Test-Time Strategies for More Efficient and Accurate Agentic RAG

Anonymous ACL submission

Abstract

Agentic retrieval-augmented generation (RAG) frameworks such as Search-R1 (Jin et al., 2025) improve performance on complex, multi-hop questions by interleaving reasoning and retrieval, but they often retrieve previously seen information and struggle to integrate evidence into the current reasoning state. We study lightweight test-time modifications to Search-R1 that target these inefficiencies without changing training: contextualization of retrieved passages, de-duplication of previously seen documents, and a hybrid of both. Evaluated on 500-example subsets of HotpotQA (Yang et al., 2018) and Natural Questions (Kwiatkowski et al., 2019) using exact match (EM), LLM-based semantic matching, and average retrieval turns, our best method—contextualization with GPT-4.1-mini—improves exact match by 5.6% and reduces retrieval turns by 10.5% relative to the Search-R1 baseline. These results suggest that improved use of retrieved evidence, rather than enforcing retrieval diversity alone, is the main driver of better answer accuracy and efficiency.

1 Introduction

RAG systems have shown promising results in complex question answering (QA) tasks by combining external document retrieval with generative language models (Lewis et al., 2020). Despite this success, traditional RAG systems that rely on a single-step retrieval and generation process often struggle to handle complex or nuanced questions, especially those requiring deep contextual understanding and multi-hop retrieval. To address these complexities, recent research has proposed agentic RAG systems which utilize large language model (LLM) agents to orchestrate retrieval, refine search queries, and optimize responses (Singh et al., 2025; An et al., 2024; Chan et al., 2024). Another popular approach is to augment the reasoning loop of LLMs with a retrieval tool, enabling the model to autonomously use retrieval while performing multi-step reasoning (Li et al., 2025; Jin et al., 2025).

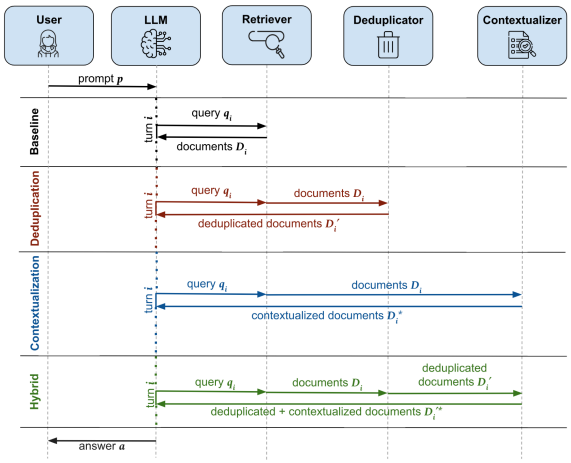

Figure 1: An illustration of the information flow for our proposed test-time strategies compared to the baseline during a single inference turn i. *Baseline:* This represents the standard Search-R1 framework, where the LLM sends a query (q_i) to the retriever and directly receives the retrieved documents (D_i) to continue its reasoning. *Deduplication:* This approach filters out previously seen content and returns only a set of novel documents (D_i') to the LLM. *Contextualization:* This approach parses the retrieved documents (D_i) and reformulates their content to improve integration into the LLM's reasoning process, returning an enhanced set of information (D_i^*). *Hybrid:* This approach combines both modules sequentially.

A notable example of this approach is the Search-R1 framework, which uses reinforcement learning (RL) to train LLMs for interleaved reasoning and retrieval (Jin et al., 2025). At inference time, the Search-R1 model first performs reasoning on a given user prompt p to either produce an answer a or generate a search query q to retrieve supporting information D. More specifically, in the i-th turn, the query q_i is sent to a dense retriever E5 (Wang et al., 2022), which returns relevant passages D_i from a 2018 Wikipedia dump. D_i is directly incor-

porated into the reasoning trace and fed back into the LLM, which continues its reasoning and repeats these steps until the final answer a is produced, see Figure 1 (Baseline). The Search-R1 models are trained using RL methods, such as PPO and GRPO, optimizing the exact match (EM) score between the ground truth and the predicted answer.

While Search-R1 has achieved substantial improvement—up to 41% over its baseline—our analysis of the Qwen2.5-7B Search-R1 model during inference has revealed several shortcomings. First, the model often performs repetitive retrieval of previously processed information, which leads to unnecessary retrieval turns and increased token consumption and latency. Second, the model often struggles to effectively contextualize retrieved passages, leading to suboptimal reasoning and inaccurate answers.

