INVAR-RAG: INVARIANT LLM-ALIGNED RETRIEVAL FOR BETTER GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation (RAG) has shown its impressive capability of providing reliable answer predictions and addressing severe hallucination problems. A typical RAG implementation adopts powerful retrieval models to extract external information and leverage large language models (LLMs) to generate corresponding answers. Different with that, recent LLM-based retrieval has raised much attention because it brings substantial improvements in information retrieval (IR) via LLMs' vigorous semantic understanding capability. However, directly applying LLM to RAG systems remains certain challenges. This may cause feature locality problems since massive parametric knowledge impedes the effective usage of the global information among all corpus, *e.g.,* a LLM-based retriever usually inputs the summary of documents instead of the whole documents. Moreover, various tasks pre-trained in LLMs induce severe variance, which further weakens its performance as the retriever. To address these issues, we propose a novel two-stage fine-tuning architecture called Invar-RAG. In the retrieval stage, a LLM-based retriever is constructed by integrating a LoRA-based representation learning to address the feature locality problem. To justify and consolidate this retrieval's performance, two patterns (*i.e.,* invariant and variant patterns) and an invariance loss are also developed to alleviate the variance in LLM. Moreover, in the generation stage, a meticulously designed fine-tuning method is devised to improve our LLM for accurate answer generation based on the retrieved information. Experimental results demonstrate that Invar-RAG significantly outperforms existing baselines across three Open-domain Question Answering (ODQA) datasets. The code is available in **Supplementary Material** to ease reproducibility.

1 INTRODUCTION

Over the past decade, large language models (LLMs) have demonstrated promising capability in processing natural language Minaee et al. (2024). Owing to the vast amount of knowledge encoded in their internal parameters, LLMs such as GPT Achiam et al. (2023) and LLaMa Touvron et al. (2023) have demonstrated remarkable performance on various downstream tasks, including Open-domain Question Answering (ODQA) Zhu et al. (2021), Reading Comprehension Cheng et al. (2023). However, the fixed parametric knowledge of LLMs has hindered the further applications of LLMs and made them prone to errors (hallucination Rawte et al. (2023) and factual errors Wang et al. (2023)).

To overcome the limitations of parametric knowledge, one promising approach is Retrieval-Augmented Generation (RAG) Wang et al. (2023); Lewis et al. (2020). Compared to relying solely on parametric knowledge, RAG enables LLMs to use retrievers to access relevant information from external knowledge sources, enhancing their question-answering abilities. Among the two components of RAG, current methods primarily focus on optimizing the retriever to return more relevant documents due to the high cost of fine-tuning and black-box LLM APIs. Previous retrievers leveraged deep learning technology (*e.g.,* dense retrieval Zhao et al. (2024)) to encode the text representations from the lexical space into the high-dimensional latent space, allowing them to model more complex semantic relationships between queries and corpora. However, the separation between the retriever and generation model has hindered their full integration, limiting their compatibility in downstream applications . Some advanced RAG systems, such as RA-DIT Lin et al. (2023b), have adopted joint training mechanisms that fine-tune both the retriever and the generation model for

better alignment. However, this approach is impractical due to the need for frequent fine-tuning and fails to utilize the LLMs' semantic understanding capabilities during the retrieval stage.

Consequently, generative retrieval (GR), also known as LLM-based retrieval, leverages the parametric memory of generative models to directly generate document identifiers (DocIDs) Li et al. (2024), which has aroused much attention. By memorizing the documents as the parametric knowledge of LLM, this kind of method breaks the limitations of traditional IR in terms of document granularity and simple relevance matching Nguyen & Yates (2023), offering more flexibility and creativity, thus better meeting practical needs. However, two severe problems hinder the current LLM-based retrieval. 1)**Feature Locality**: LLM-based retrieval normally adopt language models to learn the mapping from queries to the relevant document DocIDs. However, these DocIDs actually can not fully represent the global information of the passages. Meanwhile, directly feeding the whole passages into LLM is costly and infeasible, causing a trade-off between effectiveness and efficiency. 2)**Retrieval Variance**: Due to the inherent generative inconsistency property of large language models , current LLM-based retrieval may generate unforeseeable variances, especially when the input query or the size of context varies, directly results in undesirable and vulnerable performance which may not be preferred.

