RVR: Retrieve-Verify-Retrieve Iterative Framework for Comprehensive Retrieval

Deniz Qian[1,2*] **Hung-Ting Chen**[1*] **Eunsol Choi**[1]
[1]New York University [2]Cohere
{dq2024, hung-ting.chen, eunsol}@nyu.edu

Abstract

Comprehensively retrieving diverse documents is crucial to address queries that admit a wide range of valid answers. We introduce retrieve-verify-retrieve (RVR), a multi-round retrieval framework designed to maximize answer coverage. Initially, a retriever takes the original query and returns a candidate document set, followed by a verifier that identifies a high-quality subset. For subsequent rounds, the query is augmented with previously verified documents to uncover answers that are not yet covered in previous rounds. RVR is effective even with off-the-shelf retrievers, and fine-tuning retrievers for our inference procedure brings further gains. Our method outperforms baselines, including agentic search approaches, achieving at least 10% relative and 3% absolute gain in complete recall percentage on a multi-answer retrieval dataset (QAMPARI). We also see consistent gains on two out-of-domain datasets (QUEST and WebQSP) using different base retrievers. Our work presents a promising iterative approach for comprehensive answer recall leveraging a verifier and adapting retrievers to a new inference scenario. [1]

1 Introduction

Retrieval is a key method to equip large language models (LLMs) with up-to-date, long-tail information. Despite the recent advances in retrieval systems, comprehensively recovering relevant documents remains challenging [1, 5]. In this work, we introduce Retrieve-Verify-Retrieve (RVR), a framework that performs multiple rounds of retrieval, where each round conditions on previously retrieved documents verified by an LLM. At the next iteration, the model receives the concatenation of the query and LLM verified output documents from the previous retrieval search, aiming to search for remaining gold documents. The high-quality subsets from the initial step and the outputs returned by the subsequent retriever are merged as the final output. Figure 1 visualizes our approach.

Unlike traditional retrievers that only take a query as input to retrieve documents, our retriever is trained to explicitly condition on previously retrieved, high-quality documents to target missing ones. This formulation encourages improving coverage over iterations, in contrast to a one-shot ranking task between the original query and documents. Recent agentic approaches [13, 28] also perform iterative search, using retrieved outputs to formulate the new query. However, they only update the LLM to formulate better queries while keeping the retriever fixed. In contrast, we use a simple rule (concatenation of the query and already identified answers) to form new queries and update the retrievers. Our approach is trained and evaluated on an open-domain, multi-answer QA benchmark, QAMPARI [1]. Compared to a single retriever fine-tuned on in-domain data, recent agentic search baselines, and traditional diversification methods such as query expansion, our approach shows consistent improvements in retrieval performance. Using off-the-shelf retrievers with our iterative framework already shows some gains, but larger gains are achieved by fine-tuning retrievers for our

*Equal contribution
[1]Code: https://github.com/timchen0618/RVR

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

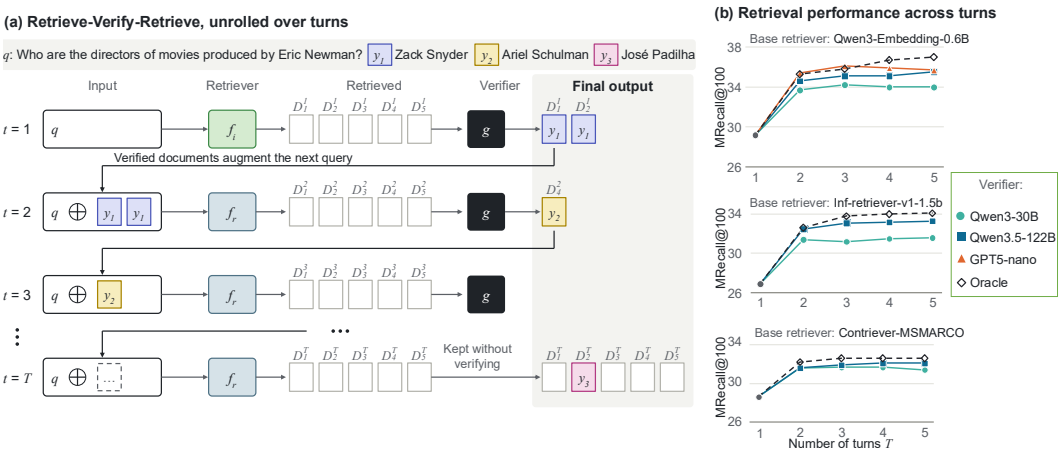

Figure 1: (a) Overview of our framework. Each query q aims to retrieve documents to get multiple answers (y_1, y_2, y_3). Given q, the initial retriever returns candidate documents, which are filtered by a verifier to identify evidence containing valid answers. Verified documents are concatenated ($\oplus$) with the original query to form query for subsequent retrieval turns. In this example, documents containing answers y_1 and y_2 are retrieved and verified in turn $t = 1$ and $t = 2$, respectively. Across turns, RVR accumulates verified documents and combines them with the final-turn ($t = T$) retrieval results to form the final output set. (b) Retrieval performance generally improves with the number of turns for different retrievers and verifiers. Improvements are bigger for a stronger verifier (Qwen3.5-122B), and oracle verifier results indicate remaining headroom.

inference setting. We further evaluate on other multi-answer QA benchmarks (QUEST [21] and WebQSP [34]), showing that our approach can generalize.

To understand the impact of verifier capabilities, we report performance under both the oracle verifier and verifiers of multiple sizes. The experimental results show that stronger verifiers can improve performance across multiple turns, and the oracle verifier can boost the performance even further, showing the headroom of the approach with better verifier models. Despite rich studies on agentic search [13, 28], most approaches focus on improving LLMs that generate new queries based on retrieval output rather than adapting retrievers for new inference scenarios. Our results showcases that fine-tuning retrievers for new inference scenarios can bring further performance gains, echoing findings from concurrent and prior work [19, 36, 8].

2 Background and Related Work

2.1 Comprehensive Retrieval: Task and Metric

Given a query q and a large corpus $\mathcal{C} = \{d_1, \ldots, d_N\}$, a retrieval system $f(q, \mathcal{C}, K)$ should identify a ranked subset of K documents $D_{\text{out}} = \{d_1, \ldots, d_K\} \subset \mathcal{C}$ that contains all answers $Y = y_1, y_2, ..., y_M$ for the query q. A retriever should aim to maximize both *relevance* and *coverage*, ensuring that the retrieved set reflects the full range of relevant, high-quality information. We assume a test set, where each query q is annotated with a set of M distinct gold answer strings $Y = y_1, y_2, ..., y_M$. We judge that document d_i covers answer y_j if the substring occurs in the document. We report the following two metrics related to task performance.

- **MRecall@K**: a binary score that equals 1 if all answers or at least K answers in the answer set $\{y_1 ... y_M\}$ are covered by D_{out}. This metric was introduced in prior work [22].
- **Recall@K**: the fraction of answers Y that are covered by at least one document in D_{out}.

2.2 Related Work

Iterative & Agentic Retrieval Several works have explored performing multiple rounds of retrieval for complex question answering [29, 33]. Qi et al. [23] introduced iterative query reformulation,

generating new retrieval queries from partially read passages. A line of work [11, 30, 31] similarly retrieves evidence iteratively, interleaving retrieval with intermediate LLM reasoning steps. A more recent line of work on agentic search systems [13, 25, 28] explored using retrievers as a tool, where an LLM agent alternates between reasoning and tool calling until it reaches the final answer. Concurrent work [26] shows that retrieving more documents at each turn and reranking them can improve answer accuracy. While most agentic approaches use off-the-shelf retrievers, recent work [19] trains retrievers for agentic search by optimizing for local relevance and global answer correctness in reasoning tasks. Our work focuses on improving answer coverage with an iterative framework, updating retrievers.

Comprehensive Retrieval We review prior retrieval datasets with queries that admit a set of answers. QAMPARI [1], WebQSP [34], and QUEST [21] expect a list of entity answers for a given query; Zhu et al. [37] requires multi-hop reasoning over multiple documents to aggregate information about multiple entities; Katz et al. [15] expects all documents representing an entity that matches a provided entity type; Chen and Choi [5] expects the candidate document set to cover multiple perspectives given a subjective question. Min et al. [22] introduces an autoregressive reranker conditioned on previously selected passages to encourage diversity and cover multiple answers. Chen et al. [6] proposed to generate multiple query embeddings autoregressively to retrieve more comprehensive document sets. In contrast, we choose an iterative framework, where the output from the previous retrieval step is provided as input to the next retrieval stage.

Verifier in Retrieval Recent work has incorporated verification into retrieval pipelines. An LLM can draft an answer, plan verification questions, retrieve supporting evidence, and revise, reducing hallucinations [10]. Self-RAG [3] integrates retrieval, generation, and critique via self-reflection signals. Verification operates *after* retrieval or generation, serving as a filtering mechanism. In contrast, our framework integrates verification *within* the retrieval loop itself: verification determines which retrieved documents are retained and used to condition subsequent rounds.

3 Method

3.1 Iterative Retrieval Framework

Our framework uses three components: an initial retriever f_i that uses only the query, an iterative retriever f_r that conditions on both the query and previously retrieved documents, and a verifier model g. Our subsequent retriever f_r can be viewed as a version of task-aware retrieval with instructions [2], which is trained to retrieve documents closer to the query q but distinct from documents in the input ($\bigoplus_{d \in D_{\text{ctx}}} d$). The initial and subsequent retrievers can be the same or different, and we evaluate both settings. Algorithm 1 and Figure 1 describe our framework.

Initial Retriever The initial retriever produces a ranked list of K documents given a query q and corpus $\mathcal{C}$($D_i = f_i(q, \mathcal{C}, K)$, line 2 in Algorithm 1).

