from a knowledge base given a query, while the generator conditions on the retrieved content to produce an answer (Lewis et al., 2020). Despite their tight coupling at inference time, the two components solve fundamentally different problems: retrieval is a discriminative ranking task, whereas generation is an open-ended generative task over language.

This intrinsic heterogeneity has led most RAG systems to train retrievers and generators independently, with the retriever optimized solely for relevance and the generator passively consuming retrieved content (Zhang et al., 2023). As a result, retrieval preferences often misalign with generation needs, causing suboptimal evidence selection and limiting end-to-end performance (Izacard et al., 2023; Shi et al., 2024).

Limitations of cooperative RAG training. To address this mismatch, recent work has explored cooperative learning strategies that jointly train retrievers and generators (Izacard et al., 2023; Liu et al., 2023; Shi et al., 2024; Zhang et al., 2023). These approaches allow downstream generation quality to provide feedback to the retriever, partially aligning retrieval with generation objectives.

While effective, such cooperative training mainly enforces shallow functional alignment. The retriever and generator still optimize asymmetric objectives and rely on coarse-grained or unidirectional feedback (e.g., likelihood-based rewards). Crucially, cooperative methods overlook the fact that each component exhibits distinct failure modes: retrievers suffer from relevance misjudgment, while generators remain vulnerable to hallucination. The accumulation of these heterogeneous weaknesses limits deeper mutual adaptation, particularly in complex reasoning scenarios where tight retrieval–generation coupling is essential.

Our method: co-opetition via adversarial game. The principle of *cooperation through competition* (Brandenburger and Nalebuff, 2011; Zineldin, 2004) suggests that entities can achieve mutual benefit by competing in a structured setting. Compe-

*Ruqing Zhang is the corresponding author.
[1]Our code are available at https://github.com/Peter0701/CARL.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5710–5725

tition exposes weaknesses and creates informative challenges, while cooperation allows each party to leverage the insights gained for improvement. Motivated by this idea, we propose CARL, a Coopetition AdveRsarial Learning framework. CARL treats the retriever and generator as adversarial partners: each component challenges the other to expose complementary failure modes, and through this interaction, both improve. This adversarial pressure (Wang et al., 2017) generates richer, bidirectional learning signals between retrieval and generation than traditional cooperative training.

Specifically, CARL optimizes the retriever and generator via a minimax objective. The retriever is trained to retrieve query-relevant documents that effectively challenge the generator, while the generator learns to distinguish informative documents from adversarially confusing ones and to produce accurate answers. To realize this adversarial interaction, generated answers are injected into the retrieval pool as pseudo-documents, creating hard negative examples. Within this retriever–generator game, the retriever improves its discrimination ability by distinguishing real documents from generated pseudo-documents, which mitigates the impact of uninformative retrievals and provides higher-quality evidence for generation.

Experiment results. Our experiments aim to answer two research questions: (RQ1): How does CARL perform compared to state-of-the-art RAG methods? (RQ2): How do factors such as the model scale, retriever, LLM, and number of iterations impact CARL's performance? To answer the above question, CARL is evaluated on seven benchmarks across four NLP tasks. The average scores of CARL surpass all the baselines on 4 of 7 datasets. CARL outperforms ATLAS, a widely used joint training RAG framework, by 4.4% on the average score of the datasets. We also analyze different retrievers, generators, model scales and iteration numbers to demonstrate CARL's generalizability.

2 Related Works

Retrieval-augmented generation RAG integrates retrieval and generation modules, allowing language models to access external knowledge during generation (Lewis et al., 2020). Early RAG systems typically couple separately trained retrievers (e.g., dense retrievers) with generators (e.g., LLMs) (Gao et al., 2023), which can lead to objective misalignment and suboptimal performance. Recent work seeks to improve retrieval–generation synergy through joint optimization or auxiliary mechanisms. For example, ITER-RETGEN (Shao et al., 2023) uses LLM-generated pseudo-documents to guide retrieval, RECOMP (Xu et al., 2023) compresses retrieved contexts while preserving salient information, and LongLLMLingua (Jiang et al., 2024) enhances retrieval via semantic reorganization. However, simple co-operative joint training methods typically employ coarse-grained or one-way interactions, making it difficult for the retriever and generator to fully adapt to each other's performance for collaborative optimization. Differently, by allowing both components to leverage their distinct capabilities and function as mutual critics, our method promotes a deeper synergy, ultimately fostering enhanced mutual adaptation.

Adversarial learning. Adversarial learning originated from adversarial examples in computer vision (Szegedy et al., 2013; Goodfellow et al., 2014), where imperceptible input perturbations can mislead models. This line of work led to robust training methods such as FGM (Goodfellow et al., 2014) and PGD, with extensions including gradient penalties for improved stability (Ross and Doshi-Velez, 2018). In NLP, adversarial techniques are typically applied at the embedding level (Miyato et al., 2016). More broadly, Generative Adversarial Networks (GANs) formalize learning as a minimax game between competing models, enabling mutual improvement through adversarial interaction. Adversarial learning has also been explored in information retrieval and question answering. IRGAN (Wang et al., 2017) first introduced a minimax framework for jointly optimizing generative and discriminative retrieval models, achieving significant gains in retrieval performance. Similarly, AR2 (Zhang et al., 2021) proposed an adversarial ranking framework that jointly trains a dual-encoder retriever and a cross-encoder ranker. Despite these advances, adversarial learning remains largely underexplored in RAG systems. Existing methods, such as RAAT (Fang et al., 2024), focus on adaptive adversarial training to improve the robustness of retrieval augmented language models against various retrieval noises. However, they leverage limited competitive interplay between retrievers and generators, missing opportunities for mutual adaptation in QA tasks. In contrast, our work employs adversarial learning to explicitly strengthen the interac-

tion between retriever and generator, thereby improving both retrieval and generation performance.

3 Our method

Existing joint training methods for RAG primarily emphasize pure cooperation between the retriever and the generator, which often results in limited interaction and superficial adaptation. Because the two modules are optimized under heterogeneous objectives, cooperative learning alone fails to explicitly surface and correct complementary failure modes, such as retrieval relevance errors and generative hallucinations. We argue that introducing structured competition yields a strictly stronger learning signal. In particular, co-opetition-oriented adversarial interaction forces each component to actively expose the other's weaknesses, making errors observable, attributable, and thus optimizable.