Considering the problems mentioned above and better leveraging the capability of LLMs, we propose a fully LLM-based architecture with a two-stage fine-tuning method called Invar-RAG, as illustrated in Figure 1. In the retrieval stage, our approach initializes the pre-trained LLaMA Touvron et al. (2023) as the backbone and follows the bi-encoder architecture in DPR Karpukhin et al. (2020) to construct our retriever. Compared to normal GR methods which need an iterative process of encoding and decoding, we introduce a component called LLM-aligned Retrieval. It first represents the input query and corpora into high-dimension space using a small language model (MiniLM) Wang et al. (2020), then introduces a new loss function constructed by KL-divergence to align the coarse query-documents pairs representation to the LLM's representation space. This allows the retriever to leverage the rich prior knowledge of LLM, typically addressing the feature locality caused by only feeding DocIDs to LLM. Moreover, based on the initial objective of our LLM-aligned Retrieval, we introduce the invariance loss to overcome the variance in the retrieval stage. By recognizing the invariant pattern that contributes the most to the performance and gradually forcing the model to rely on the invariant pattern, we can avoid the unforeseeable variances in practice and enhance the robustness of our RAG system. Finally, in the generation stage, we freeze the weights we fine-tuned before and optimize the generation function to allow the LLM to give correct answers to the retrieved documents. Our contributions are summarized as follows:

- We introduce Invar-RAG, a novel framework featuring a two-stage fine-tuning method on a single shared LLM, including the retrieval stage and generation stage.
- We introduce a novel LLM-based retrieval method containing representation learning and invariance loss, respectively addressing the issues of feature locality and retrieval variance.
- We validate Invar-RAG's performance on three public ODQA datasets, no matter for retrieval performance or generation performance, demonstrating its superiority.

2 METHODOLOGY

In this section, we introduce a novel retrieval-augmented language model architecture, Invar-RAG, which addresses the previously mentioned issues by using LLM-aligned retrieval combined with a specially designed invariance loss. We first present an overview of our proposed architecture, followed by a detailed explanation of its key components, and finally, we introduce how we construct the prompts.

2.1 OVERALL FRAMEWORK OF INVAR-RAG

In this section, we provide an overview of Invar-RAG, as shown in Fig. 1. We begin by using query rewriting and context window resizing to introduce various types of variance. Next, we apply a small language model to map these texts into a high-dimensional vector space, generating coarse representations. We then adopt LLM-aligned retrieval to align the coarse representation with the LLM's representation and compute the basic relevance score via dot product. This approach

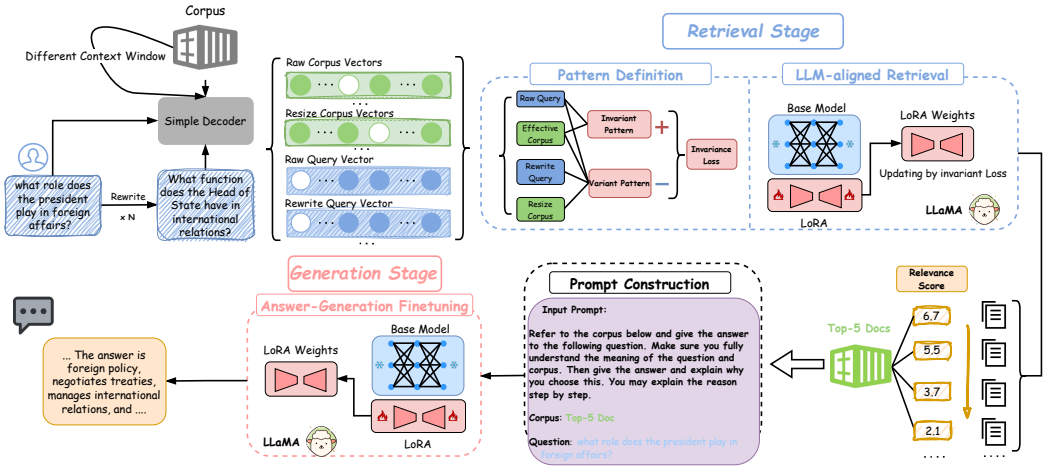

Figure 1: Overview of proposed Invar-RAG.

addresses the feature locality problem by feeding the entire document representation into the LLM, rather than a single DocID. Additionally, to address feature variance, we define an invariance loss based on the initial KL-divergence loss function in representation learning, encouraging the model to rely on invariant patterns. Finally, by constructing appropriate prompts and fine-tuning the generation stage, we optimize the LLM to better utilize the retrieved information, generating more accurate answers to the given questions.

2.2 Retrieval Stage

Architecture. For the Retriever architecture, we follow the approach of previous work Ma et al. (2024), using the bi-encoder architecture from DPR Karpukhin et al. (2020), but replacing the backbone model with LLaMA Touvron et al. (2023). Considering the efficiency, we first compute the vector embedding of a document $d_i^R \in \mathcal{D}^R$ as:

$$V_r(d_i^R) = \text{Decoder}\left(`t_1 \llcorner t_2 \llcorner \cdots \llcorner t_k'\right)[-1] \tag{1}$$

Where k represents the maximum number of trunks, and $\text{Decoder}(\cdot)$ represents the embedding layer of a small language model (MiniLM-v2), which maps the trunks $(t_1 \llcorner t_2 \llcorner \cdots \llcorner t_k)$ from the initial text space to a high-dimensional dense vector space. For the vector embedding of query q, we leverage our LLM structure to return the last layer token representation as the representation, denoted as V_q.

