# Predict the Retrieval! Test Time Adaptation for Retrieval Augmented Generation

**Anonymous ACL submission**


## Abstract

Retrieval-Augmented Generation (RAG) has emerged as a powerful approach for enhancing large language models' question-answering capabilities through the integration of external knowledge. However, when adapting RAG systems to specialized domains, challenges arise from distribution shifts, resulting in suboptimal generalization performance. In this work, we propose `TTARAG`, a test-time adaptation method that dynamically updates the language model's parameters during inference to improve RAG system performance in specialized domains. Our method introduces a simple yet effective approach where the model learns to predict retrieved content, enabling automatic parameter adjustment to the target domain. Through extensive experiments across six specialized domains, we demonstrate that `TTARAG` achieves substantial performance improvements over baseline RAG systems.


## 1 Introduction

Retrieval-Augmented Generation (RAG) (Izacard and Grave, 2021; Lewis et al., 2020; Edge et al., 2024) has emerged as a crucial approach for enhancing large language models (LLMs) (Radford et al., 2019; Brown et al., 2020; Bubeck et al., 2023) by addressing their inherent knowledge limitations. Through the integration of external knowledge sources (Pasca, 2019; Bollacker et al., 2008; Jin et al., 2019), RAG systems not only improve the accuracy of LLM responses but also help mitigate hallucination issues while eliminating the need for extensive model retraining.

However, while most current research has focused on the effectiveness of RAG systems for general domains, significant challenges persist in adapting these systems to specialized domains. These systems often struggle with distribution shifts and domain-specific data dependencies (Xu et al., 2025; Shi et al., 2024), frequently failing to accurately utilize information in domain-specific contexts (Miller et al., 2020; Liu et al., 2022). This limitation is particularly problematic in critical domains such as healthcare (Raja et al., 2024), legal services (Reji et al., 2024), and financial applications (Yepes et al., 2024), where accuracy and reliability are paramount.

To address these challenges, test-time adaptation (TTA) (Sun et al., 2020; Hardt and Sun, 2024; Karmanov et al., 2024) offers a promising solution for enhancing model performance. TTA allows models to dynamically adapt their parameters at inference time through self-supervised learning objectives, without the need for labeled data (Chen et al., 2022; Liang et al., 2024). This approach is particularly valuable when dealing with domain shifts and distribution changes that weren't anticipated during initial training. Building on these insights, we propose a simple yet powerful method for adapting RAG systems during inference: `TTARAG`. Our approach generates self-supervised learning signals by dividing retrieved passages into prefix-suffix pairs and training the model to predict suffix content from prefix context. This technique enables LLMs to perform real-time parameter updates when encountering new domains, effectively leveraging domain knowledge stored within the model parameters.

Through extensive experiments across six specialized domains, we demonstrate that `TTARAG` achieves substantial performance improvements over baseline RAG systems. Our approach consistently outperforms both standard RAG and baselines like Chain-of-Thought and In-Context Learning, achieving the best results in 19 out of 24 experimental settings while maintaining computational efficiency. These results validate the effectiveness of our approach for domain-specific applications.

## 2 Methodology

Our approach introduces a test-time adaptation mechanism for retrieval-augmented generation that enables model optimization during inference without access to ground truth labels. The key innovation lies in designing a self-supervised learning objective using retrieved passages as supervision signals.

### 2.1 Overview

Given a test input query $q$ and retrieved passages $\{p_1, ..., p_k\}$, we formulate a self-supervised adaptation objective by splitting passages into prefix-suffix pairs for prediction:

$$\mathcal{L}_{adapt} = -\sum_{i=1}^{k} \log P(p_i^{suffix} | p_i^{prefix}, q; \theta) \quad (1)$$

where $\theta$ represents the model parameters.

### 2.2 Context Processing

The adaptation process begins with careful processing of the retrieved passages to create meaningful prefix-suffix pairs for training.

**Length Filtering** To ensure sufficient context for meaningful adaptation, passages shorter than a configured minimum length threshold are filtered out.

**Passage Splitting** Each passage is split into prefix-suffix pairs using a two-tier strategy:

- **Primary Strategy** Passages are split at first natural linguistic boundaries marked by punctuation (periods, commas, semicolons, colons, exclamation marks, and question marks)
- **Fallback Strategy** When no suitable punctuation-based split exists, the passage is divided at its midpoint, ensuring each segment contains at least three words.