Verifier Retrieval is often noisy and contains irrelevant documents. Thus, we implement a verifier g that examines retrieval outputs and produces a verified subset $D_V = \{d \in D_i : g(d, q) \wedge \text{rank}(d) \leq B\}$ containing documents deemed relevant by verifier g, where B is the verifier budget (line 4). The verified subset, D_V (1) contributes the final output (line 5), and (2) forms a query for the next retrieval stage. For subsequent retrieval turns, we select the top M documents from the verified set from the current round (line 6) to form the concatenated query $q_r = [q; \bigoplus_{d \in D_{\text{ctx}}} d]$. We use only the verified documents from the previous turn as input rather than all accumulated verified documents, because of the context length limitation.

Subsequent Retriever The subsequent retriever takes as input the concatenated query q_r, and produces new output $D_i = f_r(q_r, \mathcal{C}, B)$ (line 8). Using augmented query q_r allows f_r to reason about which relevant documents remain unretrieved, promoting answer coverage across iterations.

Output To form the final output set, we accumulate verified documents across all rounds and add any remaining documents from the final iteration. Set semantics remove duplicates $D_{\text{out}} = \text{TopK}(K, \left(\bigcup_{t=0}^{T-1} D_V^{(t)}\right) \cup D_i^T)$ where $D_V^{(t)}$ denotes the verified set at round t, and we take only the top $(K - |D_{\text{out}}|)$ documents from the final retrieval $D_i^{(T)}$ to ensure the total output size is K.

3.2 Training

We describe the two retrievers f_i and f_r. We assume a training dataset, where each instance contains a query q paired with a set of gold documents D^* from corpus $\mathcal{C}$. We do not train the verifier and instead use off-the-shelf LLMs. Our novelty lies in training the retriever f_r with the proposed iterative inference scenario. We thus generate training data (positive and negative documents) from this inference scenario. We use a standard contrastive learning objective [12]:

$$\mathcal{L}_\theta = -\log \frac{\exp(s(f_\theta(x), f_\theta(d^+))/\tau)}{\sum_{d \in D_{batch}} \exp(s(f_\theta(x), f_\theta(d))/\tau)}$$

where retriever f_θ encodes the input query x, positive document d^+, and in-batch documents d, s is a similarity function between the two embeddings, and τ is a temperature hyperparameter. We describe the process of constructing training data (x, d^+, D_{batch}) below.

Data for Initial Retriever (D_i) Each query q is paired with one positive document d^+ and one randomly sampled negative document d^- from the corpus $\mathcal{C}$. Additionally, we leverage in-batch negatives [14] such that all the other positive and negative documents from other training examples in the same batch serve as additional negatives. Thus, D_{batch} denotes all the documents in the batch, including the positive document, the sampled negative, and the in-batch negatives. We use the input query q directly as x, forming (x, d^+, D_{batch}).

Data for Subsequent Retriever (D_r) f_r takes gold documents as input in addition to query q, where the number of appended gold documents is controlled by hyperparameter M. We first uniformly sample an integer $m \in \{0, ..., \min(M, |D^*|)\}$, and then sample m documents from our ground truth set D^* to form context D_{ctx}. The input query x will be $[q; D_{\text{ctx}}]$. The positive document d^+ is randomly selected from $D^* \setminus D_{\text{ctx}}$. As with the training data for the initial retriever, we randomly sample one negative d^- from the corpus $\mathcal{C}$, and use in-batch negatives.[2]

Algorithm 1 Inference procedure for Retrieve-Verify-Retrieve.

Input: An input query q, corpus $\mathcal{C}$, initial retriever f_i, subsequent retriever f_r, a verifier g. **Hyperparameters:** verifier budget per turn B, max document context budget M, maximum number of turns T, final output set size K. **Output:** A ranked set of documents $D_{\text{out}} = \{d_1, \ldots, d_K\} \subset \mathcal{C}$, sorted by relevance to q.

$D_{\text{out}} \leftarrow \emptyset$
$D_i \leftarrow f_i(q, \mathcal{C}, K)$
for $t \in \{1, \ldots, T\}$ **while** $|D_{\text{out}}| < K$ **do**
  $D_{\text{v}} \leftarrow \{d \in D_i : g(d, q) \wedge \text{rank}(d) \leq B\}$ ▷ verified subset
  $D_{\text{out}} \leftarrow D_{\text{out}} \cup D_{\text{v}}$
  $D_{\text{ctx}} \leftarrow \textsc{TopK}(M, D_{\text{v}})$ ▷ first M for context
  $q_r \leftarrow [q; \bigoplus_{d \in D_{\text{ctx}}} d]$ ▷ concatenate query with context
  $D_i \leftarrow f_r(q_r, \mathcal{C}, K)$ ▷ perform $t+1$-th retrieval
end for
$D_{\text{out}} \leftarrow D_{\text{out}} \cup D_i$ ▷ add remaining
return $\textsc{TopK}(K, D_{\text{out}})$

Notation:

- $\{d \in D : P(d)\}$: elements in D satisfying predicate P
- $\textsc{TopK}(k, D)$: first k elements of ordered set D
- $\bigoplus_{d \in D} d$: concatenation over ordered set D
- $\text{rank}(d)$: position of document d in the retrieval ranking

Implementation Details We initialize the retrievers with pre-trained, off-the-shelf dual encoder retrievers: Contriever-MSMARCO [12], Qwen3-Embedding-0.6B [35], and INF-Retriever-v1-1.5B [32]. We use instances from the training portion of the QAMPARI dataset. Every model is fine-tuned for 50k steps. We use a learning rate of 1×10^{-4}, temperature $\tau = 0.05$, batch size 48 per GPU, gradient accumulation of 2 steps, warm-up for 1K steps, and the AdamW optimizer [20]. All training is conducted on 4 NVIDIA H200 GPUs, taking approximately 15 hours for Contriever, 30 hours for Qwen3-Embedding, and 19 hours for INF-Retriever on average across f_i and f_r configurations. Inference experiments are conducted on NVIDIA H200 and L40 GPUs. We prompt an LLM (Qwen-30B [27])[3] to serve as a verifier. Given an input document d and a query q, the verifier outputs a binary label indicating whether d is relevant to the query q. Formally, $g(d, q) \in \{0, 1\}$. We use vLLM [18] for inference. See Appendix D1 for the prompt, and we discuss verifier performance in Section 6.

[2] We explored using retrieved documents from the initial retriever as input during training, which more closely matches the inference setting. However, we found that it underperformed training with documents sampled from the gold set. This is similar to how teacher-forcing can outperform a training setup that more directly resembles the inference setting.

[3] Specifically, we use `Qwen/Qwen3-30B-A3B-Instruct-2507`.

4 Experimental Settings

4.1 Dataset

QAMPARI consists of open-domain questions paired with multiple valid answer strings (one example instance is shown in Figure 1). On average, each query is annotated with a set of 16.58 gold documents and contains 14.43 unique valid answers. Unlike other datasets, such as HotpotQA [33] and BrowseCompPlus [7] which assume a single gold answer, this dataset allows studying multi-answer coverage. We use their original splits (training, evaluation).

QUEST consists of queries paired with multiple valid answers, where each answer corresponds to a relevant Wikipedia document. The queries in QUEST specify set operations such as intersection, union, and difference (e.g., "what are some films about bats that are not Superhero films" (difference), "films that are South Korean adventure comedies or Canadian fantasy comedies" (union)). Queries have 10.5 answers on average.

WebQSP consists of questions from the Google Suggest API with their semantic parses and corresponding answer entities in Freebase. We treat WebQSP as a multi-answer retrieval task as there are on average 8.75 answers per query.

Retrieval Corpus We follow Amouyal et al. [1] and use a Wikipedia dump from 21-08-01 consisting of 25.9 million passages, each with 100 words. See Appendix B1, B2, and B3 for data statistics.

4.2 Comparison Systems

Base f_i We evaluate three pre-trained off-the-shelf retrieval models without fine-tuning.

FT (D_i) Each pre-trained retriever is fine-tuned on the QAMPARI training dataset with the standard contrastive objective (initial retriever setting in Section 3.2).

Agentic Retrievers We implement two strong open-source agentic baselines, Tongyi DeepResearch agent[4] [28] and SearchR1[5] [13]. Both fix a retriever and train an LLM that generates search query. The former undergoes continued pre-training on agentic trajectories and post-trained on synthetic QA pairs. The latter is trained using PPO [24] on Natural Questions [17] and HotpotQA [33] to maximize answer accuracy. In addition, we also present SearchR1 further fine-tuned on QAMPARI training data with a reward for retrieving any correct answer. The exact prompts and hyperparameters we use are provided in Appendix D2, D3, and D4, and example trajectory traces are provided in Appendix D5 and D6. We provide the fine-tuned retriever (**FT** (D_i)) as the only search tool for them to use in all experiments.

MMR We rerank the top 200 retrieved documents using Maximal Marginal Relevance [4], which promotes query relevance while penalizing similarity to previously selected documents. The penalty is controlled by λ. We evaluate $\lambda \in \{0.3, 0.7\}$, and report results for other λ values in Appendix C13.

Query Expansion An LLM generates an additional query to retrieve another set of documents. Each query is used to retrieve 50 documents, to form a final set of 100 documents. The prompt used to expand the query can be found in Appendix D7.

Instruction-Aware Retrieval For this setting we run two turns of retrieval (FT (D_i) + FT (D_r)) without concatenating documents to the query in the second round. Instead, in turn two, the retriever is prompted with an instruction to avoid retrieving documents covering candidate answers found in the initial retrieved set. The prompts for extracting candidate answers and retrieval are in Appendix D8.

Ours We evaluate three configurations, differing in what is used for the initial retriever (f_i) and the subsequent retriever (f_r): (1) FT (D_i) + FT (D_i): uses the same fine-tuned initial retriever f_i in both rounds; (2) FT (D_i + D_r) + FT (D_i + D_r): uses a single model trained on the union of the two training sets (D_i and D_r); (3) Fine-tuned (D_i) + Fine-tuned (D_r) : uses fine-tuned f_i in round 1 and fine-tuned subsequent retriever f_r in round 2. This is more costly as it requires two retrieval indexes and models. For all, we use the following hyperparameters: T = 2 retrieval rounds, verifier budget B = 100, context budget M = 3 for Contriever-MSMARCO and M = 6 for INF-Retriever and Qwen3-Embedding.