To leverage the LLM's prior knowledge while maintaining efficiency, we further align the query-document pairs to the LLM's representation space and denote the processed document representation as $V_p(d_i^R)$. Consequently, we can compute in terms of the dot product to get the two relevance scores:

$$\begin{aligned} s_{raw}\left(V_q, V_r(d_i^R)\right) &= V_q \cdot V_r(d_i^R) \\ s_{pro}\left(V_q, V_p(d_i^R)\right) &= V_q \cdot V_p(d_i^R) \end{aligned} \tag{2}$$

where the basic relevance score between the query and documents processed by small LM denotes as P_{raw} and the target relevance score computing between query and LLM-processed documents denotes as P_{pro}.

LLM-aligned Retrieval. Building on the above structure, we enhance our LLM-based retriever's ability to return more relevant documents. We introduce LLM-aligned retrieval with invariance loss in the retrieval stage, effectively addressing the aforementioned issues. Current alignment methods, such as RA-DIT Lin et al. (2023b), primarily focus on aligning the scoring functions between the retriever and the generator. However, the initial structural differences between novel retrievers (*e.g.,* DRAGON+ Lin et al. (2023a)) and LLMs still impede further optimization of the overall RAG

system. Therefore, we design a novel LLM-based retriever to resolve this issue. Unlike previous LLM-based GR methods Li et al. (2024), we no longer need to use DocIDs to retrieve relevant documents, which may cause the feature locality issue mentioned earlier. Instead, we adopt a fine-tuned alignment process that enables the LLM to perform representation learning. We leverage LoRA architecture Hu et al. (2021) to add additional adapter parameter $\theta_{\mathcal{R}}$ to our raw representation V_r, denoted as $V_r(d_i^R, \theta_{\mathcal{R}})$. The corresponding relevance score can then be re-normalized among top-k relevant chunks $\mathcal{D}^{R'} \subset \mathcal{D}^R$ as:

$$S_{\mathcal{R}}^r(V_r(d_i^R, \theta_{\mathcal{R}})|V_q) = \frac{\exp s_{raw}\left(V_q, V_r(d_i^R, \theta_{\mathcal{R}})\right)}{\sum_{d_i^{R'} \in \mathcal{D}^{R'}} \exp s_{raw}\left(V_q, V_r(d_i^{R'}, \theta_{\mathcal{R}})\right)} \tag{3}$$

For each document in the corpus, we need to compute $S_{\mathcal{R}}^r$ for n times (n represents the number of documents in $\mathcal{D}^R$) to rank the relevance scores. Consequently, the initial loss function for representation learning can then be defined by minimizing the KL-divergence Kim et al. (2021) of two relevance scores leveraging Eq. 3.

$$\mathcal{L}_{rl}\left(\mathcal{D}^R\right) = \mathbb{E}_{d_i^R \in \mathcal{D}^R} \mathrm{KL}(S_{\mathcal{R}}^r(V_r(d_i^R, \theta_{\mathcal{R}})|V_q) \| S_{\mathcal{R}}^r(V_p(d_i^R)|V_q)) \tag{4}$$

Following the previous works Lin et al. (2023b); Ma et al. (2020), fine-tuning both encoders hurt the performance Bao et al. (2024), we only update a part of our initialized retriever, which is in charge of computing the query representation.

Invariance Loss. To further enhance retrieval accuracy while maintaining robustness, we introduce invariance loss, building on our initial KL-divergence loss. Current refinement methods primarily rely on query rewriting He et al. (2016); Chan et al. (2024) or LLM generation Fan et al. (2024) to expand the search space and re-rank document chunks. However, they fail to recognize the effectiveness of different rewriting and generation procedures, directly resulting in the invariance problem. Specifically, we begin by rewriting the query and adjusting the context window to broaden the search space. Since not all refinement methods are effective, we identify invariant patterns to preserve retrieval performance while gradually incorporating weighted variant patterns to broaden the search space. To achieve this, we use the LSR score from LM-Supervised Retrieval Shi et al. (2023) to determine whether a document effectively enhances the LLM's answer prediction capability. For a training sample (q, y), where q and y respectively represent the input query and output result, we first define the output probability of LM as:

$$p_{LM}(y|V(d_i^R \circ x)) = \sum_{d_i^R \in \mathcal{D}^R} p_{LM}(y|V(d_i^R \circ x)) \cdot P_{\mathcal{R}}(d_i^R|x) \tag{5}$$