### 2.3 Parameter Adaptation Process

The adaptation process employs a gradient-based optimization approach:

#### 2.3.1 Initialization

Prior to the adaptation process, the model parameters are reset to their original pre-trained state to ensure a clean starting point for each adaptation iteration. An AdamW optimizer is then initialized with carefully configured hyperparameters: learning rate $\alpha$ for controlling update step sizes, epsilon $\epsilon$ for numerical stability, and weight decay $\lambda$ for regularization.

#### 2.3.2 Training Loop

For each batch of prefix-suffix pairs:

$$\theta_t = \theta_{t-1} - \alpha \cdot \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \mathcal{L}_{adapt}^i \quad (2)$$

where $N$ is the gradient accumulation steps and $\mathcal{L}_{adapt}^i$ is the loss for the $i$-th pair.

During training, the complete text (prefix and suffix) is first tokenized. The model then computes the loss on the suffix prediction task, where prefix tokens are masked during label preparation. To ensure stable training, gradients are accumulated over two steps and clipped to a maximum norm threshold. The AdamW optimizer then updates model parameters using these accumulated gradients. Since we only adapt on 1-5 prefix-suffix pairs in our experiments, the computational overhead remains acceptable.

### 2.4 Response Generation

After parameter adaptation, the model generates the final response using the adapted parameters $\theta'$:

$$y = \arg\max_y P(y | q, \{p_1, ..., p_k\}; \theta') \quad (3)$$

This approach enables effective domain adaptation through self-supervised learning on retrieved passages, allowing the model to dynamically align with the target domain during inference time without requiring ground truth labels.

## 3 Experiments

### 3.1 Datasets

We conduct experiments on CRAG (Yang et al., 2024) as the evaluation benchmark. CRAG is a comprehensive RAG benchmark containing 2,706 question-answer pairs across five domains: Finance, Sports, Music, Movie, and Open domain. The questions are constructed through web content-based creation where annotators formulate real-world questions answerable through web search.

To evaluate the effectiveness of `TTARAG` in the medical domain, we conduct additional experiments on two specialized datasets: PubMedQA (Jin et al., 2019), which contains 1,000 biomedical research question-answer pairs, and BioASQ (Tsatsaronis et al., 2015), comprising 500 expert-curated question-answer pairs from the biomedical literature.

Table 1: Performance comparison across different domains. Numbers represent accuracy scores (%). Best results for each model group are shown in **bold**.

| | CRAG | | | | | | Medical | |
|---|---|---|---|---|---|---|---|---|
| **Model** | **Finance** | **Sports** | **Music** | **Movie** | **Open** | **Overall** | **BioASQ** | **PubMedQA** |
| *Llama-3.1-8b-it* | | | | | | | | |
| Base | 17.4 | 27.6 | 34.9 | 31.3 | 42.4 | 29.8 | 55.6 | 46.6 |
| CoT | 17.9 | **30.2** | 37.6 | 31.5 | **45.8** | 31.6 | 54.6 | 50.8 |
| ICL | 16.1 | 24.8 | 33.5 | 29.4 | 40.4 | 28.0 | 49.8 | 53.6 |
| TTARAG | **20.1** | 29.5 | **37.7** | **34.6** | 41.5 | **31.9** | **75.0** | **57.4** |
| Δ vs Base | +2.7 | +1.9 | +2.8 | +3.3 | -0.9 | +2.1 | +19.4 | +10.8 |
| *Llama-2-7b-chat* | | | | | | | | |
| Base | 14.7 | 23.2 | 36.5 | 30.4 | 39.2 | 27.8 | 54.1 | 47.6 |
| CoT | 15.7 | **26.7** | 34.3 | 31.4 | **41.5** | 29.1 | 55.1 | 48.2 |
| ICL | 16.0 | 24.2 | 36.1 | 31.2 | 39.2 | 28.4 | 55.6 | 43.4 |
| TTARAG | **16.4** | 25.8 | **40.7** | **33.8** | 41.1 | **30.5** | **71.8** | **54.0** |
| Δ vs Base | +1.7 | +2.6 | +4.2 | +3.4 | +1.9 | +2.7 | +17.7 | +6.4 |
| *ChatGLM-3-6b* | | | | | | | | |
| Base | 9.8 | 18.7 | 31.4 | 22.4 | 33.4 | 22.0 | 51.4 | 19.8 |
| CoT | 12.7 | 20.6 | 28.4 | **25.8** | 33.9 | 23.6 | 44.3 | 22.4 |
| ICL | 9.9 | 18.2 | 30.8 | 22.1 | 33.0 | 21.8 | 50.8 | 19.2 |
| TTARAG | **14.0** | **22.1** | **33.5** | 25.5 | **38.1** | **25.7** | **58.4** | **44.8** |
| Δ vs Base | +4.2 | +3.4 | +2.1 | +3.1 | +4.7 | +3.7 | +7.0 | +25.0 |