Here, co-opetition characterizes a dual relationship between the retriever and the generator: they (i) *compete* to reveal each other's failure modes, thereby generating informative adversarial feedback, while (ii) *cooperate* to jointly optimize the end-to-end objective of RAG. Building on this insight, we propose Co-opetition AdveRsarial Learning (CARL), which formulates retriever–generator training as a unified minimax game. This formulation enables tighter coupling between retrieval and generation, leading to more robust collaboration and improved error correction. We next present the minimax objective, training framework, and implementation details.

3.1 Minimax Optimization Objective

RAG formulation. A standard RAG system consists of a retriever R_θ parameterized by θ, and a generator G_ϕ parameterized by ϕ. The retriever R_θ models the posterior probability of a document d being relevant to a user query q, and retrieves the top-k relevant documents from the knowledge base by ranking documents:

$$R_\theta(q, d) = p_\theta(d|q), \tag{1}$$

where $p_\theta(d|q)$ denotes the probability of d given q under the retriever's parameterization θ.

The generator G_ϕ generates an answer sequence $a = \{a_1, a_2, ..., a_N\}$ (composed of N tokens) in an autoregressive manner, conditioned on the query q and retrieved documents d. Its conditional probability is defined as:

$$G_\phi(q, d, a_{1:i-1}) = p_\phi(a_i|q, d, a_{1:i-1}), \tag{2}$$

where a_i is the i-th token of the answer, $a_{1:i-1} = \{a_1, ..., a_{i-1}\}$ denotes the prefix of the answer sequence up to the $(i-1)$-th token, and ϕ represents the learnable parameters of the generator.

Following Lewis et al. (2020), the end-to-end likelihood of the RAG system generating answer a given query q is approximated by marginalizing over the top-k documents retrieved by R_θ:

$$p_{\text{RAG}}(a|q) \approx \prod_{i}^{N} \sum_{d \in \text{top}-k(p(\cdot|q))} p_\theta(d|q) p_\phi(a_i|q, d, a_{1:i-1}), \tag{3}$$

where top-$k(R_\theta(q, \cdot))$ denotes the set of the top-k documents ranked by the retriever $R_\theta(q, d) = p_\theta(d|q)$, and $p_{\text{RAG}}(a|q)$ is the end-to-end probability of generating answer a for query q.

Minimax game for RAG. CARL reframes RAG training as a two-player minimax game between the retriever R_θ and generator G_ϕ, where adversarial pressure is introduced via hard negative documents to drive mutual improvement of both components through competitive yet cooperative interaction.

From an adversarial perspective, generators are prone to hallucinations when conditioned on noisy or misleading documents, while retrievers are inherently designed to discriminate the relevance of documents to queries. Conversely, this adversarial interaction also fosters cooperation: the retriever learns to align with the generator's evidence preferences, and the generator adapts to better exploit the retrieved content for faithful generation.

Therefore, CARL optimizes the retriever and generator with complementary objectives: the retriever is trained to retrieve query-relevant yet challenging documents that test the generator's robustness, while the generator learns to distinguish ground-truth documents from adversarially sampled negative documents. This design draws inspiration from adversarial retrieval frameworks (e.g., IRGAN (Wang et al., 2017), AR2 (Zhang et al., 2021)) and is tailored to the RAG paradigm.

Formally, the minimax objective of CARL is defined as:

$$J = \min_\theta \max_\phi \mathbb{E}_{\mathbb{D}_q^- \sim R_\theta(q,\cdot)} \left[\log P_\phi(d|q; \mathbb{D}_q)\right], \tag{4}$$

where: - $\mathbb{D}_q^- = \{d_1^-, d_2^-, ..., d_n^-\}$: the set of n hard negative (irrelevant) documents sampled from the retriever's probability distribution $R_\theta(q, \cdot)$ for a given query q; - $\mathbb{D}_q = \{d\} \cup \mathbb{D}_q^-$: the document

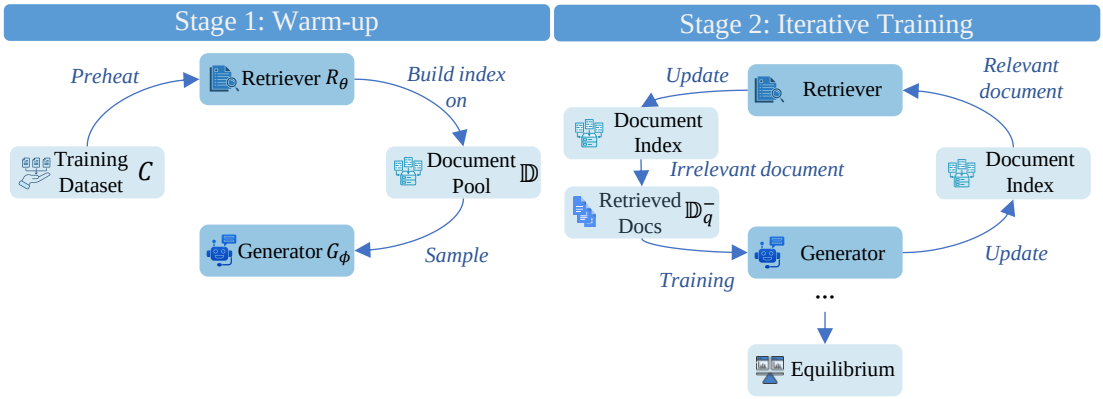

Figure 1: Diagram of CARL's training process. In the warm-up stage, the retriever and generator are initialized independently and iteratively trained to obtain stable starting points. In the adversarial iterative stage, the retriever and generator are alternately updated: (i) the retriever samples the most relevant documents to update its parameters, (ii) the document index is rebuilt, and (iii) the generator is trained on the least relevant documents to update its parameters. This alternating optimization continues until convergence.

set consisting of the ground-truth relevant document d and the hard negative set $\mathbb{D}_q^-$; - $P_\phi(d|q;\mathbb{D}_q)$: the probability that the generator G_ϕ selects the ground-truth document d from $\mathbb{D}_q$ given query q.

By optimizing this minimax objective, the retriever and generator converge to an equilibrium where retrieval relevance and generation robustness are mutually reinforced, leading to improved end-to-end performance of the RAG system. Besides, a systematic analysis of how competitive interactions mitigate component heterogeneity in RAG systems is left for future work (Section Limitations).

3.2 Training Framework

Given the minimax objective in Equation (4), we design an adversarial training framework that enables efficient retriever–generator interaction while maintaining stable optimization. CARL follows a two-stage training procedure: an independent warm-up stage followed by adversarial iterative training. During adversarial training, the retriever $R_\theta(q,\cdot)$ samples challenging documents to confuse the generator, while the generator $G_\phi(q,\cdot)$ is optimized to distinguish ground-truth documents from adversarial negatives. Figure 1 illustrates the overall training process.