Then, for the LSR score for a retrieved document d_i^R:

$$P_{LSR}(d_i|q, y) = \frac{\exp(p_{LM}(y|d_i \circ q)/\tau)}{\sum_{d_i' \in \mathcal{D}^{R'}} \exp(p_{LM}(y|d_i' \circ q)/\tau)} \approx \frac{\exp(p_{LM}(y|d_i \circ q)/\tau)}{\sum_{d_i' \in \mathcal{D}^{R}} \exp(p_{LM}(y|d_i' \circ q)/\tau)} \tag{6}$$

where τ is the temperature hyperparameter of LLM, $\mathcal{D}_R' \subset \mathcal{D}_R$ denotes the top-k retrieved trunks. Assuming the query after rewriting as q^r, documents set after resizing as $\mathcal{D}_R^{re} = (d_1^{re}, d_2^{re}, \cdots, d_n^{re})$, we can leverage the Eq. 6 to calculate the score matching from (1) q to d_i, (2) q^r to d_i, (3) q to d_i^{re} and (4) q^r to d_i^{re}. We recognize the invariant pattern as the top-l ranked documents, denoted as $\mathcal{D}_{in}$, where $0 < l < k$ is satisfied. For other documents, we assume them as variant pattern $\mathcal{D}_{var}$, which contribute little to generating effective answers.

The invariant loss function can be formalized as follows:

$$\mathcal{L}_{invar}(\mathcal{D}_{in}) = \mathrm{Var}_{\mathcal{D} \subseteq \mathcal{D}_{var}}(\mathbb{E}_{d_{in}^R \in (\mathcal{D}_{in} \cup \mathcal{D})} \mathrm{KL}(P_{\mathcal{R}}^r(V_r(d_{in}^R, \theta_{\mathcal{R}})|V_q) \| P_{\mathcal{R}}^p(V_p(d_{in}^R)|V_q))) \tag{7}$$

This invariance loss measures the variance of the model's aligning ability under multiple interventions (*i.e.,* query rewriting and context resizing) by only allowing the documents in the $\mathcal{D}_{invar}$ to update the loss. The whole training objective can then be presented as:

$$\min_{\theta_R} \mathcal{L}_{rl} + \lambda \mathcal{L}_{invar} \tag{8}$$

where the task loss $\mathcal{L}_{rl}$ is minimized to align the two different representations while the $\mathcal{L}_{invar}$ enables the model to rely more on the invariant pattern, and λ is a hyperparameter to balance between two objectives.

Table 1: The Statistics of Fine-tuning Datasets.

Dataset	HF identifier	$\mathcal{D}_R$	$\mathcal{D}_G$	Training Sample	Task
Wiki QA Yang et al. (2015)	wiki_qa	✗	✓	20360	Open-domain QA
FreebaseQA Yao et al. (2014)	freebase_qa	✓	✗	20358	Open-domain QA
MS-MARCO Bajaj et al. (2016)	ms_marco	✓	✗	80143	Open-domain QA
Web Question Dumais et al. (2002)	web_question	✗	✓	3778	Open-domain QA
SQuAD v2	squad_v2	✗	✓	130319	Reading Comprehension

2.3 Generation Stage

In the generation stage, We followed the same architecture as in the retrieval stage for answer prediction. To improve the generative capability of LLM for leveraging the retrieved information better, followed by prior works Lin et al. (2023b); Shi et al. (2023), we adopt another LoRA adapter to fine-tune our model on different tasks. Specifically, for the same training sample (x, y), we retrieve the *top-*$\overline{k}$ relevant document chunks $\mathcal{D}'_G \subset \mathcal{D}_G$ by performing our model on retrieval task. For each retrieved chunk $d_i \in \mathcal{D}'_G$, we design a special fine-tuning example by prepending it to the prompt as background information and create k independent instances for one original example: $\left\{(d_i \circ x, y)|i = 1, \cdots, \overline{k}\right\}$. Then, following the previous work Lin et al. (2023b); Qi et al. (2020), we fine-tune the language model using the next-token prediction objective and minimize the loss as follows:

$$\mathcal{L}(\mathcal{D}'_G) = -\sum_i \log p_{LM}(y|d_i \circ x) \tag{9}$$

By applying this fine-tuning method, the generation stage benefits in two ways: (i) it improves the model's performance on the generation task by providing more accurate predictions based on the retrieved information; (ii) when the retrieved documents fail to provide an accurate answer, the approach enables the LLM to rely on its parametric knowledge to generate an answer while disregarding misleading retrieved documents.

3 Experiment

In this section, we will first introduce the experiment setting. Then we present extensive experiments to evaluate the effectiveness of our proposed Invar-RAG architecture in different stages (retrieval and generation). All the reported experimental results are the average values obtained from five independent runs of the algorithm.