[4] `Alibaba-NLP/Tongyi-DeepResearch-30B-A3B`

[5] `PeterJinGo/SearchR1-nq_hotpotqa_train-qwen2.5-7b-em-ppo`

Table 1: Main Experimental Results on QAMPARI test set (N=1000). We report MRecall@100 (MR) and Recall@100 (R) using Qwen3-30B as our verifier. Our proposed methods outperform both the base retrievers (Base) and retrievers fine-tuned in-domain (FT (D_i)). Agentic baselines are reported with FT(D_i) as the retriever. RVR approaches also achieve much higher performance than agentic baselines. For our results, statistical significance is tested using bootstrap resampling with 10,000 trials at $\alpha = 0.05$. † indicates statistically significant improvement over FT(D_i). * indicates statistically significant improvement over FT(D_i) + FT(D_i).

	Base Retriever Model					
	Contriever		**Qwen3**		**INF**	
	MR@100	R@100	MR@100	R@100	MR@100	R@100
Base f_i	19.00	54.17	16.70	52.94	26.10	62.34
FT (D_i)	28.60	63.19	26.90	63.48	29.30	65.99
Tongyi	9.80	42.09	22.00	60.67	21.30	58.13
SearchR1	9.60	40.83	24.30	57.82	27.00	60.96
SearchR1 (Trained on QAMPARI)	6.10	38.70	24.50	59.45	30.00	64.21
MMR (λ=0.3)	13.30	53.42	17.30	57.56	20.10	60.14
MMR (λ=0.7)	27.20	63.73	25.30	63.06	28.10	65.54
Query Expansion	28.90	63.86	25.80	61.58	28.40	63.41
Instruction-Aware Retrieval	29.50	63.79	27.80	64.24	30.30	66.16
Ours: Retrieve-Verify-Retrieve						
FT (D_i) + FT (D_i)	28.80	63.59	30.30†	66.80†	31.10†	66.76†
FT (D_i + D_r) + FT (D_i + D_r)	**31.70**$^{\dagger *}$	66.12$^{\dagger *}$	29.20†	65.73†	32.40†	68.04†
FT (D_i) + FT (D_r)	31.60$^{\dagger *}$	**66.83**$^{\dagger *}$	**31.40**$^{\dagger *}$	**67.28**$^{\dagger *}$	**33.70**$^{\dagger *}$	**68.70**$^{\dagger *}$

5 Results

Task Performance Table 1 reports retrieval performance on the QAMPARI test set. Fine-tuning with in-domain data improves raw performance significantly, especially on weaker base retrievers (Contriever, Qwen) to be on par with the stronger base model (INF-Retriever). In-domain retrievers improve the agentic approaches, but overall, the agentic approaches underperform the baseline across all base models, even when paired with the fine-tuned retriever. Both Tongyi and SearchR1 are originally trained on single-answer multi-hop tasks (NQ, HotpotQA) rather than multi-answer coverage. The SearchR1 trained in-domain shows some gains with stronger base retrievers, yet still underperforms our method. MMR underperforms RVR and degrades MRecall as λ decreases, indicating that diversity in embedding space does not correspond to answer diversity. Query expansion and instruction-aware retrieval closely match FT(D_i) but do not reach RVR performance, highlighting the advantage of iterative conditioning over single-pass diversification strategies.

Our approach outperforms all comparison systems for all base retrieval settings. Even using the same fine-tuned retriever D_i as the baseline, verification step can bring gains when paired with stronger base retrievers (Qwen3, INF). Training the retriever to collect complementary output D_r and using separate models (Fine-tuned (D_i) + Fine-tuned (D_r)) both show further improvements.[6]

Efficiency Table 2 reports efficiency in terms of two main metrics: latency (seconds per query (s/q) for both retrieval and verification/query generation components) and memory requirements (GPU memory usage in gigabytes). We measure latency on a single NVIDIA H200 GPU. Retrieval time includes query encoding and k-nearest-neighbor search over the document index. Verification time depends on the verifier budget B (number of documents verified per query) for our system, and the agentic search query generation process for agentic models.

The single-pass baseline requires the fewest resources, while agentic search is substantially slower. Two-round RVR incurs additional overhead scaling with the verifier budget, making it 2–3$\times$ slower

[6]We focus on the retrieval metrics, as described in Section 2. We additionally present results on downstream question answering performance, which correlate with retrieval metrics, in Appendix C14.

Table 2: **Efficiency and memory comparison** on QAMPARI test set using Qwen3-30B as the verifier. *Left*: number of retrieval calls (↓) and total wall-clock time in seconds/query (↓); RVR is significantly more efficient than agentic baselines. See Appendix C1 for a per-component breakdown. *Right*: memory requirements (all in GB, ↓) for LLM and retrieval index; retriever model sizes (GB) are in parentheses. SearchR1 (Q): SearchR1 further trained on QAMPARI. We report two configurations of our system, RVR 1 (w/ T=2, B=1, f_i=f_r) and RVR 2 (w/ T=2, B=100, $f_i \neq f_r$).

	Latency (# calls, seconds per query)						**Memory (LLM, Retrieval Index)**					
	Contriever		**Qwen3**		**INF**		**Contr. (0.48)**		**Qwen3 (1.50)**		**INF (5.43)**	
Base	**1**	**1.91**	**1**	**1.34**	**1**	**3.75**		73.98	-	98.6	-	147.9
Tongyi	14.9	190.0	15.85	316.39	16.42	344.7	61.1	74.30	61.06	99.3	61.0	139.0
SearchR1	2.92	21.84	2.96	25.38	2.93	42.40	17.41	73.98	17.4	98.6	17.4	147.9
SearchR1 (Q)	2.04	20.96	3.40	25.82	3.39	42.86	17.41	73.98	17.4	98.6	17.4	147.9
RVR 1	2	3.87	2	5.11	2	7.62	56.93	73.99	56.9	98.6	56.9	147.9
RVR 2	2	4.78	2	6.02	2	8.52	56.93	147.9	56.9	197.3	56.9	295.9

Table 3: **Out-of-Domain Generalization.** MRecall@100 on QUEST (N=1727) and WebQSP (N=1639) test sets using Qwen3-30B as the verifier. For our results, statistical significance is tested using bootstrap resampling with 10,000 trials at $\alpha = 0.05$. † indicates statistically significant improvement over Base. * indicates statistically significant improvement over Base + Base. See Appendix C2 for Recall@100.

	QUEST			**WebQSP**		
	Contriever	**Qwen3**	**INF**	**Contriever**	**Qwen3**	**INF**
Base	3.24	3.13	4.75	62.16	61.00	62.47
FT (D_i)	3.13	2.43	4.75	49.45	46.68	51.60
Tongyi	0.93	2.49	3.30	52.33	57.56	58.29
SearchR1	0.81	3.19	3.53	54.91	61.49	61.98
SearchR1 (Trained on QAMPARI)	1.33	3.82	4.57	38.45	48.53	59.89
Ours: Retrieve-Verify-Retrieve						
Base + Base	3.42	3.30	4.81	**62.96**†	**62.53**†	**63.21**†
Base + FT (D_r)	**4.52**$^{\dagger *}$	**4.40**$^{\dagger *}$	**6.02**$^{\dagger *}$	61.49	60.81	62.72

than the baseline. Both RVR and agentic baselines require additional GPU memory. Using separate f_i and f_r models doubles the retrieval index size.

Generalization to Other Datasets We evaluate our models on two out-of-domain datasets: QUEST [21] and WebQSP [34]. Fine-tuned f_i underperforms off-the-shelf models here, potentially due to domain shift. Thus, we use the base retriever for the initial retrieval in our RVR framework. We evaluate two settings: (1) using the off-the-shelf retriever for both initial and subsequent retrieval (Base + Base), and (2) using the base retriever for initial retrieval and a subsequent retriever D_r fine-tuned on QAMPARI (Base + FT(D_r)). Table 3 reports results with B=100, while Table 11 in the Appendix shows similar trends with B=50. Our RVR framework largely outperforms the one-round baseline across both datasets. For QUEST, Base + FT(D_r) achieves the strongest performance, while for WebQSP it performs slightly worse than Base + Base. We attribute this to how much of the answer set the initial retrieval leaves uncovered: FT(D_r) is trained to retrieve documents complementary to already-covered answers, so its value depends on the residual coverage. On QUEST, base coverage is low (MRecall@100 below 5) and many answers remain to be found, so FT(D_r) helps substantially despite being fine-tuned on QAMPARI. On WebQSP, base coverage is already high (MRecall@100 above 61) and queries have the fewest answers of the three datasets (8.75, vs. 14.43 for QAMPARI and 10.50 for QUEST; Table 8), leaving little to recover, so the QAMPARI-specific bias of FT(D_r) can slightly outweigh its benefit. The transfer of FT(D_r) depends on base coverage: it helps most when many answers remain uncovered.

Table 4: RVR combined with LLM reranking on QAMPARI. Reranking retrieves N documents and uses Qwen3-30B, the same LM as the verifier, to select the top 100. RVR with reranking outperforms both standalone approaches. Reranking alone plateaus around 33.1 MRecall regardless of budget.