Warm-up stage. We first initialize the retriever and generator independently to obtain stable starting points. (i) First, we initialize the retriever R_θ and the generator G_ϕ using an off-the-shelf retriever and LLM. We then use the training dataset C to preheat R_θ to obtain the first-generation retriever R_θ^0; (ii) then, we build the index for the query on $\mathbb{D}$, and retrieve and sample the irrelevant samples in order to preheat it to obtain the first-generation generator G_ϕ^0; (iii) Finally, we iterate the retriever and generator until training converges.

Adversarial iterative stage. Retriever and generator are then updated alternately: (i) First, we sample n documents from the index that are most relevant to the user query to train the retriever, and update its parameters to get the new generation of retriever R'_θ; (ii) then, we update the document index once during the training process for each generation of retriever training; (iii) Finally, we sample n more documents from the updated index that are least relevant to the user query, $\mathbb{D}_q^-$, to train the generator, and update its parameters to get the new generation of generator G'_ϕ. This alternating optimization continues until convergence, at which point the retriever and generator reach an approximate equilibrium.

3.3 Training Details

Document collection and relevance definition. We use the 2018 Wikipedia dump (Karpukhin et al., 2020) as the shared document pool for all tasks. Document relevance is defined by vector similarity in the embedding space: documents with the lowest similarity to a query are treated as the most irrelevant. All methods operate on the same corpus, ensuring fair comparison; performance differences arise solely from retrieval architectures rather than document availability.

Document index construction. We use the document encoder of the retriever to compute the embedding $E(d;\theta)$ of each document d in the knowledge base $d \in C$. An approximate nearest neighbor (ANN) index is then built once using FAISS (Johnson et al., 2019) with inner-product similarity.

Retriever warm-up via generator distillation. During the warm-up stage, we distill knowledge from the generator into the retriever by adding a regularization term to the retriever objective:

$$J_{\mathcal{R}} = H(P_\phi(\cdot|q;\mathbb{D}), P_\theta(\cdot|q;\mathbb{D})), \quad (5)$$

where $J_{\mathcal{R}}$ denotes the regularization term; $H(\cdot)$ refers to the cross-entropy function; $P_\phi(\cdot|q;\mathbb{D})$ is the generator-induced document distribution; $P_\theta(\cdot|q;\mathbb{D})$ is the retriever's predicted distribution over the document pool.

In practice, optimization is performed on a restricted candidate set $\mathbb{D}_q = \{d\} \cup \mathbb{D}_q^-$ (Section 3.1), yielding the following equivalent form:

$$J'_{\mathcal{R}} = H(P_\phi(\cdot|q;\mathbb{D}_q), P_\theta(\cdot|q;\mathbb{D}_q)). \quad (6)$$

which is the regularization term adopted in CARL.

4 Experiment Settings

Datasets. Following (Izacard et al., 2023), we evaluate CARL on four representative NLP tasks: open-domain QA, fact verification, slot filling, and open-domain dialogue, using a total of seven widely adopted datasets. Specifically, (i) The open-domain QA task, including **NQ** (Kwiatkowski et al., 2019), **HotpotQA** (Yang et al., 2018), and **TriviaQA** (Joshi et al., 2017), focuses on directly generating answers from unstructured documents; (ii) The fact verification task assesses the veracity of claims based on retrieved evidence, we adopt **FEVER** (Thorne et al., 2018); (iii) The slot filling task involves extracting slot values for predefined attributes of a given entity from large-scale corpora, we use **T-REx** (Elsahar et al., 2018) and **zsRE** (Levy et al., 2017); and (iv) For open-domain dialogue, which engages users in free-form conversations grounded in external knowledge sources, we employ **WOW** (Dinan et al., 2018).

Evaluation metrics. For open-domain QA (Izacard et al., 2023), we adopt the Exact Match (**EM**) score (Fabian, 2011), which measures the percentage of predictions that exactly match the ground-truth answer in content, order, and spelling. For fact verification and slot filling, we use accuracy (**Acc**), which measures the proportion of correctly classified samples. For dialogue, we use the **F1** score (Manning et al., 2008), the harmonic mean of precision and recall, to assess response quality.

Baselines. We introduce four types of baselines for comparison with our method: (i) Zero-shot LLM: Uses LLMs to generate the answer directly, without retrieval and training, including Qwen2-7B (Bai et al., 2023), Llama3-8B (Touvron et al., 2023), and DeepSeek's distillation model DeepSeek-R1-Distill-Qwen-7B (DeepSeek-R1 for short) (Bi et al., 2024). (ii) Independent RAG: Trains the retriever and generator separately, or trains only one of them. Specifically, AAR (Yu et al., 2023), RePlug (Shi et al., 2024), and SCARLet (Xu et al., 2025) only train retriever, while RAG (Lewis et al., 2020), IUM (Salemi and Zamani, 2025), RALM (Ram et al., 2023), and Re^{2}G (Glass et al., 2022) train the components of RAG separately. (iii) Joint RAG: Jointly optimizes the retriever and generator of RAG, including FiD (Hofstätter et al., 2023), SRAG (Zamani and Bendersky, 2024), RbFT (Tu et al., 2025), MINT (Tang et al., 2025), and ATLAS (Izacard et al., 2023). (iv) Adversarial RAG: We adopt RAAT (Fang et al., 2024), which adversarially trains the retriever and generator in an online data augmented manner to improve the noise robustness.

For fair comparison, we replaced the retrieval and generation models used by most methods in Independent RAG, Joint RAG, and Adversarial RAG (except RALM (Ram et al., 2023)) with the same model as our method and conducted the experiment again. For RALM (Ram et al., 2023) , we did not modify its generative model due to its use of a larger parameter generative model.

Implementation details. For CARL training, we use Adam (learning rate=5e-6) with a convergence threshold of 0.1 and a temperature of 0.7. The training data is divided into small batches of 4 sets, limiting the longest input to 512. Following (Gu and Qin, 2024), CARL employs Qwen2-7B (Bai et al., 2023) for generation and is built upon the standard RAG framework (Lewis et al., 2020). The experiments are deployed on a single Nvidia A800 GPU 80GB. For the warm-up stage, CARL is initialized with Qwen2-7B (Bai et al., 2023) as the base generator model and DPR (Karpukhin et al., 2020) as the base retrieval model. The retriever uses the 2018 Wikipedia dump (Karpukhin et al., 2020) as the knowledge source.