3.1 Setting

3.1.1 Datasets

Following the prior works Lin et al. (2023b); Asai et al. (2023), we choose two ODQA datasets (FreebaseQA Yao et al. (2014) and MS-MARCO Bajaj et al. (2016)) and one reading comprehension (RC) dataset to do the representation learning in the retrieval stage (denoted as $\mathcal{D}_R$) while leveraging other three datasets (Web Question Dumais et al. (2002), Wiki Question Answering Yang et al. (2015)) and SQuAD v2[1] for fine-tuning the LLM in the generation stage (denoted as $\mathcal{D}_G$). The statistic of chosen datasets is shown in Tab.1. For detailed descriptions and complied templates, please refer to **Appendix A**.

3.1.2 Evaluation

To access our performance, we conduct the evaluation on four knowledge-intensive datasets, such as *i.e.,* TriviaQA (denoted as TQA)[2], Natural Question (denoted as NQ)[3] and PopQA[4], that are not involved in the training progress. For the evaluation metric, we evaluate our model's generation

[1] https://rajpurkar.github.io/SQuAD-explorer/explore/v2.0/dev/Prime_number.html

[2] https://nlp.cs.washington.edu/triviaqa

[3] https://ai.google.com/research/NaturalQuestions

[4] https://huggingface.co/datasets/akariasai/PopQA

performance using the Exact Match Wongsuphasawat et al. (2012), which indicates whether gold answers are included in the model generations followed by the setting in prior work Lin et al. (2023b); Mao et al. (2024). Furthermore, to evaluate our proposed retriever's performance, we employ the Acc@5 and Acc@20 as evaluation metrics, which are widely used in related studies Chen et al. (2024); Izacard et al. (2021). These metrics assess the proportion of questions where the correct answers appear in the top-5 or top-20 retrieval results, offering a comprehensive evaluation of the retrieval performance. For more details, please refer to the description and methods in **Appdendix B**.

3.1.3 Implementation Details

In this section, we provide a detailed description of our framework's implementation. The code can be found in **Supplementary Material**. For both the retrieval and generation stages, the LLaMA-2-7B checkpoint[5] is leveraged to initialize the pre-trained weights of our architecture. For the GPU selection, We perform our further fine-tuning on 4×40G NVIDIA V100 GPUs.

Retrieval Stage. Following the previous work's setting Ma et al. (2024), as LLaMA is a decoder-only architecture, we append an end-of-sequence token <EOS> to the input sequence and regard the last layer representation as the dense representation to calculate the similarity score. Considering the possible effect caused by the size of each dense representation, we also employ the normalization procedure to map the original representation into unit vectors during both the training and inference stages. For the fine-tuning progress in the retrieval stage, we adopt LoRA architecture Hu et al. (2021) to reduce the high cost of GPU memory. The detailed hyperparameters we used can be found in **Appendix C**.

Generation Stage. We hold $\overline{k}$ in Sec.2.3 equal to 5 to generate instances for a single example and append multiple examples together to improve the efficiency (the length is limited to 4096 tokens). The used hyperparameters are also shown in the **Appendix C**. Other implementation details are the same as original papers Lin et al. (2023b); Shi et al. (2023).

3.1.4 Baselines

To demonstrate the effectiveness of our proposed architecture, we compare the retrieval performance of our Invar-RAG with state-of-the-art retrieval methods, including sparse retrieval (BM25 Ram et al. (2023)), dense retrieval (BGE Xiao et al. (2024), Contriever Izacard et al. (2021)) and LLM-based retrieval (LLM-embedder Zhang et al. (2023) and RepLLaMA Ma et al. (2024)). Furthermore, for the corresponding RAG performance, we conduct extensive experiments compared to the novel retriever + generation model to show our superiority. The descriptions for each baseline are listed in the **Appendix D**.

3.2 Overall Performance

In this section, we present performance comparison experiments on two stages, respectively, with three knowledge-intensive ODQA datasets. The results show that our Invar-RAG architecture outperforms all competing sparse, dense, and LLM-based baselines in retrieval and their downstream RAG in generation. Such a comparison highlights the effectiveness of our unique design for two-stage fine-tuning.

3.2.1 Retrieval Performance

In this section, we will present and analyze the retrieval performance of our designed architecture. As illustrated in Tab.2, the sparse retriever BM25 fails to map the given text to proper representations. Although employing an additional model as the re-ranker improves the performance to some extent, the retrieval capability remains sub-optimal due to the inferiority of BM25. Besides, Novel dense retrievers, like BGE and Contriever, present comparable performance over the three datasets, suggesting their effectiveness in leveraging contrastive learning or task-specific fine-tuning. However, they still slightly lag behind our designed Invar-retrieval because of the neglect of rich semantic information Ma et al. (2024). Current researchers have proposed several LLM-based retrievers *i.e.,*

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

Table 2: Retrieval performance comparison between our designed retriever in Invar-RAG and other baselines. The best results are bold, and the second-best are underlined.