### 3.2 Baselines

We evaluate TTARAG against several strong baselines, including prompting techniques (**Chain-of-Thought** (Wei et al., 2022), **In-Context Learning** (Brown et al., 2020)) and state-of-the-art pretrained RAG models (**Ret-Robust** (Yoran et al., 2024), **RAAT** (Fang et al., 2024) , **Self-RAG** (Asai et al., 2023)). Detailed descriptions of each baseline are provided in Appendix B.

### 3.3 Experimental Results

Table 1 presents comprehensive evaluation results across different domains and model architectures. Several key observations emerge from our experiments: TTARAG demonstrates consistent improvements across specialized domains, with Llama-3.1-8b-it showing notable gains in Finance (+2.7%), Music (+2.8%), and Movie (+3.3%) domains, and particularly strong performance in medical domains (BioASQ +19.4%, PubMedQA +10.8%). All three model architectures benefit from our approach: Llama-3.1-8b-it achieves the highest overall accuracy (31.9%), Llama-2-7b-chat shows remarkable adaptation capability in medical domains (+17.7% on BioASQ), and ChatGLM-3-6b demonstrates significant improvements in PubMedQA (+25.0%) and consistent gains across CRAG domains (+3.7% overall). While both CoT and ICL show some improvements over the base models, TTARAG consistently outperforms these baselines in specialized domains, with the only exception being Open domain tasks where CoT occasionally shows stronger performance, particularly with Llama-3.1-8b-it (45.8% vs 41.5%).

Table 2 presents a performance comparison between different RAG models across various domains. Notably, three of the models (Ret-robust, RAAT, and Self-rag) are pre-trained RAG models based on Llama-2. Despite Ret-robust using the larger Llama-2-13b as its base, and RAAT and Self-rag using Llama-2-7b, all three pre-trained RAG models perform worse than the Llama-2-7b-chat model (which achieves 27.8% overall accuracy). This underperformance is consistent across most domains, with only RAAT showing strength in the BioASQ medical domain (64.9%). The results suggest that current RAG pre-training methods have limited generalization capabilities, as they

Table 2: Performance comparison with state-of-the-art pretrained RAG models.

| | CRAG | | | | | | Medical | |
|---|---|---|---|---|---|---|---|---|
| **Model** | **Finance** | **Sports** | **Music** | **Movie** | **Open** | **Overall** | **BioASQ** | **PubMedQA** |
| Base | 14.7 | 23.2 | 36.5 | 30.4 | 39.2 | 27.8 | 54.1 | 47.6 |
| Ret-Robust | 14.6 | 20.6 | 33.2 | 32.4 | 33.5 | 26.1 | 24.7 | 28.4 |
| RAAT | 13.4 | 18.1 | 28.6 | 25.2 | 31.7 | 22.7 | 64.9 | 46.6 |
| Self-rag | 11.4 | 19.8 | 22.5 | 20.9 | 26.7 | 19.8 | 57.1 | 43.4 |
| TTARAG | **16.4** | **25.8** | **40.7** | **33.8** | **41.1** | **30.5** | **71.8** | **54.0** |

fail to match or exceed the performance of the base model, even when using larger model architectures. TTARAG outperforms all other models across all domains, demonstrating the effectiveness of its approach compared to existing RAG pre-training methods.

**The effectiveness of segment-based adaptation** We compare our segment-based approach (splitting passages into prefix-suffix pairs) with a baseline that does not segment the passage, where we perform next-token prediction on the entire passage without segmentation. The results in Table 3 demonstrate that the segmentation strategy yields consistent performance gains across all model architectures: +1.1% for Llama-3.1-8b-it, +0.4% for Llama-2-7b-chat, and +0.7% for ChatGLM-3-6b. We attribute these improvements to the front-to-back prediction task better aligning with natural language understanding compared to token-by-token prediction, enabling more effective parameter updates. The larger improvement observed with Llama-3.1-8b-it (+1.1%) suggests that higher-capacity models may particularly benefit from structured adaptation approaches.