For fair comparison, we follow Lin et al. (2024) and set the number of retrieved passages to 3 across all retrieval-based methods. It is worth noting that the RAG method REALM (Guu et al., 2020) points out that the higher the frequency of index update of retrieved documents, the better the performance

Methods	Open-Domain QA			Fact.	Slot Filling		Dial.	Avg.
	NQ EM	**HotpotQA** EM	**TriviaQA** EM	**FEVER** Acc.	**T-REx** Acc.	**zsRE** Acc.	**WOW** F1	
Zero-shot LLM								
Qwen2-7B	22.1	23.1	60.9	36.7	62.4	49.8	19.8	39.3
Llama3-8B	27.8	28.6	66.0	40.8	65.1	54.4	22.5	43.6
DeepSeek-R1	30.0	29.7	67.1	43.5	65.2	56.5	24.0	45.1
Independent RAG								
AAR	34.1	29.7	-	66.6	28.7	-	10.1	-
RePlug	33.7	34.0	-	71.4	26.9	-	12.2	-
SCARLet	38.2	35.4	-	74.3	29.7	-	12.6	-
RAG	44.2	43.0	67.9	73.3	76.6	72.3	26.1	57.6
IUM	42.5	35.5	70.3	**86.9**	**88.9**	62.4	-	-
RALM	44.4	46.5	74.6	76.4	76.9	72.6	27.1	59.8
$\mathrm{Re}^2\mathrm{G}$	44.9	48.1	70.0	80.8	78.4	75.2	24.0	60.2
Joint RAG								
FiD	45.6	25.6	57.6	80.6	76.0	81.1	11.9	54.2
SRAG	46.0	33.1	64.7	84.8	78.3	**87.1**	19.2	59.0
RbFT	45.4	48.5	76.8	79.9	77.0	73.1	21.9	60.4
MINT	<u>46.5</u>	44.1	70.8	81.1	78.8	76.1	<u>28.0</u>	60.8
ATLAS	45.4	48.6	76.6	79.9	78.1	74.8	27.6	61.6
Adversarial RAG								
RAAT	46.3	48.8	76.9	80.1	78.5	75.9	27.8	62.0
Our Method								
CARL w/o ret.	45.1	47.9	75.0	70.0	76.8	72.7	27.7	59.3
CARL w/o reg.	46.3	<u>49.2</u>	<u>77.4</u>	81.6	79.7	76.0	27.8	<u>62.6</u>
CARL	**47.6***	**50.2***	**78.4***	<u>84.9</u>	<u>80.7</u>	<u>81.7</u>	**29.1***	**64.7***

Table 1: A comparative analysis of different methods across different NLP tasks. Best scores are in bold and second-best are underlined. * indicates statistically significant improvements over all the baselines (p-value ≤ 0.05).

of the retrieval enhancement generation system. Therefore, an attempt should be made to update the document index once during the training process for each generation of retriever training. Our CARL method also follows this recommendation.

5 Experimental Results

Our experiments aim to answer two research questions: (RQ1): How does CARL perform compared to state-of-the-art RAG methods? We address this in Section 5.1. (RQ2): How do factors such as the model scale, retriever, LLM, and number of iterations impact CARL's performance? We analyze these in Sections 5.2, 5.3, 5.4, and 5.5 respectively.

5.1 Main Results

Overall performance. The results of the performance comparison between the CARL method and the baseline methods are shown in Table 1. We can observe that: (i) LLMs exhibit a substantial performance gap compared to RAG methods, particularly in fact verification and slot filling tasks. This highlights that unfine-tuned zero-shot LLMs lack external knowledge, whereas RAG methods, especially adversarial RAG variants (RAAT), effectively leverage in-context knowledge from the retrieval module to improve outputs. (ii) Judging from the first three lines in Table 1, small-parameter LLMs distilled from larger models (DeepSeek-R1-Distill-Qwen-7B) outperform LLMs of comparable size (Qwen2-7B and Llama3-8B) across multiple tasks, demonstrating the strength of knowledge distillation. This observation also motivates the design of regularization terms in CARL. (iii) Independent RAG methods, such as RALM, show limited improvements over

the original RAG approach, while the joint training method ATLAS has improved by up to 8.7% compared to the RAG method. This underscores the benefits of interactive component training within RAG systems.

When we look at our proposed model CARL, we find that: (i) CARL achieves an absolute gain +1.5∼5 points over the jointly-trained RAG method ATLAS, with average scores across all tasks exceeding all baselines. It ranks first on 4 of 7 datasets, outperforming the second-best method by 1.3-3.9%. These results suggest that adversarial training provides richer guidance than standard joint training, enabling more effective retriever–generator interaction. (ii) CARL significantly outperforms the prior adversarial method RAAT, achieving an absolute gain +2.7 points on the average score of all the 7 datasets. RAAT primarily targets robustness to retrieval noise rather than general task performance. In contrast, CARL explicitly guides both retriever and generator to generate mutually beneficial signals, achieving deeper interaction and better adaptation to task-specific data.

Ablation study. We further investigate the contributions of the retriever and the regularization term in CARL. As shown in Table 1, *w/o reg.* denotes a variant without the regularization term (cross-entropy), and *w/o ret.* indicates a configuration without the retriever. Observations include: (i) Removing the regularization term leads to performance drops across all tasks, demonstrating that training regularization effectively helps leverage external knowledge and learn from each other. (ii) The fact verification task suffers the largest decline without regularization, highlighting its strong dependence on retriever guidance. Our regularization effectively distills knowledge into the retriever, boosting its capability. (iii) Removing the retriever reduces performance on open-domain QA and slot filling to naive RAG levels, emphasizing the critical role of the adversarially trained retriever in fetching relevant documents. Dialogue tasks are less affected, likely due to stronger reliance on the generator than on the retriever.

In the following, we conduct detailed analyses on one representative dataset per task: NQ for open-domain QA, FEVER for fact verification, zsRE for slot filling, and WOW for dialogue.

5.2 Scaling Model Size

In RAG systems, the generation step typically dominates latency, making the choice of LLM size a critical trade-off between performance and efficiency. To evaluate the robustness of CARL under varying Qwen2 model scales, we compare its performance across generative models of different parameter sizes on four representative datasets.