Models	TQA		NQ		PopQA	
	Acc@5	Acc@20	Acc@5	Acc@20	Acc@5	Acc@20
BM25 Ram et al. (2023)	62.5	73.0	49.0	67.0	35.5	51.5
BM25+BGE(re-ranker) Chen et al. (2024)	<u>72.5</u>	78.0	68.0	76.5	54.0	60.0
Contriever Izacard et al. (2021)	68.0	<u>80.5</u>	68.0	84.0	62.0	77.5
BGE-base Xiao et al. (2024)	69.5	80.0	<u>77.0</u>	86.0	<u>72.0</u>	**83.0**
LLM-embedder Zhang et al. (2023)	67.5	77.5	75.5	<u>86.5</u>	70.0	79.5
RepLLaMA Ma et al. (2024)	66.5	76.0	72.0	85.5	68.5	74.5
Invar-retrieval (ours)	**74.0**	**81.5**	**80.5**	**88.0**	**73.5**	<u>82.5</u>
Improv.	2.1%	1.2%	4.6%	1.7%	2.1%	-0.6%

Table 3: Generation performance comparison between our designed Invar-RAG and other baselines. The best results are bold, and the second-best are underlined.

Models	TQA	PopQA	NQ
	Exact Match		
BGE-base + LLaMA-2-7B	<u>74.1</u>	49.8	52.1
BM25 + BGE(re-rank) + LLaMA-2-7B	72.3	48.2	51.6
LLM-embeder + LLaMA-2-7B	71.8	<u>51.1</u>	<u>54.1</u>
Contriever + LLaMA-2-7B	72.6	48.6	51.8
Invar-RAG	**75.3**	**53.6**	**56.2**
Improv.	1.6%	4.9%	3.9%

LLM-embedder Zhang et al. (2023) and RepLLaMA Ma et al. (2024), which leverage the rich prior knowledge that LLM initially has. However, due to the high cost of processing the massive corpus, it is infeasible to handle all the chunks within the LLM. Moreover, the variance problem that happens in LLM also leads to relatively inferior performance. Correspondingly, we propose our LLM-based retrieval model, Invar-retrieval, as a part of our designed Invar-RAG. The results shows that our methods outperform all the sparse, dense and LLM-based retrievers, especially under the Acc@5 measurement, contributing to our designed invariance loss in reducing the variant and ineffective patterns.

3.2.2 Generation Performance

In this section, we will analyze the answer generation capability for our designed Invar-RAG. Based on the astonishing performance of our designed Invar-retrieval, we further fine-tune the language model to leverage the retrieved documents for better question-answering capability. From the experimental results presented in Tab.3, our Invar-RAG shows reasonable performance on the three ODQA datasets, echoing the performance of the retrievers designed in Tab.2.

3.3 Ablation Study

In this section, we analyze the efficacy of the two-stage fine-tuning in the Invar-RAG architecture, including the retrieval stage (*LLM-aligned Retrieval with Invariance Loss*) and the generation stage. We design three variants: (1)*w/o representation learning*: this variant uses the coarse text representation mapped by small language model (MiniLM-v2 [6]) to calculate the relevance score and adopt the same generation fine-tuning method in Sec.2.3. (2)*w/o invariance loss*: the second variant leverages the KL-divergence loss without the additional invariance loss to perform the representation learning. (3)*w/o generative fine-tuning*: this variant directly feeds retrieved documents and the corresponding question as a prompt to generate the answer. The fine-tuning datasets for each variant we used are presented in Tab.4. From the performance comparison in Tab.5, We can conclude that:

- With the representation learning method, LLM-based retrieval contributes to improving the retrieval and corresponding generation performance.

[6] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Table 4: The Statistics of Datasets in Ablation study.

Model Variants	Retrieval Fine-tuning		Generation Fine-tuning		
	Freebase QA	MS-MARCO	Wiki QA	Web Question	SQuAD v2
Default	✓	✓	✓	✓	✓
w/o representation learning	✗	✗	✓	✓	✓
w/o invariance loss	✓	✓	✓	✓	✓
w/o generative fine-tuning	✓	✓	✗	✗	✗

Table 5: Ablation Study on TQA.

Model Variants	Retrieval		Generation
	Acc@5	Acc@20	Exact Match
Default	74.0	81.5	75.3
w/o representation learning	63.5	73.6	74.1
w/o invariance loss	71.5	81.0	74.6
w/o generative fine-tuning	/	/	73.4

- Invariance loss significantly boosts our designed Invar-RAG by making the prediction rely more on invariant patterns.
- Generative fine-tuning is crucial for enhancing LLM's capability of giving predictions based on retrieved information. Moreover, it shows the effectiveness of the two-stage fine-tuning for a single LLM.