Research questions

- Will a concise representation of relevant information help an LLM become more efficient and accurate in question-answering tasks?

- Can preventing redundant document retrieval encourage greater contextual diversity, thereby improving efficiency and answer accuracy?

Proposed Approach

Our work builds upon and extends the Search-R1 framework (Jin et al., 2025) and investigates test-time approaches to improve the framework's reasoning efficiency and final answer accuracy. We address the limitations of Search-R1 through three test-time modifications that process the retrieved results D: (1) a **contextualization** module, (2) a **De-duplication** module, and (3) a **hybrid** approach that combines both.

2 Related Work

Prior work has improved retrieval-augmented generation by managing previously retrieved content (Shi et al., 2024), extracting useful information from retrieved documents before further reasoning (Li et al., 2025), and training retrieval-aware reasoning models with reinforcement learning (Huang et al., 2025). In contrast, we study lightweight test-time modifications to Search-R1 without changing model training or architecture.

3 Approach

Our qualitative analysis of Search-R1 reasoning traces revealed two recurring issues: **information forgetting**, where the model repeatedly retrieves previously seen content, and **ineffective information extraction**, where useful evidence is retrieved but not incorporated effectively into later reasoning. We therefore evaluate three test-time modifications to the Search-R1 pipeline.

3.1 Contextualization

The Contextualization module, shown in Figure 1 (Contextualization), uses an external language model to extract task-relevant information from retrieved documents after each retrieval step. Given the user prompt p and retrieved documents D_i at turn i, it produces a concise summary D_i^* containing only information useful for answering the question. This summary is appended to a persistent memory cache and provided at the next reasoning step.

The goal is to reduce information forgetting and improve the use of retrieved evidence without modifying Search-R1. By passing forward concise relevant context rather than only raw retrieved passages, the model can reason over both newly retrieved and previously retained information.

3.2 De-duplication of retrieved documents

To reduce repeated retrieval, we introduce a De-duplication module that filters out documents seen in earlier steps. For a given prompt p, we maintain a set of document IDs retrieved in previous turns. At each retrieval step, if one of the top-k documents D_i has already been seen, it is replaced by the next highest-ranked unseen document, yielding a set of unseen documents D_i' (Figure 1, Deduplication).

This intervention is both practical and diagnostic: it tests whether repeated retrieval reflects a genuine need to revisit the same evidence or a failure to use previously retrieved information effectively. If the same passages are necessary, answer quality should decline when duplicates are blocked. Otherwise, stable or improved performance would suggest that the main issue is not missing evidence, but ineffective use of evidence already retrieved.

3.3 Hybrid

The Hybrid approach combines Contextualization and De-duplication to test whether retaining rele-

vant information while enforcing retrieval diversity improves both answer quality and retrieval efficiency.

4 Experiments

4.1 Setup

4.1.1 Data source

Search-R1 (Jin et al., 2025) reports performance on the HotpotQA (Yang et al., 2018) and Natural Questions (NQ) (Kwiatkowski et al., 2019) datasets. Since their labeled test sets are not publicly available, we follow prior work and use the validation sets. Retrieval is performed on the 2018 Wikipedia dump with the E5 retriever.

4.1.2 Data splits

To reduce the cost associated with querying external LLMs, we created a smaller subset of question-answer pairs for evaluation. Specifically, we randomly sampled 500 question-answer pairs from the HotpotQA (Yang et al., 2018) and NQ (Kwiatkowski et al., 2019) validation sets. This subset is solely used for evaluation; no hyperparameter tuning or training is performed using this subset. All reported metrics are based on this validation set.

4.1.3 Baselines

We utilize the already trained Qwen2.5-7B Search-R1-base (PPO) as our main baseline for comparison. While running inference with both Qwen2.5-3B Search-R1-base (PPO) and Qwen2.5-3B Search-R1-instruct (GRPO), the latter model exhibited difficulties in adhering to the structured output format specified by the Search-R1 framework. Inference outputs showed frequent failures to generate required output tags such as <*think*> and <*search*> within the iterative reasoning loop. Additionally, the model often generated retrieved information under <*information*> tags by itself after the search query, and also occasionally fails to produce a final answer at the end of the reasoning chain. These behaviors indicate limitations in the ability to reliably follow instruction-guided formatting. Therefore, we only enhance the superior Qwen2.5-7B Search-R1-base (PPO) model to test our modules and report the corresponding results in Table 1. For all approaches, we run inference on our validation dataset of 500 questions and compute exact match, LLM Match, and the average number of turns.