As shown in Table 3, we find that: We can observe that: (i) Performance generally degrades as model size decreases, with the largest drop in slot filling, followed by fact verification, and the smallest impact on dialogue. This is consistent with reduced knowledge and reasoning capacity in smaller models. Notably, dialogue performance remains relatively stable, and CARL consistently yields strong gains, maintaining robustness even with smaller models. (ii) Despite using the smallest generative model, CARL outperforms most baselines across multiple tasks (Table 1), indicating that adversarially trained RAG systems can achieve superior performance with smaller LLMs, improving efficiency and practicality. (iii) Overall, CARL exhibits strong stability across model scales. Its adversarial interaction effectively mitigates performance degradation under model scaling, demonstrating robust and scalable effectiveness.

5.3 Impact of Different LLMs

Some baseline methods use generators different from our default Qwen2-7B model. To enable a fair comparison, we replace CARL's generator with the models used by ATLAS (T5-base-lm-adapt) and RAAT (Llama2-7B) and evaluate performance against the original baseline results.

As shown in Table 2, we find: (i) As expected, performance varies across different LLMs, with Qwen2-7B consistently achieving the highest scores due to its stronger pre-trained inference capabilities. This confirms the impact of generator backbone choice on absolute performance. (ii) Importantly, CARL maintains stable and superior performance even when adopting the backbone generators of baselines. Specifically, it improves over ATLAS and RAAT across most tasks, demonstrating the robustness and universality of our adversarial training framework. This stability arises because CARL explicitly fosters interaction between the retriever and generator, allowing the system to adapt and produce reliable outputs regardless of the underlying LLM.

Generators	Methods	Open-Domain QA	Fact.	Slot Filling	Dial.	Avg.
		NQ EM	**FEVER** Acc.	**zsRE** Acc.	**WOW** F1	
T5-base-lm-adapt	ATLAS	43.6	76.9	73.6	15.5	52.4
	CARL	46.6	81.7	75.1	27.8	57.8
Llama2-7B	RAAT	45.4	78.7	75.6	27.1	56.7
	CARL	46.0	82.0	76.5	28.8	58.3
Qwen2-7B	**CARL(Origin)**	**47.6**	**84.9**	**81.7**	**29.1**	**60.8**

Table 2: Performance of CARL and baselines across various LLMs. For each specific LLM backbone, CARL significantly outperforms the corresponding baseline (p-value ≤ 0.05).

Generators	QA	Fact.	Slot.	Dial.	Avg.
	NQ EM	**FEVER** Acc.	**zsRE** Acc.	**WOW** F1	
0.5B	46.0	81.2	75.7	28.0	57.7
1.5B	46.2	81.5	75.9	28.0	57.9
7B(Origin)	**47.6**	**84.9**	**81.7**	**29.1**	**60.8**

Table 3: Comparison of CARL's performance using Qwen2 LLMs of varying sizes.

5.4 Impact of Different Retrievers

In RAG systems, the retriever plays a crucial role as the first step, providing the LLM with relevant documents that directly affect downstream generation quality. To evaluate the robustness of CARL across retrieval backbones, we experiment with three different retrievers: E5 (Wang et al., 2022), co-condenser-wiki (Gao and Callan, 2022), and the original DPR.

Retrievers	QA	Fact.	Slot.	Dial.	Avg.
	NQ EM	**FEVER** Acc.	**zsRE** Acc.	**WOW** F1	
E5	45.3	80.1	76.0	27.4	57.2
co-condenser-wiki	**47.6**	82.0	75.4	27.7	58.2
DPR(Origin)	**47.6**	**84.9**	**81.7**	**29.1**	**60.8**

Table 4: Comparison of CARL's performance across various retrievers.

The results are summarized in Table 4. Key observations include: (i) The quality of the retriever significantly impacts overall RAG performance. Stronger or more sophisticated retrievers can fetch more relevant documents, which translates into notable improvements in downstream tasks, particularly in fact verification and slot filling. (ii) CARL maintains stable and competitive performance across all retriever choices. Even when using weaker retrievers such as E5, CARL still achieves reasonable task performance, and it consistently outperforms most baseline methods. This demonstrates that the adversarially trained interaction between retriever and generator enhances the system's adaptability, making CARL robust to variations in retriever quality and architecture.

5.5 Impact of the Number of Iterations

The number of adversarial training iterations is critical for balancing effectiveness and stability. We evaluate CARL on four representative datasets with varying iteration budgets, as shown in Figure 2. We observe the following trends: (i) Across all datasets, CARL achieves strong performance in the first iteration, already outperforming most baselines in Table 1, demonstrating the effectiveness of the warm-up stage in providing a well-aligned initialization. (ii) In general, increasing the number of iterations leads to steady performance improvements, although the marginal gains vary significantly across tasks. Most datasets exhibit performance saturation around the 10th iteration, while the dialogue task (WOW) converges noticeably faster. This suggests that short-term adversarial learning is particularly effective for dialogue tasks, likely because their retrieval and generation distributions are easier to align under adversarial feedback. (iii) On FEVER, zsRE, and WOW, performance slightly degrades after reaching the peak, indicating that adversarial training is not monotonically beneficial. Excessive iterations may over-optimize adversarial patterns, weakening the robustness–generalization balance.

6 Conclusion

We present CARL, a co-opetition adversarial learning framework for retrieval-augmented generation, which explicitly aligns heterogeneous re-

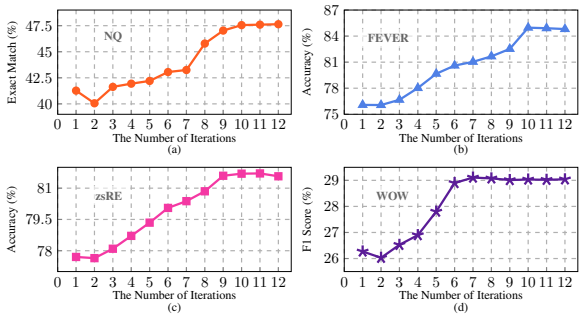

Figure 2: CARL's performance of different number of iterations.

trievers and generators through adversarial interaction. By exposing complementary failure modes and enabling fine-grained, bidirectional adaptation, CARL improves evidence selection and reduces generative hallucination. Extensive experiments on seven benchmark datasets across four NLP tasks demonstrate that CARL consistently outperforms state-of-the-art RAG methods, including jointly trained baselines, and remains effective across different retrievers, generators, and model scales. These results highlight the effectiveness and generalizability of adversarial retriever–generator training as a principled approach for enhancing RAG systems.