	Contriever		**Qwen3**		**INF**	
	MR@100	R@100	MR@100	R@100	MR@100	R@100
FT(D_i)	28.60	63.19	26.90	63.48	29.30	65.99
Rerank (N=200)	32.10	66.55	29.70	66.00	32.10	67.70
Rerank (N=500)	33.10	68.09	31.50	67.37	32.50	68.58
RVR	31.60	66.83	31.40	67.28	33.70	68.70
RVR w/ Rerank (N=500)	**33.10**	**68.31**	**32.10**	**68.02**	**34.90**	**69.84**

Multi-turn Setting So far, we have reported results with a single extra search turn. Figure 1(b) compares the performance across five iterations with three verifiers. With Qwen3-30B, gains plateau after the second iteration, as the LLM verifier tends to select redundant documents in later iterations even when the retrieved set contains documents with unique answers. Stronger verifiers yield two key improvements: overall performance increases, and gains continue beyond T=2. Qwen3.5-122B yields meaningful improvements at T=3 across all base retrievers, with continued gains at T=5. GPT-5-nano (evaluated on INF only due to cost) similarly shows notable improvements beyond T=2. Oracle verifier results (Appendix C4) confirm that gains can continue steadily up to Turn 5 given perfect verification, indicating substantial headroom for better verification mechanisms. These results confirm that verifier quality is a key bottleneck for higher performance, and that RVR's multi-turn design can be meaningfully exploited beyond two turns with stronger verifiers.

Robustness of Our Method We experiment how RVR's gain is consistent across various experimental settings.

- **Different Retrieval Architecture:** We experiment with ColBERT [16], a costly yet performant multi-vector retriever, as the base retriever. Since ColBERT does not support long input, we expand queries using keywords from verified documents rather than full document text. RVR shows gains on top of base ColBERT, demonstrating generality across retriever architectures (Appendix C12).
- **Reciprocal Rank Fusion (RRF) [9]:** We run experiments with RRF (k=60), which leverages two retrievers at each round. The RVR + RRF system shows increased gains over both RVR and RRF individually, demonstrating that the two methods are orthogonal. Full results are in Appendix C7.
- **Reranking:** We experiment with applying reranking, first retrieving N candidate documents and Qwen3-30B selects the top 100. We evaluate $N \in \{200, 500\}$. Pairing RVR with reranking further improves performance, highlighting the complementary nature of the two approaches. The time costs and the reranking prompt are in Appendix C9 and D9, respectively.
- **Varying output size** RVR improvements are also consistent with smaller output sets; results at K=20 and K=50 show similar gains, with relatively larger improvements at smaller K (Appendix C8).

Ablating Input to Subsequent Retriever If the performance gain stems exclusively from the verified documents retrieved in the earlier rounds, changing the input to the subsequent retriever should have little effect on overall performance. Therefore, we ablate the input to the subsequent retriever, in a two-turn setting (T=2). We present the results in Table 5. "Random relevant documents" denotes documents sampled randomly from the Turn 1 retrieval set of the same query; "Irrelevant documents" denotes documents pulled from the Turn 1 retrieval set of *other* queries. We also experiment with masking answer strings in the verified documents, and with keeping only the answer strings alongside the query. Results show that performance falls off as the context becomes less relevant (random relevant or irrelevant documents), showing that the subsequent retrieval does not rely on added noisy context only. Using only answer strings hurts performance, possibly because short answer strings differ in surface form from the training data. Masking the answer mentions does not hurt the performance, potentially as the answer mentions can be inferred from the document. By comparing RVR with answer strings masked versus only retaining answer strings, we show that the subsequent retriever does not merely rely on verified answers.

Table 5: Ablation Results. We fix the retriever setting to be FT (D_i) + FT (D_r), the best system from Table 1, and vary the input to the subsequent retreival. The top row is the proposed RVR framework. RVR with answer string masked achieves the highest performance (roughly on par with RVR).

	Base Retriever Model					
	Contriever		Qwen3		INF	
	MR@100	R@100	MR@100	R@100	MR@100	R@100
RVR (Query + Verified Documents)	31.60	66.83	31.40	67.28	**33.70**	68.70
RVR (Query + Verified Documents), with Answer String MASKED	**31.80**	**66.84**	**31.90**	**68.13**	**33.70**	**68.73**
Query + Answer String Only	30.80	66.00	29.50	66.57	32.90	68.53
Query + Random Relevant Documents	30.60	65.22	28.00	64.58	31.30	66.55
Query + Irrelevant Documents	29.00	64.14	26.80	62.65	28.80	64.40

Table 6: Verifier Model Intrinsic Evaluation Average verifier performance on top-100 retrieved documents on QAMPARI test set. Qwen3-30B-A3B-Inst-2507 achieves the highest recall and is used as our main verifier.

VERIFIER	PREC.	REC.	ACC.
RANDOM BASELINE	6.71	6.42	64.85
GPT-5-NANO	50.25	54.70	82.29
QWEN3-4B-INST	43.62	52.57	77.63
QWEN3-30B-INST	34.06	74.05	68.43

Table 7: Verifier ablation: Comparing LLM verifier (Qwen3-30B) with ORACLE (prioritizing documents with answer strings) and TOPK (initial retriever selection) on QAMPARI. MR@100 with FT(D_i) + FT(D_r), B=100. LLM verifier approaches the oracle and outperforms TOPK.

VERIFIER	CONTRIEVER	QWEN3	INF
ORACLE	**33.80**	**33.60**	**36.30**
LLM	31.60	31.40	33.70
TOPK	26.90	27.50	28.50

6 Analysis

Intrinsic Verifier Evaluation The goal of the verifier is to identify retrieved documents that are relevant to the original query. To evaluate its performance, we generate the following data: for each query in QAMPARI test set, we use three fine-tuned retrieval models f_i from different base retrievers to retrieve 100 documents. We judge whether they contain a gold answer using the gold label set. For this set, 21.07% of documents are positive and 78.93% are negative. We report the performance of three LLMs, GPT-5-nano, Qwen3-4B-Instruct-2507 [27], and Qwen3-30B-A3B-Instruct-2507 [27], as verifiers on this dataset. Table 6 presents the results. Qwen3-30B-A3B-Instruct-2507 model achieves the highest recall, which aligns with our objective of maximizing retrieval coverage. Therefore, we use this as the verifier for the main experiments.

Extrinsic Verifier Evaluation We isolate the impact of verifier performance on end-to-end retrieval performance. We evaluate verifier performance by comparing against a baseline and an upper bound. As a baseline (TopK), we use the top M documents ranked by the initial retriever to form the query context without a verifier. The final output set (K=100) combines the top 50 documents from the initial retriever with the top 50 non-duplicate documents from the subsequent retriever. As an upper bound, we use an oracle verifier that marks a document as gold if it contains any gold answer string.

Table 7 presents the comparison across verifier settings: Oracle, LLM (Qwen3-30B), and TopK. Across all base models, we see noticeable gains with the oracle verifier, suggesting performance gains can be achieved by improving the verifier. We also find that a smaller verifier (Qwen3-4B) achieves comparable performance to Qwen3-30B, suggesting RVR is robust to verifier size and practitioners can substitute a smaller verifier at lower memory cost (Appendix C3).

The Impact of Verifier Budget Figure 2 plots the system performance for varying verifier budget (10, 20, 50, 100 documents) to evaluate models under more resource-constrained settings. The scores for the 100-document setting are the same as reported in Table 1. We compare two settings of our Retrieve-Verify-Retrieve, one with FT(D_i) and another with FT(D_i) and FT(D_r). Across all models, the absolute performance decreases as the verifier budget shrinks. Using the fine-tuned subsequent retriever FT(D_r) is particularly helpful in more restrictive budget setting.

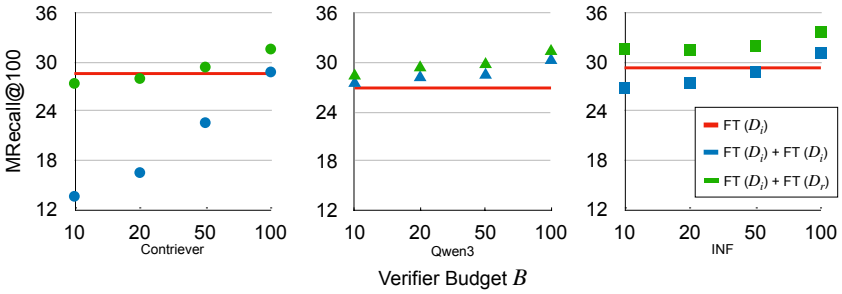

Figure 2: Varying the verifier budget with Qwen3-30B verifier. We evaluate MR@100 across three new verifier budgets on QAMPARI dataset. RVR is shown in green and blue, while our one-round baseline in red. See Appendix C5 for Recall@100 results.

The Contribution of 1st Turn vs. 2nd Turn Retrieval We analyze the individual contributions of our first-stage (base) and second-stage (iterative) retrievers. Across all models, the first stage contributes approximately 50 gold documents, corresponding to 7 unique answers, per question on average, demonstrating that the initial retriever already captures a substantial portion of relevant information. The second-stage iterative retriever, conditioned on verified documents from the first stage, contributes an additional 22-26 gold documents, up to 1 unique answer,[7] per question. This consistent improvement demonstrates that conditioning on previously retrieved context enables the model to discover initially missed documents. The full results are in Table 22 and Appendix C10.

The Impact of Varying Input Length We study the impact of varying the maximum number of input documents provided to the subsequent retriever. We do not report results for Contriever due to its sequence length limit of 512 tokens, resulting in truncation when we provide more than 3 input documents. We run inference with the RVR FT (D_i) + FT (D_r) setting, where FT (D_r) is fine-tuned with a maximum of 3, 6, and 12 input documents. We further vary the context budget, M, used at inference time. Figure 3 shows MRecall@100 results, and Figure 6 displays Recall@100 trends, as we sample from 0 up to 12 input documents.

We find that increasing the number of documents at inference time beyond 6 does not show strong gains. We also observe that on average, models fine-tuned with up to 6 input documents perform better.

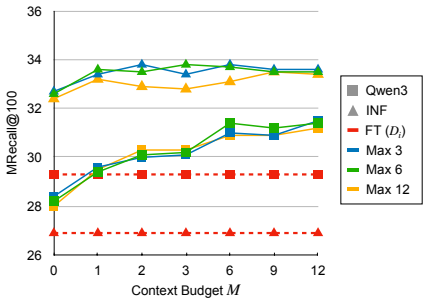

Figure 3: MRecall@100 with varying context budget M at inference time. Colors denote fine-tuning with different max document counts (3, 6, 12). Recall@100 is in Appendix C6.