Table 3: The effectiveness of segmentation.

| Strategy | Llama-3.1-8b-it | Llama-2-7b-chat | ChatGLM-3-6b |
|---|---|---|---|
| TTARAG | 31.9 | 30.5 | 25.7 |
| wo seg | 30.8 | 30.1 | 25.0 |

We also conduct hyper-parameter analysis about the number of adaptation pairs and learning rate in Section C.

**On the computation efficiency** To evaluate the computational overhead of our approach, we measure the total inference time across different configurations and compare it with baseline methods. Table 4 shows the total and average inference times for different numbers of adaptation pairs (1-5), compared against Chain-of-Thought (CoT) and the original model without adaptation. The results are based on processing 2,706 queries from the CRAG dataset.

Table 4: Computation time analysis

| Metric | 1pair | 2pair | 3pair | 4pair | 5pair | CoT | Vanilla |
|---|---|---|---|---|---|---|---|
| Total | 4,740 | 5,723 | 6,621 | 7,001 | 7,023 | 11,688 | 961 |
| Avg | 1.75 | 2.11 | 2.45 | 2.59 | 2.60 | 4.32 | 0.36 |

While our method does introduce additional computational overhead compared to the original model, it remains significantly more efficient than CoT. The average processing time per query ranges from 1.75s (1-pair) to 2.60s (5-pair), which is substantially lower than CoT's 4.32s. This demonstrates that TTARAG achieves its performance improvements with reasonable computational cost, making it practical for real-world applications.

## 4 Conclusion

In this paper, we present TTARAG, a test-time adaptation approach for retrieval-augmented generation that enables dynamic model optimization during inference. Our method introduces a simple yet effective self-supervised learning objective where the model learns to predict retrieved content, allowing automatic parameter adjustment to target domains without requiring labeled data. Through extensive experiments across six specialized domain, we demonstrate that TTARAG achieves consistent improvements over the base RAG system, suggesting that test-time adaptation is a promising direction for improving RAG systems' performance in specialized domains while maintaining computational efficiency.

## Limitations

While TTARAG demonstrates strong performance improvements across various domains, there are several important limitations to consider:

The test-time adaptation process introduces additional computational overhead during inference. As shown in our experiments, the adaptation step increases the average inference time by 1.75-2.60 seconds per query compared to the base model, depending on the number of adaptation pairs used. This additional latency may impact real-time applications where response speed is critical. What's more, our approach requires additional GPU memory during inference for adaptation training compared to standard RAG systems. For larger models, this increased memory requirement may limit deployment options, particularly in resource-constrained environments.

## Ethical Considerations

Test-time adaptation may potentially affect the model's safety alignment due to parameter updates. However, since our method only updates parameters for a limited number of iterations, the model's safety alignment likely remains largely intact, with minimal risk of disruption. Nevertheless, we believe it is important to investigate the extent to which gradient updates on domain-specific data can impact a model's established safety alignment without compromising it. This represents an important direction for future research to better understand the relationship between adaptation and safety preservation.

## References


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

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. 2022. Parameter-free online test-time adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8344–8353.

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

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. 2022. Contrastive test-time adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 295–305.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

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. *arXiv preprint arXiv:2405.20978*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Moritz Hardt and Yu Sun. 2024. Test-time training on nearest neighbors for large language models. *Preprint*, arXiv:2305.18466.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. 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 7036–7050, Mexico City, Mexico. Association for Computational Linguistics.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. 2024. Efficient test-time adaptation of vision-language models. *Preprint*, arXiv:2403.18293.

I. Kuzborskij and F. Orabona. 2013. Stability and Hypothesis Transfer Learning. In *International Conference on Machine Learning (ICML)*, pages 942–950.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jian Liang, Ran He, and Tieniu Tan. 2024. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, pages 1–34.

Jian Liang, Dapeng Hu, and Jiashi Feng. 2020. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 6028–6039.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2024. RA-DIT: Retrieval-augmented dual instruction tuning. In *ICLR*.

Linqing Liu, Patrick Lewis, Sebastian Riedel, and Pontus Stenetorp. 2022. Challenges in generalization in open domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2014–2029, Seattle, United States. Association for Computational Linguistics.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Chatqa: Surpassing gpt-4 on conversational qa and rag. In *NeurIPS*.