Limitations

(i) Although CARL is designed as an effective adversarial training framework for RAG systems, our evaluation primarily relies on seven representative NLP task datasets. These datasets mainly cover classical retrieval and generation scenarios, and do not sufficiently reflect the challenges of advanced intelligent agent settings or high-difficulty scientific reasoning tasks. In future work, we plan to extend the evaluation to more challenging and recently proposed benchmarks, to better assess the scalability and generality of adversarial training in RAG systems. (ii) For overall performance comparison, we aggregate results across datasets using weighted averaging, which may obscure fine–grained differences across evaluation metrics and task types. A more principled aggregation strategy that accounts for metric heterogeneity, as well as additional RAG-specific evaluation indicators, remains an open research direction. (iii) This work focuses on a general adversarial training paradigm for RAG systems. Although CARL outperforms most selected baselines, we do not disentangle the effects of adversarial alignment from cooperative or robustness-oriented alignment objectives studied in prior work. Future ablation studies comparing different alignment targets are needed to isolate the sources of performance gains. (iv) CARL incorporates several optimization strategies, including customized preheating procedures and knowledge distillation. However, the individual contributions and theoretical guarantees of these components have not been fully validated. Their impact on training stability and robustness warrants more systematic investigation. (v) Finally, while CARL is motivated by the hypothesis that competitive interactions can better mitigate component heterogeneity in RAG systems, we do not provide sufficient theoretical or empirical evidence to conclusively support this claim. Future work will aim to analyze what retrievers and generators learn through adversarial interactions, and clarify when and why competitive training is particularly beneficial.

Ethical Considerations

This article is aimed at the problem that the training effect of RAG systems used in the real world is not significantly improved. To mitigate this issue, we propose CARL, an iterative adversarial learning framework in which the retriever and generator are trained to expose and respond to each other's weaknesses. With respect to data usage, all experiments are conducted exclusively on publicly available academic benchmarks, including Natural Questions, FEVER, as well as a standard Wikipedia dump. No private, proprietary, or sensitive data is used at any stage of data collection, training, or evaluation. Furthermore, the proposed method does not introduce or expose any new personal information. As a result, this study does not raise concerns related to privacy, data misuse, or unauthorized access, and complies with common ethical standards for academic research in information retrieval and natural language processing.

Acknowledgements

This work was funded by the Strategic Priority Research Program of the CAS under Grant No. XDB0680102, the National Natural Science Foundation of China under Grants No. 62472408, U25B2076 and 62441229, and the National Key Research and Development Program of China under Grants No. 2023YFA1011602.

All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, and 1 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Adam M Brandenburger and Barry J Nalebuff. 2011. *Co-opetition*. Crown Currency.

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

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Pedregosa Fabian. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research 12*, page 2825.

Feiteng Fang, Yuelin Bai, Shiwen Ni, Min Yang, Xiaojun Chen, and Ruifeng Xu. 2024. Enhancing noise robustness of retrieval-augmented language models with adaptive adversarial training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10028–10039.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Jiakai Gu and Donghong Qin. 2024. An arxiv paper question-answering system based on qwen and rag. In *2024 6th International Conference on Frontier Technologies of Information and Computer (ICFTIC)*, pages 1354–1361. IEEE.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Sebastian Hofstätter, Jiecao Chen, Karthik Raman, and Hamed Zamani. 2023. Fid-light: Efficient and effective retrieval-augmented text generation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1437–1447.

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

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: A

benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *21st Conference on Computational Natural Language Learning, CoNLL 2017*, pages 333–342. Association for Computational Linguistics (ACL).

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

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, and 1 others. 2024. Ra-dit: Retrieval-augmented dual instruction tuning. In *ICLR*.

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. Webglm: towards an efficient web-enhanced question answering system with human preferences. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 4549–4560.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge university press.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Andrew Ross and Finale Doshi-Velez. 2018. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Alireza Salemi and Hamed Zamani. 2025. Learning to rank for multiple retrieval-augmented models through iterative utility maximization. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, pages 183–193.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. Replug: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8364–8377.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Yubao Tang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2025. Boosting retrieval-augmented generation with generation-augmented retrieval: A co-training approach. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2441–2451.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yiteng Tu, Weihang Su, Yujia Zhou, Yiqun Liu, and Qingyao Ai. 2025. Rbft: Robust fine-tuning for retrieval-augmented generation against retrieval defects. *arXiv preprint arXiv:2501.18365*.

Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 515–524.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.

Yilong Xu, Jinhua Gao, Xiaoming Yu, Yuanhai Xue, Baolong Bi, Huawei Shen, and Xueqi Cheng. 2025. Training a utility-based retriever through shared context attribution for retrieval-augmented language models. *arXiv preprint arXiv:2504.00573*.

Task	Dataset	Train Size	Dev. Size	Test Size
Open-Domain QA	**NQ** (Kwiatkowski et al., 2019)	307,372	7,830	7,842
	HotpotQA (Yang et al., 2018)	90,447	7,405	7,405
	TriviaQA (Joshi et al., 2017)	138,384	18,669	17,210
Fact Verification	**FEVER** (Thorne et al., 2018)	311,431	37,566	19,998
Slot Filling	**T-REx** (Elsahar et al., 2018)	1,274,264	318,566	122
	zsRE (Levy et al., 2017)	163,196	19,086	19,086
Dialogue	**WOW** (Dinan et al., 2018)	18,430	1,948	1,933

Table 5: Datasets statistics of our benchmark datasets.

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

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023. Augmentation-adapted retriever improves generalization of language models as generic plug-in. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2421–2436.

Hamed Zamani and Michael Bendersky. 2024. Stochastic rag: End-to-end retrieval-augmented generation through expected utility maximization. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2641–2646.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2021. Adversarial retriever-ranker for dense text retrieval. *arXiv preprint arXiv:2110.03611*.

Hengran Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2023. From relevance to utility: Evidence retrieval with feedback for fact verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6373–6384.

Mosad Zineldin. 2004. Co-opetition: the organisation of the future. *Marketing intelligence & planning*, 22(7):780–790.

A Details of Datasets

The statistical details of datasets used in this paper are shown in Table 5. The 'Dev. Size' column in the table means the size of validation set.