7 Conclusion

Despite progress in retriever architectures and agentic search, comprehensively recovering relevant yet distinct documents remains challenging. To address this, we introduced Retrieve-Verify-Retrieve (RVR), a framework that conditions on previously retrieved evidence and explicitly optimizes for answer coverage. By training retrievers to predict missing gold documents and integrating verifier-guided iteration, our approach consistently expands retrieval coverage while reducing redundancy. Results on QAMPARI and zero-shot generalization to QUEST and WebQSP demonstrate that iterative conditioning provides a robust and general mechanism for improving retrieval completeness. Our work suggests existing retrieval frameworks fall short in addressing multi-answer queries, and fine-tuning retrievers for this scenario as well as designing an efficient framework to leverage outputs from previous retrieval rounds can improve performance. Future work can explore agentic methods that are targeted towards comprehensive retrieval. Another promising direction is integrating a complement-seeking retriever as a tool into the agentic search paradigm.

[7] These account for 10.4 to 13.2% of answers newly found in the second round.

Acknowledgments and Disclosure of Funding

We thank the reviewers for helpful discussions. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise. The work is partially funded by NSF CAREER award 2443271 and NSF award RI-2521091.

References

[1] S. Amouyal, T. Wolfson, O. Rubin, O. Yoran, J. Herzig, and J. Berant. QAMPARI: A benchmark for open-domain questions with many answers. In S. Gehrmann, A. Wang, J. Sedoc, E. Clark, K. Dhole, K. R. Chandu, E. Santus, and H. Sedghamiz, editors, *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 97–110, Singapore, Dec. 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.gem-1.9/.

[2] A. Asai, T. Schick, P. Lewis, X. Chen, G. Izacard, S. Riedel, H. Hajishirzi, and W. tau Yih. Task-aware retrieval with instructions. *ArXiv*, abs/2211.09260, 2022. URL https://api.semanticscholar.org/CorpusID:253581733.

[3] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

[4] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA, 1998. Association for Computing Machinery. ISBN 1581130155. doi: 10.1145/290941.291025. URL https://doi.org/10.1145/290941.291025.

[5] H.-T. Chen and E. Choi. Open-world evaluation for retrieving diverse perspectives. In L. Chiruzzo, A. Ritter, and L. Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8508–8528, Albuquerque, New Mexico, Apr. 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.431. URL https://aclanthology.org/2025.naacl-long.431/.

[6] H.-T. Chen, X. Liu, S. Ravfogel, and E. Choi. Beyond single embeddings: Capturing diverse targets with multi-query retrieval. *arXiv preprint arXiv:2511.02770*, 2025.

[7] Z. Chen, X. Ma, S. Zhuang, P. Nie, K. Zou, A. Liu, J. Green, K. Patel, R. Meng, M. Su, S. Sharifymoghaddam, Y. Li, H. Hong, X. Shi, X. Liu, N. Thakur, C. Zhang, L. Gao, W. Chen, and J. Lin. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent, 2025. URL https://arxiv.org/abs/2508.06600.

[8] Z. Chen, X. Ma, S. Zhuang, J. Lin, A. Asai, and V. Zhong. AgentIR: Reasoning-aware retrieval for deep research agents. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=rn8kjxilhF.

[9] G. V. Cormack, C. L. Clarke, and S. Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759, 2009.

[10] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston. Chain-of-verification reduces hallucination in large language models. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.212. URL https://aclanthology.org/2024.findings-acl.212/.

[11] J. Fang, Z. Meng, and C. MacDonald. KiRAG: Knowledge-driven iterative retriever for enhancing retrieval-augmented generation. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18969–18985, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.929. URL https://aclanthology.org/2025.acl-long.929/.

[12] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=jKN1pXi7b0.

[13] B. Jin, H. Zeng, Z. Yue, J. Yoon, S. Arik, D. Wang, H. Zamani, and J. Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. In *Conference on Language Modeling*, 2025.

[14] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. Dense passage retrieval for open-domain question answering. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

[15] U. Katz, M. Vetzler, A. Cohen, and Y. Goldberg. NERetrieve: Dataset for next generation named entity recognition and retrieval. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3340–3354, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.218. URL https://aclanthology.org/2023.findings-emnlp.218/.

[16] O. Khattab and M. Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020.

[17] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

[18] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[19] W. Liu, X. Ma, Y. Zhu, Y. Li, D. Shi, D. Yin, and Z. Dou. Agentic-R: Learning to retrieve for agentic search. In M. Liakata, V. P. Moreira, J. Zhang, and D. Jurgens, editors, *Findings of the Association for Computational Linguistics: ACL 2026*, pages 15987–16005, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.785. URL https://aclanthology.org/2026.findings-acl.785/.

[20] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[21] C. Malaviya, P. Shaw, M.-W. Chang, K. Lee, and K. Toutanova. QUEST: A retrieval dataset of entity-seeking queries with implicit set operations. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14032–14047, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.784. URL https://aclanthology.org/2023.acl-long.784/.

[22] S. Min, K. Lee, M.-W. Chang, K. Toutanova, and H. Hajishirzi. Joint passage ranking for diverse multi-answer retrieval. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6997–7008, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.560. URL https://aclanthology.org/2021.emnlp-main.560/.

[23] P. Qi, X. Lin, L. Mehr, Z. Wang, and C. D. Manning. Answering complex open-domain questions through iterative query generation. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *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 2590–2602, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1261. URL https://aclanthology.org/D19-1261/.

[24] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[25] R. Shao, A. Asai, S. Z. Shen, H. Ivison, V. Kishore, J. Zhuo, X. Zhao, M. Park, S. G. Finlayson, D. Sontag, et al. Dr tulu: Reinforcement learning with evolving rubrics for deep research. *arXiv preprint arXiv:2511.19399*, 2025.

[26] S. Sharifymoghaddam and J. Lin. Rerank before you reason: Analyzing reranking tradeoffs through effective token cost in deep search agents. *arXiv preprint arXiv:2601.14224*, 2026.

[27] Q. Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[28] T. D. Team, B. Li, B. Zhang, D. Zhang, F. Huang, G. Li, G. Chen, H. Yin, J. Wu, J. Zhou, et al. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*, 2025.

[29] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

[30] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.557. URL https://aclanthology.org/2023.acl-long.557/.

[31] W. Xiong, X. Li, S. Iyer, J. Du, P. Lewis, W. Y. Wang, Y. Mehdad, S. Yih, S. Riedel, D. Kiela, and B. Oguz. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=EMHoBG0avc1.

[32] J. Yang, J. Wan, Y. Yao, W. Chu, Y. Xu, E. Wang, and Y. Qi. inf-retriever-v1 (revision 5f469d7), 2025. URL https://huggingface.co/infly/inf-retriever-v1.

[33] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning. 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, 2018.

[34] W.-t. Yih, M. Richardson, C. Meek, M.-W. Chang, and J. Suh. The value of semantic parse labeling for knowledge base question answering. In K. Erk and N. A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2033. URL https://aclanthology.org/P16-2033/.

[35] Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin, F. Huang, and J. Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

[36] Y. Zhou, S. Dai, C. Qu, L. Pang, J. Xu, and J.-R. Wen. Learning to retrieve from agent trajectories. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '26, page 2544–2555, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400725999. doi: 10.1145/3805712.3809675. URL https://doi.org/10.1145/3805712.3809675.

[37] A. Zhu, A. Hwang, L. Dugan, and C. Callison-Burch. FanOutQA: A multi-hop, multi-document question answering benchmark for large language models. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 18–37, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-short.2. URL https://aclanthology.org/2024.acl-short.2/.

A Limitations

We discuss several limitations of our work. First, while our LLM verifier enables iterative gains, performance saturates after Turn 2 with the current Qwen3-30B verifier. Experiments with stronger verifiers (Qwen3.5-122B, GPT-5-nano) show continued gains beyond T=2 (Figure 1), indicating that verifier quality is the primary bottleneck. Second, our optimal configuration with separate f_i and f_r models requires more GPU memory (up to 300GB for INF). However, a single-model configuration achieves comparable performance at roughly half the cost ($\sim$130GB), comparable to agentic baselines. Third, our evaluation is limited to list-style questions with well-defined gold answer sets using Wikipedia corpora. Extending to other retrieval scenarios such as research paper retrieval or web search would require domain-specific training data and appropriate evaluation metrics, which we leave to future work. Designing a framework that can work for various types of questions, including multi-hop, list-style, and subjective questions, would be another promising direction for future work.

B Dataset Details

B1 QAMPARI

Table 8 displays our dataset statistics in detail. An example of a gold document from our QAMPARI test set is shown below, along with the query and answer strings.

Query: What manga was drawn by Ryoichi Ikegami?

Answer Strings: Heat, ***Mai, the Psychic Girl***, Wounded Man, Sanctuary, Crying Freeman, Strain

Document Text: ***Mai, the Psychic Girl***, known simply as in Japan, is a manga written by Kazuya Kudo and illustrated by Ryoichi Ikegami. The main character is Mai Kuju, a 14-year-old Japanese girl with powerful psychic abilities. She is being pursued by the Wisdom Alliance, an organization which secretly strives to control the world. The alliance already controls four other powerful psychic children, and it has hired the Kaieda Intelligence Agency to capture Mai. Media. Manga. ***Mai, the Psychic Girl*** is one of the first manga series to be fully published in English.

B2 QUEST

Below, we show an example query from the QUEST test set, its associated answer string, and a document from our corpus that contains this answer. Table 8 shows additional dataset statistics.