Meta-AI. 2024. Llama 3 model card.

John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. 2020. The effect of natural distribution shift on question answering models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6905–6916. PMLR.

Marius Pasca. 2019. Wikipedia as a resource for text analysis and retrieval. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, page 24, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Mahimai Raja, E Yuvaraajan, et al. 2024. A rag-based medical assistant especially for infectious diseases. In *2024 International Conference on Inventive Computation Technologies (ICICT)*, pages 1128–1133. IEEE.

Sneha Ann Reji, Reshma Sheik, A Sharon, Avisha Rai, and S Jaya Nirmala. 2024. Enhancing llm performance on legal textual entailment with few-shot cot-based rag. In *2024 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, pages 1–6. IEEE.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. 2020. Improving robustness against common corruptions by covariate shift adaptation. In *Advances in Neural Information Processing Systems*, volume 33, pages 11539–11551.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Hang Wu, Carl Yang, and May D Wang. 2024. Medadapter: Efficient test-time adaptation of large language models towards medical reasoning. In *EMNLP*.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12991–13013, Bangkok, Thailand. Association for Computational Linguistics.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR.

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

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2021. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*.

Haoyu Wang, Tuo Zhao, and Jing Gao. 2024. Blendfilter: Advancing retrieval-augmented large language models via query generation blending and knowledge filtering. In *EMNLP*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2024. Instructrag: Instructing retrieval-augmented generation with explicit denoising. *ArXiv preprint*, abs/2406.13629.

Ran Xu, Hui Liu, Sreyashi Nag, Zhenwei Dai, Yaochen Xie, Xianfeng Tang, Chen Luo, Yang Li, Joyce C. Ho, Carl Yang, and Qi He. 2025. Simrag: Self-improving retrieval-augmented generation for adapting large language models to specialized domains. *Preprint*, arXiv:2410.17952.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, et al. 2024. Crag–comprehensive rag benchmark. *arXiv preprint arXiv:2406.04744*.

Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde, and Renyu Li. 2024. Financial report chunking for effective retrieval augmented generation. *arXiv preprint arXiv:2402.05131*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *ICLR*.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. In *NeurIPS*.

# A Related Work

## A.1 Retrieval Augmented Generation

Retrieval Augmented Generation (RAG) (Lewis et al., 2020; Borgeaud et al., 2022; Izacard and Grave, 2021) has emerged as a powerful paradigm for enhancing large language models (LLMs) with external knowledge. By integrating a retrieval system with LLMs, RAG enables models to access and leverage external knowledge sources during generation, effectively addressing the limitations of static, parameterized knowledge in LLMs.

Recent advances in RAG have focused on several key directions. First, researchers have explored dynamic retrieval processes (Jiang et al., 2023; Jeong et al., 2024; Su et al., 2024) to improve the relevance of retrieved content. Second, various filtering mechanisms (Yoran et al., 2024; Yu et al., 2024; Wang et al., 2024) have been developed to eliminate irrelevant contexts and enhance RAG robustness. Additionally, instruction-tuning methods (Liu et al., 2024; Lin et al., 2024; Wei et al., 2024) have been specifically designed to improve LLMs' search and RAG capabilities.

## A.2 Test-time inference adaptation

Test-time inference adaptation aims to adapt pre-trained models to unlabeled test data during inference time without accessing the source training data. This paradigm has gained increasing attention as a practical solution for handling distribution shifts in real-world applications (Wang et al., 2021; Chen et al., 2022; Boudiaf et al., 2022). Unlike traditional domain adaptation methods that require simultaneous access to both source and target domains, test-time adaptation only needs the pre-trained model and target data, making it more privacy-friendly and storage-efficient (Liang et al., 2020).

Early works in this direction focused on hypothesis transfer learning (Kuzborskij and Orabona, 2013), where models trained on source domains are adapted to target domains with limited labeled data. Recent advances have extended this to fully unsupervised scenarios, leveraging techniques like entropy minimization (Wang et al., 2021), self-training (Sun et al., 2020), and test-time normalization statistics calibration (Schneider et al., 2020) to adapt models using only unlabeled test samples.

Building on these advances, `TTARAG` introduces a simple yet effective approach for test-time adaptation in retrieval-augmented generation. By learning to predict subsequent tokens in retrieved passages, our method enables fully unsupervised adaptation without requiring access to source domain data or labeled examples.