For ablation results on other two datasets (NQ and PopQA), please found them in **Appendix E**.

3.4 Invariance Analysis

In this section, we leverage a special example in TQA to illustrate the effectiveness of our designed invariance loss in two parts: (i) the importance of defining different patterns, (ii) the difference in retrieval performance that invariance loss brings.

As mentioned in Sec. 2.2, we return four different sets of retrieved documents and rerank them by LSR score to identify the invariant pattern. There are two reasons to explain this:

- Rewriting the query and resizing the context window does affect the normal relevance score computing by the dot product, leading to the variance in practice when we feed different lengths or formats of questions to the RAG system to ask for the answer.
- Prior works Zhang et al. (2024) have shown that adding a suitable amount of irrelevant or relatively ineffective documents does help improve the retrieval performance.

To verify that, we present the normal relevance score and LSR score of each retrieved document in four different sets in Fig. 3.4. The darker color represents the change that happened in the Top-5 documents. We can see that, for the question: 'Who was the man behind The Chipmunks', the relevance score for the top-5 documents in each set shows substantial changes while the LSR score does not vary a lot, which means the variance caused by rewriting query or resizing context window change the importance of documents, directly resulting in poor retrieval performance. Moreover, as illustrated in Fig. 3.4, the variant without invariance loss gives the wrong answer to our selected question example while our designed Invar-RAG system accurately predicts the result.

4 Related Work

Information Retrieval: Advancements in deep learning have revolutionized information retrieval systems, enhancing their personalization and accuracy in retrieving relevant documents. Early information retrieval frameworks employed sparse retrievers Ram et al. (2023) or dense retrievers Izacard et al. (2021); Xiao et al. (2024) to represent large corpora but struggled to capture deep semantic relationships Karpukhin et al. (2020). LLM-based retrievers (generative retrieval) have since emerged as notable methods, leveraging the rich prior knowledge of LLMs to significantly improve performance by converting documents into parametric knowledge and generating them instead of computing similarity scores Zhu et al. (2021). However, the frequent encoding and decoding processes in LLMs

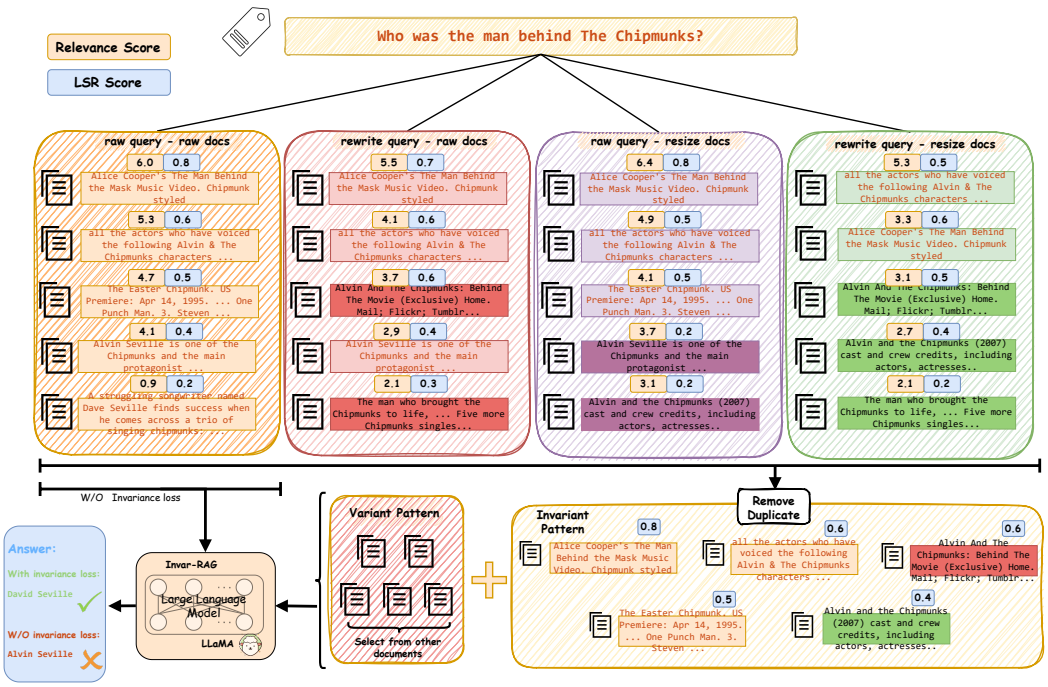

Figure 2: Special example for illustrating the effectiveness of invariance loss.

severely hinder efficiency Zhu et al. (2021). To address the trade-off between effectiveness and efficiency, we propose invar-retrieval in our architecture, enabling the model to efficiently retrieve the most relevant documents without introducing variance.