Query: what are some 1950s comedy mystery or spy comedy films

Answer Strings: The Hole (1957 film), Charade (1953 film), Hot Stuff (1956 film), The Trouble with Harry, Father Brown (film), The Runaway Bus, Boston Quackie, ***My Favorite Spy***, The Fuller Brush Girl, Scared Stiff (1953 film), Our Man in Havana (film), Spy Chasers, Mrs. O'Malley and Mr. Malone, Clipped Wings (1953 film), Commotion on the Ocean, Down Among the Z Men, Knock on Wood (film), Top Secret (1952 film)

Document Text: ***My Favorite Spy*** is a 1951 comedy film directed by Norman Z. McLeod and starring Bob Hope and Hedy Lamarr. Plot. US intelligence agents recruit burlesque comic Peanuts White to pose as international spy Eric Augustine, whom he resembles, to acquire a million-dollar microfilm

Table 8: Dataset statistics showing the number of questions split across train, development, and test sets, along with the average number of answers and gold documents per question.

	# examples			# answers	# gold docs
Dataset	Train	Dev	Test	Avg.	Avg.
QAMPARI	34,425	6,191	1,000	14.43	16.58
QUEST	—	—	1,727	10.50	—
WebQSP	—	—	1,639	8.75	—

Table 9: Efficiency breakdown showing retrieval (Ret.) and verification (Ver.) time in seconds per query. See Table 2 in the main paper for # Calls and total time (Sum).

	Contriever		Qwen3		INF	
	Ret.↓	Ver.↓	Ret.↓	Ver.↓	Ret.↓	Ver.↓
Base	**1.91**	-	**1.34**	-	**3.75**	-
Tongyi	97.72	92.32	232.79	83.60	264.30	80.40
SearchR1	19.43	2.41	22.95	2.42	39.86	2.54
RVR (w/ T=2, B=1)	3.80	0.07	5.05	0.06	7.55	0.07
RVR (w/ T=2, B=50)	3.80	0.53	5.05	0.52	7.55	0.53
RVR (w/ T=2, B=100)	3.80	0.98	5.05	0.97	7.55	0.97

in Tangier. There, he encounters the irresistible Lily Dalbray, Augustine's one-time friend, who is now in league with his arch-enemy, Brubaker.

B3 WebQSP

Below, we show an example query from the WebQSP test set, its associated answer string, and a document from our corpus that contains this answer. Table 8 shows additional dataset statistics.

Query: what does jamaican people speak

Answer Strings: ***Jamaican English***, Jamaican Creole English Language

Document Text: In ***Jamaican English***, normally reduced English vowels are sometimes not reduced, and other times are hyper-reduced, so that ẗokenïs not but, yet c̈ementc̈an be as reduced as; the exact nuances of the rules at play here are also highly debated. Language use: Jamaican Standard English versus Patois. Jamaican Standard English and Jamaican Patois exist together in a post-creole speech continuum. Jamaican (Creole/Patois) is used by most people for everyday, informal situations - it is the language most Jamaicans use at home and are most familiar with, as well as the language of most local popular music.

B4 Instruction Used for Instruction Tuned Models

We use the same instruction to fine-tune our Qwen3-0.6B-Embed and inf-retriever-1.5B models: *Given a query, retrieve relevant passages that answer the query*. We use this instruction during both fine-tuning and inference.

C Additional Results

C1 Full Efficiency Breakdown

Table 9 reports the per-query retrieval (Ret.) and verification (Ver.) time breakdown. See Table 2 in the main paper for # Calls and total time.

Table 10: Recall@100 out-of-domain generalization results on QUEST (N=1727) and WebQSP (N=1639) with verifier budget 100. † indicates statistically significant improvement over Base. * indicates statistically significant improvement over Base + Base. "SearchR1 (Trained on QAMPARI)" indicates SearchR1 retrained on QAMPARI using PPO with a reward for retrieving any correct answer.

	QUEST			**WebQSP**		
	Contriever	**Qwen3**	**INF**	**Contriever**	**Qwen3**	**INF**
Base	23.79	21.38	26.60	77.38	76.06	77.39
FT (D_i)	18.67	18.38	26.31	65.72	61.50	67.06
Tongyi	10.51	19.52	21.95	65.84	71.80	72.37
SearchR1	9.54	20.83	23.01	68.92	76.67	76.82
SearchR1 (Trained on QAMPARI)	9.89	19.40	24.75	51.74	63.53	75.23
Ours: Retrieve-Verify-Retrieve						
Base + Base	24.85†	22.67†	27.21†	**77.93**†	**77.54**†	**78.20**†
Base + FT (D_r)	**26.01**$^{\dagger *}$	**25.43**$^{\dagger *}$	**30.53**$^{\dagger *}$	76.81	76.28	77.91

Table 11: **Results with Verifier Budget of 50.** Results on QUEST (N=1727) and WebQSP (N=1639) test sets. We report MRecall@100 (MR) and Recall@100 (R). We report performances of different base retrieval models and RVR methods. For RVR, we use a verifier budget of 50. For our results, statistical significance is tested using bootstrap resampling with 10,000 trials at $\alpha = 0.05$. † indicates statistically significant improvement over Base. * indicates statistically significant improvement over Base + Base

.

	QUEST						**WebQSP**					
	Contriever		**Qwen3**		**INF**		**Contriever**		**Qwen3**		**INF**	
	MR	R	MR	R	MR	R	MR	R	MR	R	MR	R
Base	3.24	23.79	3.13	21.38	4.75	26.60	62.16	**77.38**	61.00	76.06	62.47	77.39
FT (D_i)	3.13	18.67	2.43	18.38	4.75	26.31	49.45	65.72	46.68	61.50	51.60	67.06
Ours: Retrieve-Verify-Retrieve												
Base + Base	2.72	21.98	2.95	20.21	4.11	24.47	**62.22**	76.94	**61.73**	**76.93**†	**63.15**	**77.90**†
Base + FT (D_r)	**3.82***	**22.89***	**4.05**$^{\dagger *}$	**23.88**$^{\dagger *}$	**5.73**$^{\dagger *}$	**28.71**$^{\dagger *}$	60.07	75.51	60.14	75.56	62.47	77.48

C2 Out-of-Domain Results

Table 10 reports Recall@100 for all methods on QUEST and WebQSP. Table 11 reports MRecall@100 and Recall@100 at verifier budget B=50.

C3 Smaller Verifier Results

Table 12 compares Qwen3-4B (higher precision, lower recall) against Qwen3-30B (lower precision, higher recall) as verifiers using the INF retriever. The two perform comparably, demonstrating that RVR is robust to verifier size.

C4 Multi-turn Results with Oracle Verifier

Figure 4 illustrates MRecall@100 and Recall@100 across five iterations for FT(D_i) + FT(D_i) and FT(D_i) + FT(D_r) with a verifier budget of 100, comparing the LLM verifier (Qwen3-30B) against an oracle verifier that selects only documents containing gold answer strings. Performance with the LLM verifier plateaus after the second iteration for both configurations, whereas the oracle verifier shows continued improvement across all five iterations, indicating substantial headroom for better verification mechanisms.

Table 12: Comparison of smaller (Qwen3-4B) vs. larger (Qwen3-30B) verifier on QAMPARI test set using INF retriever. The smaller verifier performs comparably, demonstrating RVR is robust to verifier size.

	QWEN3-4B		QWEN3-30B	
	MR@100	R@100	MR@100	R@100
FT(D_i) + FT(D_r) , T=2	34.20	68.81	33.70	68.70
FT(D_i) + FT(D_r) , T=3	**35.00**	**69.08**	34.20	69.09

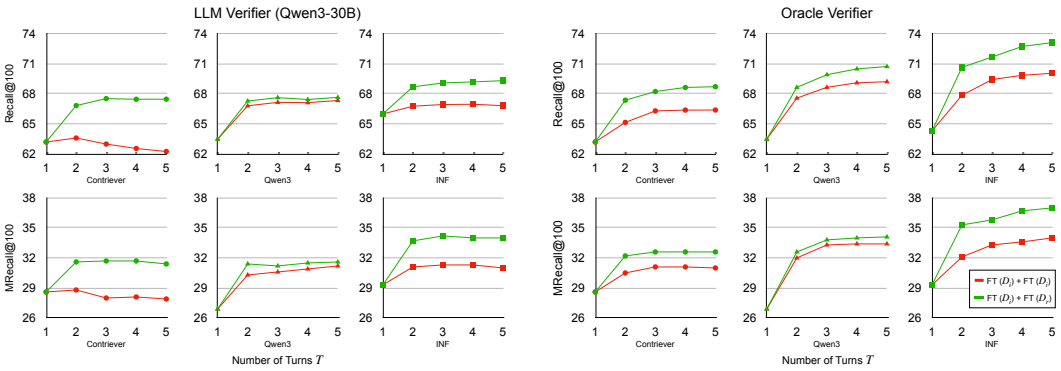

Figure 4: Multi-turn Generalization Results for FT(D_i) + FT(D_i) and FT(D_i) + FT(D_r). Recall@100 and MRecall@100 across five iterations with a verifier budget of 100. Left panels show results with LLM verifier (Qwen3-30B); right panels show results with oracle verifier that selects documents containing unique answer strings. Performance with the LLM verifier plateaus after the second iteration, whereas the oracle verifier shows continued improvement.

C5 Impact of Verifier Budget on Recall@K

Figure 5 displays Recall@100 with verifier budgets of 10, 20, 50, and 100.

C6 Impact of Varying Input Documents on Recall@K

Figure 6 displays additional results for our experiment where we vary the number of input documents at inference time.

C7 Retrieval Fusion (RRF)

Table 13 shows results combining RVR with Reciprocal Rank Fusion (RRF) on QAMPARI, using Contriever FT(D_i) and INF FT(D_i) as the two retrievers (k=60). We evaluate two fusion strategies. **FT(D_i) w/ RRF** fuses the first-round retrieval results from Contriever FT(D_i) and INF FT(D_i) directly using RRF, with no second retrieval round. **RVR w/ RRF** first fuses the initial FT(D_i) results from both retrievers using RRF, verifies the fused set, then runs a second retrieval round

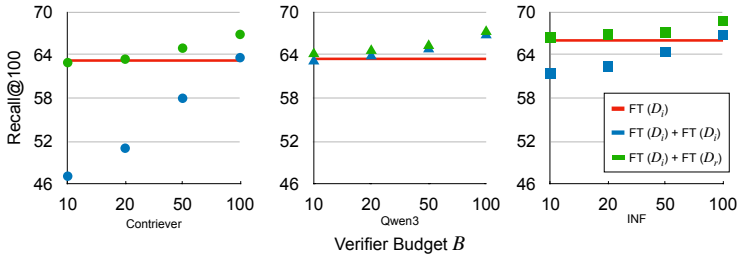

Figure 5: This figure shows the results of lowering verifier budget beyond 100 for Recall@100.