B Prompts

In this section, we will provide prompts used by the generative model in CARL. Listing 1 shows the prompts of the generator in CARL when answering QA tasks, and Listing 2 provides reference documentation for this task.

```
prompt='<<SYS>>\n{system_prompt}\n<</SYS
    >>\n\n[INST] {query} [/INST]'

sys = 'You need to complete the question
    -and-answer pair following the
    format provided in the example. The
    answers should be short phrases or
    entities, not full sentences. Here
    are some examples to guide you.'

examples = [
            '\nExample 1:\nQuestion:
    What is the capital of France?\
    nAnswer: Paris.',
            '\nExample 2:\nQuestion: Who
     invented the telephone?\nAnswer:
    Alexander Graham Bell.',
            '\nExample 3:\nQuestion:
    Which element has the atomic number
    1?\nAnswer: Hydrogen.'
        ]
```

Listing 1: The prompt used when generating the model to answer questions.

```
sys_ctx = 'The following contexts will
    help you complete the question-and-
    answer pair.\nContext1{ctx_0}\
    nContext2{ctx_1}\nContext1{ctx_2}'.
    format(ctx_0=ctx_list[0]['text'],
    ctx_1=ctx_list[1]['text'],ctx_2=
    ctx_list[2]['text'])
```

Listing 2: The prompt used when providing documents to help generate models to answer questions.

C Cost Tradeoff of CARL

To clarify efficiency, we report the average wall-clock time per iteration in Table 6. As shown, index rebuilding is the dominant component (∼2090 s), while retriever and generator optimization are comparatively lightweight.

Despite this per-iteration cost, CARL converges in only 7–11 iterations across tasks, resulting in

Time (s)	QA	Fact.	Slot.	Dial.	Avg.
	NQ	**FEVER**	**zsRE**	**WOW**	
Generator training	304.6	281.5	260.9	364.2	302.8
Retriever training	190.7	181.8	175.0	198.9	186.6
Index update	**2096.4**	**2082.1**	**2085.7**	**2098.3**	**2090.6**
Total	2611.7	2565.4	2540.2	2681.0	2599.6

Table 6: Average wall-clock time per iteration.

total training times of 5.2–7.8 hours (avg. 6.8 h) on a single A800 GPU. Thus, the overall training cost remains moderate even for Wikipedia-scale corpora.

Importantly, index rebuilding is required in CARL because the retriever distribution changes after each update; refreshing the index ensures that adversarial negatives are sampled from the current model state rather than a stale embedding space. This step is therefore essential for maintaining the validity of the minimax training signal rather than an implementation artifact.

We note that CARL already operates within practical time budgets, and further optimizations such as incremental indexing, asynchronous updates, partial corpus re-indexing, or periodic refresh can reduce cost without changing the algorithmic design.

Furthermore, we compared total training time with representative strong baselines from independent and joint RAG paradigms in Table 7.

Time (h)	QA	Fact.	Slot.	Dial.	Avg.
	NQ	**FEVER**	**zsRE**	**WOW**	
Re^2G	5.28	4.46	9.52	4.21	5.87
ATLAS	1.57	1.61	0.85	0.94	1.24
CARL	**7.25**	**7.13**	**7.76**	**5.21**	**6.84**

Table 7: Total training time of representative strong baselines from independent and joint RAG paradigms.

The results show that CARL requires longer training than lightweight joint methods (e.g., ATLAS), primarily due to index refresh operations, but remains within the same practical time scale as full retriever–generator training approaches.

This difference reflects a deliberate trade-off: CARL performs additional adversarial interaction steps to improve retriever–generator alignment. While this increases runtime, it yields significant performance gains across datasets (Avg. column of Table 1), achieving absolute average improvements of +4.5 points over Re^2G and +3.1 points over ATLAS. We therefore interpret CARL's cost–effectiveness as favorable: the additional computation scales linearly with the number of iterations, while performance gains remain stable across tasks and model settings.

Importantly, there are several concrete directions to further reduce training cost without modifying the core algorithm: (i) Incremental index updates: only re-embedding documents whose similarity scores change significantly rather than rebuilding the full index; (ii) Stale-index training: updating the index every k iterations instead of each iteration; (iii) Asynchronous indexing: rebuilding the index in parallel with generator optimization; (iv) Partial corpus refresh: updating only top-candidate regions of the embedding space.

These optimizations are orthogonal to CARL's design and can be integrated directly.

D Setting of Statistical Tests

For the reported significance in Table 1, we used a paired two-tailed t-test over multiple random seeds for each dataset to compare CARL against each baseline. This test accounts for the dependency between repeated runs and is standard practice in NLP evaluation. No additional multiple-comparison correction was applied because the comparisons are independent pairwise evaluations against each baseline.

Across all datasets, the reported improvements are statistically significant at $p \leq 0.05$, supporting the claim of superiority over all baselines.While multiple pairwise comparisons are performed, each comparison is independent. Following common NLP practice, pairwise significance is typically reported without correction.We have conducted Bonferroni-corrected tests to account for multiple comparisons.Let m denote the number of baselines compared per dataset; the corrected significance threshold is α/m, where $\alpha = 0.05$. All reported improvements remain statistically significant under this stricter criterion, confirming that our results are robust. Together with the paired t-test described above, this supports the reliability of our claims.

E Supplementary Explanation on Experimental Setup

The performance differences mainly stem from variations in experimental settings, evaluation protocols, and model scales. We clarify these below.

Dataset source.

(i) For the NQ dataset, it can be obtained from https://ai.google.com/research/

NaturalQuestions/download.

(ii) For the HotpotQA dataset, it can be obtained from https://hotpotqa.github.io/.

(iii) For the TriviaQA dataset, it can be obtained from https://nlp.cs.washington.edu/triviaqa/.

(iv) For the FEVER dataset, it can be obtained from https://fever.ai/dataset/fever.html.

(v) For the T-REx dataset, it can be obtained from https://hadyelsahar.github.io/t-rex/downloads/.

(vi) For the zsRE dataset, it can be obtained from https://drive.google.com/file/d/1TMYvAbe9wsB5GiWcUL5bMAs9x6CpvnAj/view.

(vii) For the WOW dataset, it can be obtained from https://parl.ai/projects/wizard_of_wikipedia/.

Setting of NQ. Our experiments use a few-shot setting (k = 64) with a base-scale generator. Under this controlled setup, CARL surpasses other RAG methods using comparable parameter sizes. Some reported leaderboard results employ substantially larger models (e.g., 70B-scale LLMs) or more shots, which are not directly comparable.

Setting of HotPotQA. Our reported scores follow the distractor setting and use Joint EM, consistent with standard RAG evaluation. The only leaderboard method exceeding our score (Beam Retrieval) is a retriever-only approach rather than a full RAG system, so it falls outside our comparison scope. Among full RAG methods under the same setting, our approach achieves the best performance.