## B Baseline Details

We evaluate `TTARAG` using three state-of-the-art instruction-tuned LLMs: **Llama-2-7b-chat** (Touvron et al., 2023), **Llama-3.1-8b-it** (Meta-AI, 2024), and **ChatGLM3-6b** (GLM et al., 2024). We compare against two widely adopted baselines:

**Chain-of-Thought (CoT)** A prompting technique that guides the model to generate step-by-step reasoning before producing the final answer.

**In-Context Learning (ICL)** A method that provides relevant examples in the input prompt to demonstrate the desired task behavior.

We also compare `TTARAG` with the three state-of-the-art general domain pretrained RAG models:

**Ret-Robust** An approach focused on improving retrieval robustness through strategic passage selection during training. The model learns to discriminate between high-quality and low-quality retrieved content by being trained on a carefully curated mix of passages with different relevance levels.

**RAAT** A retrieval-augmented model that introduces a novel noise-aware training strategy. It specifically targets the challenge of distinguishing between helpful and misleading retrieved information by incorporating an adaptive training mechanism that exposes the model to varying types of retrieval noise.

**Self-RAG** utilizes instruction fine-tuning to adaptively retrieve passages based on the question and determine if the passage contains useful information for answering the question.

## C Hyperparameter Analysis

**Learning Rate Analysis** We investigate the sensitivity of our method to different learning rates during test-time adaptation with number of adaptation pairs of 3. As shown in Figure 1, we evaluate learning rates ranging from 1e-6 to 1e-4 across all three model architectures. Llama-3.1-8B-it achieves optimal performance at 1e-5 (31.9% accuracy), with performance gradually declining at higher learning rates. ChatGLM-6B shows more robust behavior across different learning rates, reaching peak performance at 5e-6 to 1e-5 (25.8% accuracy). Llama-2-7B-chat demonstrates the most stable performance curve, with accuracy varying only slightly (30.4-30.8%) across all tested learning rates, peaking at 1e-6 (30.8% accuracy). These results suggest that smaller learning rates (1e-6 to 1e-5) generally provide better and more stable adaptation, likely because they prevent over-aggressive parameter updates that could disrupt the model's pre-trained knowledge. All models show consistent improvement over their original performance (indicated by dashed lines) across most learning rates, validating the robustness of our approach.

**Number of Adaptation Passages** We examine how the number of retrieved passages used for adaptation affects performance. This study helps determine the optimal amount of context needed for effective adaptation while considering computational efficiency. As shown in Figure 2, we observe different optimal points across model architectures. Llama-3.1-8B-it achieves peak performance with 3 adaptation pairs (31.7% accuracy), while Llama-2-7B-chat shows optimal results at 4 pairs (31.7% accuracy). ChatGLM-6B maintains relatively stable performance between 2-5 pairs, peaking at 5 pairs (25.8% accuracy). Notably, all models show performance degradation when using 10 pairs. This degradation likely stems from over-aggressive parameter updates that disrupt the model's pre-trained knowledge. Too many adaptation pairs may cause excessive deviation from the original parameters, compromising the valuable knowledge acquired during pre-training. These results indicate that a moderate number of adaptation pairs (3-5) generally provides the best balance between adaptation effectiveness and preserving the model's pre-trained knowledge.