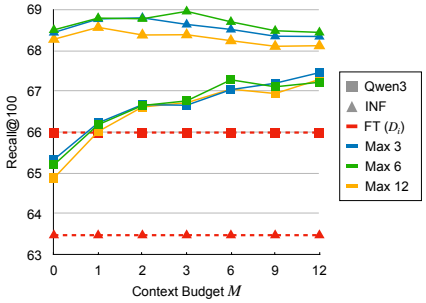

Figure 6: This figure shows the results of increasing the number of input documents on Recall@100 for INF and Qwen3 models fine-tuned with different maximum document counts: 3, 6, and 12. We evaluate across 7 different context budgets.

Table 13: RRF results on QAMPARI test set combining Contriever FT(D_i) and INF FT(D_i) (k=60). RVR and RRF are complementary, with their combination outperforming either alone.

	MR@100	R@100
FT(D_i) (Contriever)	28.60	63.19
FT(D_i) (INF)	29.30	65.99
FT(D_i) w/ RRF	32.80	67.89
RVR (Contriever)	31.60	66.83
RVR (INF)	33.70	68.70
RVR w/ RRF	**34.30**	**69.56**

conditioned on the verified documents using FT(D_r) from both Contriever and INF, and fuses the two second-round outputs using RRF again.

RRF alone improves over individual retrievers by ensembling complementary model biases, while RVR further targets answers not yet covered by conditioning the second round on verified documents. Their combination outperforms either method alone, confirming they address orthogonal failure modes. We note that RRF+RVR requires two distinct retriever architectures with separate indexes, while standalone RVR achieves comparable performance using a single retriever.

C8 Results at Smaller K

Table 14 reports MRecall and Recall at K=20 and K=50 on QAMPARI with verifier budget B=100. RVR improvements are consistent across all values of K, with relatively larger gains at smaller K, suggesting RVR is particularly effective in precision-critical settings.

C9 Time Cost Comparing RVR and LLM Reranking

Table 15 reports inference time costs of RVR and reranking of different budgets. Reranking at N=200 incurs 1.97 s/q in LLM time versus RVR's 0.98 s/q, making reranking roughly twice as expensive in

Table 14: Results at K=20 and K=50 on QAMPARI test set with verifier budget B=100. RVR consistently outperforms baselines across all K values, with relatively larger gains at smaller K.

	Contriever-MSMARCO				**Qwen3-Embedding-0.6B**				**INF-Retriever-v1-1.5B**			
	MR	R	MR	R	MR	R	MR	R	MR	R	MR	R
Base	14.80	47.40	9.90	36.75	13.90	45.48	11.10	35.96	19.80	54.51	12.90	42.30
FT(D_i)	20.70	53.96	12.00	39.41	19.00	54.26	10.30	41.22	21.80	56.61	12.20	43.48
FT(D_i) + FT(D_i)	24.40	58.50	15.20	46.33	23.10	59.75	13.90	47.02	25.10	60.13	14.90	47.78
FT(D_i) + FT(D_r)	**25.40**	**59.62**	**15.50**	**46.49**	**24.40**	**60.68**	**14.00**	**47.44**	**26.20**	**61.39**	**15.30**	**48.58**

Table 15: Time cost comparison (seconds per query) for RVR vs. LLM reranking pipelines on QAMPARI. Reranking at N=200 incurs roughly twice the LLM cost of RVR verification.

	LLM (s/q)	Retrieval (s/q)	Total (s/q)
FT(D_i)	0	2.47	2.47
Rerank (N=200)	1.97	2.83	4.80
Rerank (N=500)	4.01	2.47	6.48
RVR	0.98	5.47	6.45
RVR w/ Rerank (N=500)	4.92	8.13	13.05

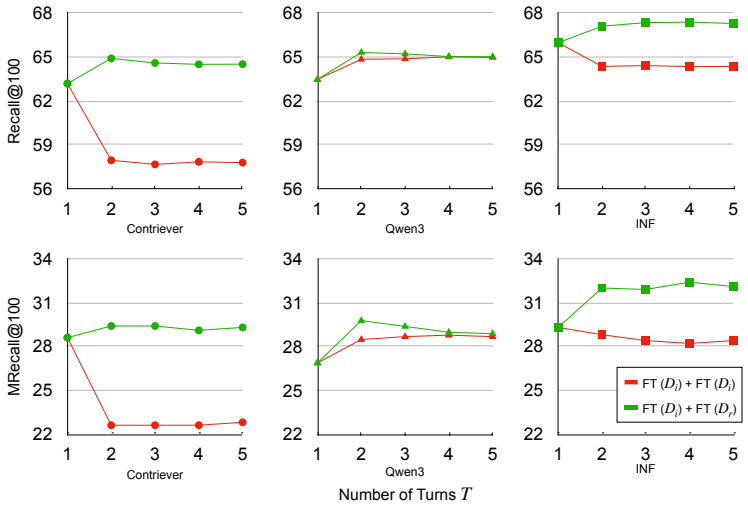

Figure 7: Multi-turn Generalization Results. This figure illustrates the change in Recall@100 and MRecall@100 across five iterations with a verifier budget of 50.

LLM calls at comparable total cost. In settings where LLM inference is the primary bottleneck, RVR provides a more cost-effective path to performance gains.

C10 Stage Contribution Analysis (Verifier Budget B=50)

Figure 7 shows multi-turn MRecall@100 and Recall@100 across five iterations at B=50. Table 16 reports oracle vs. LLM verifier performance at B=50. Table 23 shows the per-stage contribution at B=50, complementing the B=100 results (Table 22).

C11 Ablation Studies on Retrievers

Table 17 presents an ablation study of retriever components. In rows 1 and 2, we report using off-the-shelf base retriever instead of fine-tuned retriever for f_i. As long as we fine-tune the subsequent retriever, the performance is quite competitive, sometimes even outperforming the setting in which both initial and subsequent retrievers are fine-tuned.

Table 16: Comparing the performance of using oracle verifier vs. LLM verifier (Qwen3-30B) on QAMPARI test set. We evaluate MRecall@100 in RVR setting FT (D_i) + FT (D_r) with a verifier budget of 50.

VERIFIER	CONTRIEVER	QWEN3	INF
ORACLE	**29.60**	**30.50**	**33.50**
LLM (QWEN3-30B)	29.40	29.80	32.00
TOPK	26.90	27.50	28.50

Table 17: Ablation study on QAMPARI test set. Each row represents an RVR configuration, denoted in the format of f_i + f_r (initial + subsequent retriever). **Base**: off-the-shelf pretrained retriever; **FT(D_i)**: fine-tuned on initial retrieval data; **FT(D_r)**: fine-tuned on subsequent retrieval data with document context; **FT(D_i + D_r)**: fine-tuned on both objectives jointly.

	Contriever		**Qwen3**		**INF**	
	MR	R	MR	R	MR	R
Base + Base	21.30	58.10	20.20	58.12	28.50	65.29
FT(D_i) + FT(D_i)	28.80	63.59	30.30	66.80	31.10	66.76
FT(D_i + D_r) + FT(D_i + D_r)	**31.70**	66.12	29.20	65.73	32.40	68.04
Base + FT (D_r)	30.70	**67.25**	29.20	66.36	**34.20**	**69.29**
FT (D_i) + FT (D_r)	31.60	66.83	**31.40**	**67.28**	33.70	68.70

C12 ColBERT Results

We evaluate RVR with ColBERT [16] as the base retriever on QAMPARI. We fine-tune ColBERT on QAMPARI training data following the same procedure as our main retrievers (FT(D_i)). ColBERT is a multi-vector retrieval model that does not support long input queries; running the full evidence-augmented query causes out-of-memory errors. We therefore use keyword extraction from verified documents to expand the query rather than concatenating full document text. Table 18 shows that RVR improves over the ColBERT baseline in both MRecall and Recall, demonstrating that the framework generalizes to multi-vector retrievers. The keyword-based query expansion is a lightweight approximation; replacing it with full document context may yield further gains if memory constraints are addressed.

Table 18: RVR results on QAMPARI test set with ColBERT as the base retriever. Query expansion uses keywords from verified documents due to ColBERT's input length constraints.

	MR@100	**R@100**
Base	33.30	68.85
FT(D_i)	37.60	70.85
RVR Base + Base (keywords)	37.40	72.01
RVR FT(D_i) + FT(D_i) (keywords)	**38.50**	**72.43**

C13 Varying MMR λ

Table 19 reports MRecall@100 (MR) and Recall@100 (R) for MMR across all λ values. As λ increases toward 1.0, the diversity penalty diminishes and performance approaches the FT(D_i) baseline (λ=1.0). Lowering λ increases diversity in embedding space but consistently hurts both MR and R across all retrievers, confirming that embedding-space diversity does not correspond to answer coverage.

C14 Question Answering Evaluation Results

We evaluate retrieval exclusively, on the premise that a reader cannot recover an answer whose evidence was never retrieved, so coverage upper-bounds downstream answer recall. However, it is unclear if the higher retrieval coverage of the proposed framework will transfer to higher downstream question answering performance. To that end, we experiment with GPT-5-mini as the reader LM, providing it with the query and the top 100 document set retrieved by each method to generate a list of predicted answer strings. The prediction list is then evaluated against the list of ground truth answers. We report the results on a subset of 200 examples on QAMPARI in Table 20. **EM** stands for exact match, measuring if the list of predicted answers is exactly the same as the ground truth list. **Final Answer List Recall** measures the percentage of answers covered by the predicted list. **F1** measures the set-level F1 score, with both the prediction and ground truth as sets.