Setting of TriviaQA. We restrict retrieval strictly to the Wikipedia domain and report Full-EM, a stricter metric. Some higher public results rely on very large proprietary or 70B-scale models. Our comparisons instead focus on methods with similar model scale, where CARL shows consistent advantages.

What "CARL w/o ret." means. This variant does not remove retrieval or reduce to a standalone LLM. Instead, it removes the trainable retriever by fixing it as the original DPR and disabling any retriever updates during training. The system still performs retrieval from the same corpus with the same top-k setting, and the generator is still conditioned on retrieved documents. However, the adversarial co-training loop is partially disabled: the retriever no longer adapts or produces dynamically challenging negatives, and only the generator is optimized.

Motivation of "CARL w/o ret.". CARL is designed to improve performance through competitive interaction between retriever and generator. By freezing the retriever (original DPR), we isolate the effect of this interaction. This ablation tests whether improvements come merely from generator-side training or specifically from adversarial retriever–generator co-adaptation.

Interpretation of "CARL w/o ret." results. The performance drop in "CARL w/o ret." setting, particularly on open-domain QA and slot filling, shows that CARL's gains are not explained by generator training alone. Instead, they depend on the retriever's ability to iteratively refine its ranking and supply informative hard negatives.

Overall, the key point is that our experiments control for model size, retrieval corpus, evaluation metric, and shot number to ensure fair comparison. When these factors are aligned, CARL consistently outperforms prior RAG approaches.

F Case Study

We agree that averaged scores alone may hide task-specific behaviors, and qualitative analysis can better reveal when and why CARL is effective or limited. In the appendix, we added a dedicated case study section illustrating strengths, weaknesses, and behavioral differences.

F.1 Fine-grained Performance is Already Observable

Table 1 reports per-dataset metrics in addition to the weighted average, allowing readers to inspect task-level variation. The gains of CARL are consistent across datasets rather than driven by a single task, indicating that improvements are systematic rather than metric-aggregation artifacts.

F.2 Strength Case: Improved Evidence Discrimination

In the example shown in Table 8, the independent RAG baseline retrieves loosely related passages and composes an incorrect answer, while CARL retrieves documents explicitly mentioning the correct film and answers correctly. This reflects CARL's

adversarial objective: the generator is trained to distinguish useful from misleading evidence, which sharpens document selection boundaries and reduces reliance on spurious correlations.

Question: River Phoenix died during the making of which movie?
Documents by RAG: Further Phoenix at Rio's Attic: I Love You To Death William Richert ... River Phoenix as Devo Nod. ... it would be during the making of this movie that they would ...
Response of RAG: Devo Nod
Documents by CARL: River Phoenix - A number of ... 10 Stars Who Died During the Filming of a Movie. ... by Me,ẅas near the end of filming D̈ark Bloodẅhen he died of a drug ...
Response of CARL: Dark Blood
Answer: Dark Blood

Table 8: Strength case of CARL.

F.3 Failure Case: Adversarial Negatives Can Mislead

In the example shown in Table 9, CARL is misled by a generated negative document that is topically relevant but factually incorrect, leading to a wrong prediction. This suggests a limitation of adversarial training: if negative samples are overly plausible yet incorrect, the generator may overfit their distribution rather than learning robust evidence validation. Future improvement: we plan to introduce confidence-aware filtering or calibration on generated negatives to reduce this failure mode.

Question: In which sitcom did Penelope Wilton play the wife of Richard Briers?
Negative Documents by CARL: Richard Briers, The Good Life ... In 1970s BBC sitcom The Good Life, Briers and Felicity Kendal played ... Penelope Wilton, Richard Briers Briers appeared with Peter ...
Response of CARL: The Good Life
Answer: Ever Decreasing Circles

Table 9: Failure case of CARL.

F.4 Behavioral Comparison with Joint Training Methods

In the example shown in Table 10, both ATLAS and CARL answer correctly, but CARL retrieves multiple supporting documents that collectively provide structured evidence, while ATLAS relies on a single passage. This indicates that CARL's competitive interaction encourages retrieval diversity and complementary evidence coverage rather than single-document reliance. Such behavior is consistent with our design goal of exposing complementary failure modes between retriever and generator.

Question: What substance, best known as a poison, was used in small doses in medications as a stimulant, as a laxative, and for enhancing performance in sports?
Documents by ATLAS: Although it is best known as a poison, small doses of strychnine were ... historically for enhancing performance in sports. ... is the most complex substance known.
Response of ATLAS: strychnine
Documents by CARL: (1) Although it is best known as a poison, small doses of strychnine were once used in medicine as a stimulant, as a laxative, ... enhancing performance in sports. (2) ... it was recorded in dictionary.sensagent.com/c21h22n2o2/en-en ... (3) Strychnidin-10-one is best known for its stimulant, ... Strychnidin-10-one's ability to improve physical performance is well known among well ...
Response of CARL: Strychnine sulfate (Strychnidin-10-one, C21H22N2O2)
Answer: Strychnine sulfate

Table 10: Behavioral comparison between CARL and ATLAS.

G Supplementary Explanation on Component-wise Contributions

First, the key components of CARL (adversarial sampling and pseudo-document injection) are jointly embedded in a minimax training loop where the retriever and generator update each other iteratively. Because each component changes the optimization target of the other, removing one module alters the training dynamics rather than cleanly isolating its effect. This makes standard drop-one ablations less informative for this class of coupled adversarial frameworks.

Instead, we performed component-level proxy evaluations to assess whether each module independently improves after CARL training, i.e., generator quality and retriever quality.

(i) Importantly, we observe consistent improvements across all evaluated datasets for both

modules after CARL training. Here we report NQ as a representative example.

(ii) For the generator, standalone evaluation without retrieval shows that EM improves from 22.1 (base Qwen2-7B) to 33.9 after CARL training, an absolute gain of +11.8 points, indicating that adversarial training enhances intrinsic reasoning ability rather than only benefiting joint inference.

(iii) For the retriever, following the standard DPR evaluation protocol, Top-20 accuracy increases from 79.4 to 82.5 (+3.1) and Top-100 from 86.0 to 87.1 (+1.1). The larger gain at lower recall depths suggests that CARL particularly improves early precision, which is most influential for downstream generation.

(iv) These results demonstrate that both components benefit individually from CARL training, supporting that performance gains are not solely due to joint inference but arise from improved module capability.