4.1.4 Implementation details

We built on top of the publicly available Search-R1 source code on GitHub, where we made modifications to the model prompt to optimize the model's behavior. For inference, we use the HuggingFace Transformers library for the model forward pass and the E5 dense retriever provided in the Search-R1 GitHub repository over Wikipedia article chunks. Contextualization and LLM-as-a-Judge evaluation are performed by GPT-4.1-mini via the OpenAI API.

4.1.5 Evaluation Metrics

We report Exact Match (EM), as in Search-R1, and additionally introduce **LLM Match** together with the average number of retrieval turns. We include LLM Match because exact match can produce false negatives when the predicted answer does not exactly match the golden answer string despite referring to the same underlying entity, for example "2" vs. "Two" or "950 Pesos" vs. "P950".

For LLM Match, an external LLM (GPT-4.1-mini) is given the predicted answer and a set of ground-truth answers, and is asked to determine whether the prediction is semantically equivalent to any gold answer. Minor differences in phrasing are allowed as long as the predicted answer conveys the same meaning. The prompt directs the model to assign a binary score:

- **1** if the predicted answer is semantically equivalent to any ground truth answer, and
- **0** if it is incomplete or diverges in meaning.

This allows us to scale semantic evaluation without human annotation. We also report the average number of retrievals as a measure of inference efficiency, but interpret it together with answer accuracy, since a model could trivially reduce retrieval count by hallucinating an answer without retrieving evidence.

4.2 Results

In terms of answer accuracy, the Contextualization approach achieves a 5.6% increase in EM and a 6.7% increase in LLM Match score compared with the Search-R1 baseline. In addition, it is also the most efficient, reducing the average number of searches to 2.142, compared with the baseline which has 2.392 searches. While the Deduplication and Hybrid approaches have similar gains in EM and LLM Match over the baseline,

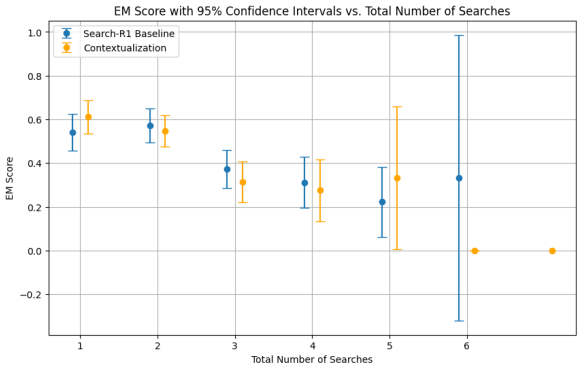

Figure 2: Illustration that questions requiring more agentic turns are inherently more difficult, as shown by the downward trend in Exact Match (EM) score for both the Search-R1 baseline and our Contextualization module. While the Contextualization module achieves a slightly higher mean EM at some points, the overlapping 95% confidence intervals suggest no clear difference between the two approaches at any given search count.

only the Hybrid approach reduces the average number of retrievals, similar to the Contextualization. The De-duplication pipeline is less efficient than the baseline, with 2.498 versus 2.392 average retrievals. Overall, the Contextualization approach still achieves the highest EM, LLM Match, and lowest average number of retrievals. All metrics along with the baseline are shown in Table 1.

Comparing the baseline and De-duplication outputs suggests that repeated retrieval is often not the main bottleneck. When duplicates are filtered out, the model tends to continue issuing similar queries for additional context, increasing retrieval count with little gain in answer accuracy. This suggests that ineffective use of retrieved information is a larger problem than retrieval overlap.

Taken together, these results suggest that the main advantage of Contextualization is not only better answer selection, but also better carryover of useful evidence across reasoning steps. This is consistent with its simultaneous gains in Exact Match, LLM Match, and average retrieval count, whereas De-duplication alone provides smaller accuracy gains while increasing the number of searches. In other words, improving how retrieved evidence is compressed and reused appears more effective than enforcing novelty in the retrieved set alone.

Figure 2 shows a 95% confidence interval around exact match score of the Search-R1 baseline and the Contextualization pipeline, conditioned on the total number of searches performed. While we do not observe a clear separation between the two

Table 1: Performance comparison of our proposed modules against the Search-R1 baselines on our 500-question evaluation set. We report Exact Match, LLM Match, and the average number of turns. Our modules are applied as test-time enhancements to the Qwen2.5-7B base (PPO) model. Cont: Contextualization (Section 3.1). De-Dup: De-duplication (Section 3.2). Hybrid (Section 3.3). Best results for our approaches are in bold.