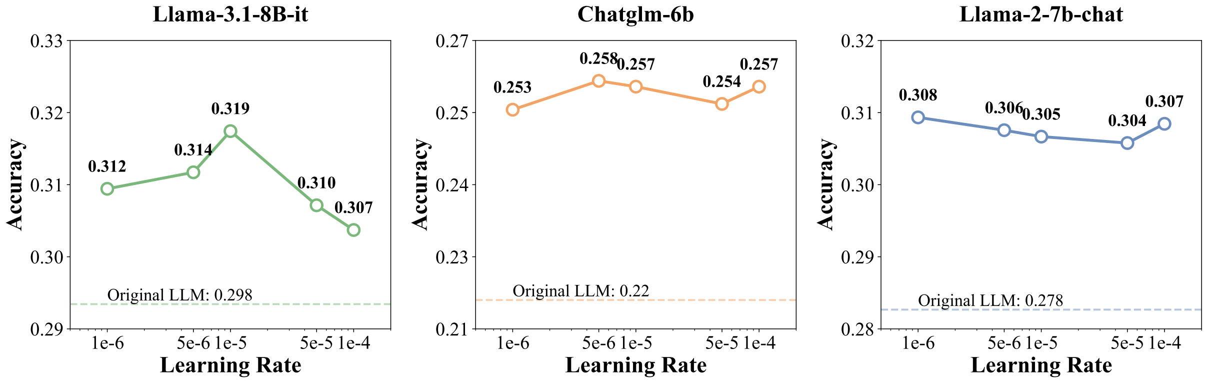


Figure 1: Accuracy vs. Learning Rate

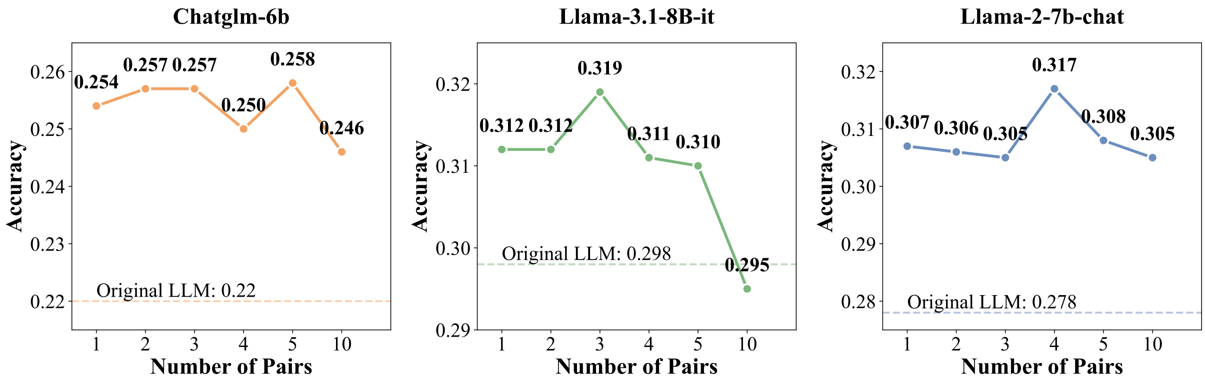


Figure 2: Accuracy vs. Number of Adaptation Pairs

## D Implementation Details

We use Llama-3.1-8b-instruct, ChatGLM-3-6b, Llama-2-7b-chat as our backbone models. Here we detail the hyperparameters and configuration settings used in our implementation. For the optimization process, we employ a learning rate of 1e-5, which provides a balance between adaptation effectiveness and stability. To improve training efficiency while managing memory constraints, we implement gradient accumulation with 2 steps. Gradient clipping is set at 0.1 to prevent gradient explosions, particularly important during rapid adaptation to new contexts. We use the AdamW optimizer with weight decay of 0.01 and epsilon of 1e-8, which helps prevent overfitting while maintaining numerical stability. Additional controls include filtering out sentences shorter than 6 tokens and limiting adaptation to 3 pairs per step. These parameters were determined through extensive experimentation across various domains, optimizing for both adaptation performance and computational efficiency. All experiments were conducted three times and the average results are reported.

All experiments are conducted on NVIDIA A100 GPUs with 80GB of memory. We utilize a fixed random seed of 42, and the experimental results are reported within a single run. For implementation, we use the following library versions: transformers 4.30.2, torch 2.1.0.

Table 5: Number of samples in each domain of CRAG dataset.

| **Domain** | Finance | Sports | Music | Movie | Open |
|---|---|---|---|---|---|
| #Samples | 661 | 519 | 373 | 611 | 542 |

## E Dataset Statistics

The statistics of the CRAG dataset are shown in Table 5.

## F Licensing

The CRAG, BioASQ and PubMedQA datasets are released for academic usage. These datasets are designed for evaluating RAG systems. Thus, our use of these datasets is consistent with their intended use.

The language models used in our experiments are released under the following licenses: **Llama-**

**2-7b-chat** (Touvron et al., 2023) is released under the Meta Llama 2 Community License Agreement. It is a variant of the Llama 2 family released in July 2023, featuring 7 billion parameters and specifically optimized for dialogue applications. **Llama-3.1-8b-it** (Meta-AI, 2024) is released under the Llama 3 License. Released in April 2024, it features 8 billion parameters and is specifically designed for instruction-following tasks, representing one of the most advanced open-source LLMs. **ChatGLM3-6b** (GLM et al., 2024) is released under the Apache 2.0 License. It is a bilingual conversational language model featuring 6 billion parameters, demonstrating strong performance in both English and Chinese tasks. All these models are open for academic usage.