Retrieval-augmented Language Model: Currently, retrieval-augmented language models have proven effective in answering questions by leveraging external information through the integration of novel retrievers and LLMs Zhu et al. (2021). However, the architectural gap between retrieval and generation continues to hinder unified optimization across the entire retrieval-augmented generation system Wang et al. (2023). To address the isolation between retrieval and generation, a novel architecture called RA-DIT was introduced Lin et al. (2023b). By aligning retriever scoring with LSR scoring Shi et al. (2023), it has been shown to deliver state-of-the-art performance across various tasks. However, it still employs dense retrievers like DRAGON+ Lin et al. (2023a) in the retrieval stage, which fails to eliminate the problem at its source and introduces inefficiencies throughout the process. Correspondingly, we introduce a representation learning method and invariance loss in our Invar-RAG architecture, which partially addresses these issues and explores a novel approach to using a single LLM for multiple roles within the RAG system.

5 Conclusion

In this paper, we analyze the challenges and problems of current methods to apply the large language model as a retriever in the RAG system and propose a novel framework, Invar-RAG, to address these challenges. We introduce an LLM-aligned retrieval method, incorporating a well-designed representation learning approach to align coarse query-document pairs with the LLM's representation space, allowing our architecture to leverage the extensive parametric knowledge of the LLM to compute relevance scores. Additionally, to address retrieval variance, we propose invariance loss, building on our initial KL-divergence loss, during the retrieval stage to reduce the impact of irrelevant documents. Finally, we perform additional fine-tuning on the same LLM for the answer-generation task, enabling our architecture to better utilize the retrieved information and provide more accurate predictions. Extensive experiments on three open-domain question-answering datasets confirm Invar-RAG's superiority and validate the effectiveness of each module.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*, 2023.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

Hangbo Bao, Li Dong, Wenhui Wang, Nan Yang, Songhao Piao, and Furu Wei. Fine-tuning pretrained transformer encoders for sequence-to-sequence learning. *International Journal of Machine Learning and Cybernetics*, 15(5):1711–1728, 2024.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. Rq-rag: Learning to refine queries for retrieval augmented generation. *arXiv preprint arXiv:2404.00610*, 2024.

Xiaoyang Chen, Ben He, Hongyu Lin, Xianpei Han, Tianshu Wang, Boxi Cao, Le Sun, and Yingfei Sun. Spiral of silences: How is large language model killing information retrieval?–a case study on open domain question answering. *arXiv preprint arXiv:2404.10496*, 2024.

Daixuan Cheng, Shaohan Huang, and Furu Wei. Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations*, 2023.

Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. Web question answering: Is more always better? In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 291–298, 2002.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6491–6501, 2024.

Yunlong He, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. Learning to rewrite queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 1443–1452, 2016.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

Taehyeon Kim, Jaehoon Oh, NakYil Kim, Sangwook Cho, and Se-Young Yun. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. *arXiv preprint arXiv:2105.08919*, 2021.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474, 2020.

Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. From matching to generation: A survey on generative information retrieval. *arXiv preprint arXiv:2404.14851*, 2024.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*, 2023a.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. Ra-dit: Retrieval-augmented dual instruction tuning. *arXiv preprint arXiv:2310.01352*, 2023b.

Shuming Ma, Dongdong Zhang, and Ming Zhou. A simple and effective unified encoder for document-level machine translation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 3505–3511, 2020.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2421–2425, 2024.

Yuren Mao, Xuemei Dong, Wenyi Xu, Yunjun Gao, Bin Wei, and Ying Zhang. Fit-rag: Black-box rag with factual information and token reduction. *arXiv preprint arXiv:2403.14374*, 2024.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

Thong Nguyen and Andrew Yates. Generative retrieval as dense retrieval. *arXiv preprint arXiv:2306.11397*, 2023.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*, 2020.

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

Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*, 2023.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023.

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

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*, 2023.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020.

Krist Wongsuphasawat, Catherine Plaisant, Meirav Taieb-Maimon, and Ben Shneiderman. Querying event sequences by exact match or similarity search: Design and empirical evaluation. *Interacting with computers*, 24(2):55–68, 2012.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 641–649, 2024.

Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 2013–2018, 2015.

Xuchen Yao, Jonathan Berant, and Benjamin Van Durme. Freebase qa: Information extraction or semantic parsing? In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pp. 82–86, 2014.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. Retrieve anything to augment large language models. *arXiv preprint arXiv:2310.07554*, 2023.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*, 2024.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60, 2024.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*, 2021.