Variant	Exact Match	LLM Match	avg. # searches
Qwen2.5-3B Search-R1			
base (PPO)	0.292	0.356	1.410
instruct (GRPO)	0.310	0.396	2.054
Qwen2.5-7B Search R1			
base (PPO)	0.464	0.538	2.392
base (PPO) w/ Cont **(Ours)**	**0.490**	**0.574**	**2.142**
base (PPO) w/ De-Dup **(Ours)**	0.478	0.560	2.498
base (PPO) w/ Hybrid **(Ours)**	0.480	0.568	2.154

approaches at any given search count, both show a downward trend. This suggests that Exact Match is negatively correlated with the number of retrievals.

For the Search-R1 baselines and the three non-training approaches, the LLM Match is 16 to 18% greater than the exact match score. Inspecting cases where the LLM judge marks the predicted and gold answers as equivalent but Exact Match fails, we observe two common patterns: numerical answers and abbreviated or shortened names.

5 Conclusion

In this work, we evaluated three test-time modifications to the Search-R1 pipeline: Contextualization, De-duplication, and a Hybrid of both. All three improve answer accuracy over the Search-R1 baseline. Contextualization performs best overall, improving both answer quality and retrieval efficiency, while De-duplication alone increases retrieval count. The Hybrid approach improves both accuracy and efficiency, though less strongly than Contextualization alone.

Our findings suggest that lightweight memory-oriented test-time interventions can yield meaningful gains without retraining the underlying agentic RAG model or retriever.

Limitations

This work has several limitations. First, our evaluation is limited to two open-domain question answering benchmarks, HotpotQA and Natural Questions, using 500 sampled validation examples from each dataset. While these datasets are standard and allow comparison with prior work, they do not cover all retrieval settings, domains, or question types. As a result, the observed gains may not generalize to other tasks such as domain-specific QA, long-document retrieval, or settings with noisier corpora.

Second, our experiments focus on a single Search-R1 backbone, Qwen2.5-7B Search-R1-base (PPO), because smaller variants did not reliably follow the required inference format. This means our conclusions are about the behavior of our proposed test-time modifications on this particular model and setup, rather than about agentic RAG systems in general. Different base models, retrievers, or retrieval depths may interact with our modules differently.

Third, the contextualization module depends on an external proprietary model, GPT-4.1-mini, both for information extraction and for the LLM-as-a-Judge evaluation. This introduces an additional source of variance and potential bias. The quality of contextualization may depend on the external model's capabilities, prompting sensitivity, and API behavior. Similarly, although LLM-based evaluation helps account for semantically correct answers that exact match misses, it is still an imperfect proxy and may occasionally overestimate or underestimate answer correctness.

Fourth, our efficiency analysis is limited. We report the average number of retrieval turns, which captures part of the inference cost, but it does not fully measure latency, monetary cost, or total token usage. In particular, the contextualization module adds extra model calls, so fewer retrieval turns do not necessarily imply lower overall runtime or cheaper inference in deployment settings.

Finally, our work studies only test-time modifications and does not examine whether similar improvements could be obtained more robustly through training, fine-tuning, or better retriever optimization. A broader evaluation across more datasets, models, and cost metrics would strengthen the conclusions.

References

Zhiyu An, Xianzhong Ding, Yen-Chun Fu, Cheng-Chung Chu, Yan Li, and Wan Du. 2024. Golden-retriever: High-fidelity agentic retrieval augmented generation for industrial knowledge base. *Preprint*, arXiv:2408.00798.

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

Jerry Huang, Siddarth Madala, Risham Sidhu, Cheng Niu, Julia Hockenmaier, and Tong Zhang. 2025. Rag-rl: Advancing retrieval-augmented generation via rl and curriculum learning. *Preprint*, arXiv:2503.12759.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *Preprint*, arXiv:2503.09516.

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

Patrick 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*, volume 33, pages 9459–9474. Curran Associates, Inc.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. *Preprint*, arXiv:2501.05366.

Yunxiao Shi, Xing Zi, Zijing Shi, Haimin Zhang, Qiang Wu, and Min Xu. 2024. Enhancing retrieval and managing retrieval: A four-module synergy for improved quality and efficiency in rag systems. *Preprint*, arXiv:2407.10670.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. *Preprint*, arXiv:2501.09136.

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*.

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, Brussels, Belgium. Association for Computational Linguistics.