Table 19: MMR results on QAMPARI test set for varying diversity parameter λ. λ=1.0 (FT(D_i)) has no diversity penalty and is equivalent to the standard fine-tuned retriever baseline.

	Contriever		**Qwen3**		**INF**	
	MR	R	MR	R	MR	R
λ=0.1	10.40	49.58	15.80	55.85	15.80	57.93
λ=0.3	13.30	53.42	17.30	57.56	20.10	60.14
λ=0.5	19.50	59.47	21.90	60.45	25.30	63.65
λ=0.7	27.20	63.73	25.30	63.06	28.10	65.54
FT(D_i) λ=1.0	28.60	63.19	26.90	63.48	29.30	65.99

Table 20: Evaluation of downstream question answering results with a simple retrieval-augmented setting. Metric notations: **EM** (exact match) - whether the list of predicted answers exactly match the ground truth list; **Final Answer List Recall** - % of answers covered by the predicted list; **F1** - the set-level F1 score.

	Retrieval Method	**EM**	**Final Answer List Recall**	**F1**	**MR@100**
Contriever	FT (D_i)	6.5	38.07	36.31	28.60
	FT (D_i) + FT (D_i)	6.5	37.74	35.70	28.80
	FT (D_i) + FT (D_r)	6.0	**40.43**	**37.48**	**31.60**
Qwen3	FT (D_i)	6.5	39.83	37.82	26.90
	FT (D_i) + FT (D_i)	7.5	40.22	38.51	30.30
	FT (D_i) + FT (D_r)	8.0	**43.07**	**40.95**	**31.40**
INF	FT (D_i)	6.5	38.79	36.74	29.30
	FT (D_i) + FT (D_i)	6.0	40.86	38.50	31.10
	FT (D_i) + FT (D_r)	6.5	**41.84**	**38.76**	**33.70**

Results show that list recall and F1 correlate with correlates with retrieval coverage (MRecall @ 100). We ran a Spearman rank correlation to measure this further. Across all nine configurations, answer list recall is strongly rank correlated with retrieval MRecall@100 (Spearman ρ = 0.80, p = 0.038), and F1 shows the same positive trend (ρ = 0.62), indicating that improved document coverage translates into better final answers. Exact match accuracy does not track retrieval quality (ρ = 0.11, p = 0.855), as expected for a strict binary metric. The full RVR method (FT(D_i) + FT(D_r)) achieves the highest list recall and F1 in most settings, most clearly for Qwen3 (43.07 recall, 40.95 F1, versus 39.83 / 37.82 for the single-pass baseline). EM stays flat and noisy across the board, which is what we would expect from a list-EM metric that goes to zero whenever a single answer in the list is missed, so F1 is the more meaningful signal here. To summarize, **the improved coverage does carry through to the generated answers rather than staying confined to retrieval**.

D Experimental Details & Prompts

D1 Prompt for Qwen3 Verifier

The prompt we used to verify a document as gold using Qwen3-30B-Instruct is displayed in Figure 8.

D2 Agentic Baseline Inference Settings

We follow their original implementations to retrieve k_t = 5 documents for Tongyi and k_s = 3 documents for SearchR1, respectively. To obtain the final candidate document set, we combine the documents returned for each retriever call without duplicates. For each agent, we run until it outputs an answer or it has collected over K = 100 documents in total. If the agent has only collected K_a documents ($K_a < K$) upon termination, we additionally retrieve a set of $(K - K_a)$ documents using the last query issued to the retriever and append them to the candidate document list.

Table 21: Spearman correlation between QA metrics and retrieval performance (MRecall @ 100). Numbers are derived from results in Table 20.

QA Metric	p	Spearman ρ
Answer-List Recall	0.038	$+0.80$
F1	0.136	$+0.62$
EM	0.855	-0.11

D3 Prompt for SearchR1

The prompt used for SearchR1 is displayed in Figure 9.

D4 Prompt for Tongyi

The prompt used for Tongyi is displayed in Figure 10.

D5 Trajectory for SearchR1

An example trajectory trace for SearchR1 is provided in Figure 14. For this specific example, the retrieval model is the initial INF retriever f_i.

D6 Trajectory for Tongyi

A portion of the trajectory trace for Tongyi is provided in Figure 15. Same as above, the retrieval model is initial INF retriever f_i.

D7 Prompt for Query Expansion

The prompt used to generate an expanded query for the query expansion baseline is displayed in Figure 13.

D8 Prompt for Instruction-Aware Retrieval

The prompts used for instruction-aware retrieval are displayed in Figure 11. The first prompt extracts candidate answers from the initial retrieval set; the second prompt instructs the retriever to avoid documents covering those answers.

D9 Prompt for LLM Reranking

The prompt used to rerank candidate documents with Qwen3-30B is displayed in Figure 12.

E Impact Statement

This paper introduces a retrieval framework designed to improve answer coverage and reduce redundancy in information retrieval systems. By enabling retrieval models to iteratively reason over previously retrieved evidence, the proposed approach may improve the completeness and reliability of systems used in applications such as search engines, question answering, and knowledge discovery.

Potential positive impacts include more thorough information access, reduced bias toward dominant viewpoints, and improved support for users seeking comprehensive answers. As with any retrieval technology, the quality of results ultimately depends on the underlying data sources and the design of downstream systems. Potential risks mirror those of existing retrieval systems, including surfacing misinformation or amplifying biases present in the corpus.

Table 22: Contribution analysis of first and second stage retrieval on QAMPARI test set using Qwen3-30B as the verifier. We report the average number of new gold documents retrieved and the average number of new answers covered per question in each stage, with verifier budget B=100. We also report the number of repeated answers in Turn 2, as well as the percentage of new answers found.

		# New Gold Docs.		**# New Uniq. Ans.**		Turn 2 Repeated Answer	Turn 2 New Answer %
		Turn 1	Turn 2	Turn 1	Turn 2		
Contriever	FT (D_i) + FT (D_i)	49.59	17.60	7.12	0.44	6.79	6.1%
	FT (D_i) + FT (D_r)	49.59	**22.83**	7.12	**0.86**	6.92	11.1%
Qwen3	FT (D_i) + FT (D_i)	50.94	22.35	7.17	0.73	7.05	9.3%
	FT (D_i) + FT (D_r)	50.94	**23.07**	7.17	**0.82**	7.04	10.4%
INF	FT (D_i) + FT (D_i)	55.32	22.73	7.50	0.55	7.33	6.9%
	FT (D_i) + FT (D_r)	55.32	**26.12**	7.50	**1.11**	7.31	13.2%

Table 23: Contribution analysis of first-stage and second-stage retrieval on QAMPARI test set. We report the average number of gold documents retrieved and the average number of new answers covered per question in each stage with verifier budget of 50.

		# New Gold Docs.		**# New Uniq. Ans.**	
		Turn 1	Turn 2	Turn 1	Turn 2
Contr-iever	FT (D_i) + FT (D_i)	27.93	10.69	5.68	0.51
	FT (D_i) + FT (D_r)	27.93	**13.98**	5.68	**1.05**
Qwen3	FT (D_i) + FT (D_i)	28.95	13.53	5.84	0.89
	FT (D_i) + FT (D_r)	28.95	**13.61**	5.84	**0.93**
INF	FT (D_i) + FT (D_i)	30.98	13.50	6.13	0.66
	FT (D_i) + FT (D_r)	30.98	**15.09**	6.13	**1.04**

LLM Verifier Prompt

Prompt: You are a document relevance verifier. Your task is to determine if a given document contains a direct answer to a specific question.

Question: *Harmony Korine was both screenwriter and director of what movie?*

Document Title: *Harmony Korine*

Document Text: *That Mr. Korine appears to be having it both (or many) ways may seem like a cop-out, but only if you believe that the role of the artist is to be a didact or a scold\", wrote \"The New York Times\". Principal filming wrapped up on March 30, 2012. The film was selected to compete for the Golden Lion at the 69th Venice International Film Festival. \"Spring Breakers\" received its world premiere at the 2012 Venice International Film Festival, and later was shown at the Toronto International Film Festival before being released to the general public in March 2013.*

Instructions:
- Answer with ONLY "YES" if the document directly answers the question or contains clear, specific information that answers the question.
- Answer with ONLY "NO" if the document does not contain a direct answer, is only tangentially related, or does not provide the specific information asked for
- Be strict in your evaluation. The document should contain factual information that directly addresses the query

Your answer (YES or NO):

Figure 8: Prompt used for the Qwen3-30B verifier.

SearchR1 Prompt

Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, if you find you lack some knowledge, you can call a search engine by <search> query </search> and it will return the top searched results between <information> and </information>. You can search as many times as you want. If you find no further external knowledge needed, you can directly provide the answer inside <answer> and </answer>, without detailed illustrations. For example, <answer> Beijing </answer>. Question: {question}

Figure 9: Prompt used for the SearchR1 baseline.

Tongyi Prompt

You are an expert in researching and answering complex questions. Your core function is to conduct thorough, multi-source investigations into any topic. You must handle both broad, open-domain inquiries and queries within specialized academic fields. For every request, synthesize information from credible, diverse sources to deliver a comprehensive, accurate, and objective response. Each of the questions has multiple answers. When you have gathered sufficient information and are ready to provide the definitive response, you must enclose the entire final answer within <answer></answer> tags. The answer should be a list of answers, separated by semi-colons.
Tools
You may call the search tool one or multiple times to assist with the user query.
You are provided with the search tool, its signature within <tools></tools> XML tags:
<tools>
{"type": "function", "function": {"name": "search", "description": "Perform Google web searches then returns a string of the top search results. Accepts a single query.", "parameters": {"type": "object", "properties": {"query": {"type": "string", "description": "The search query."}}, "required": ["query"]}}}
</tools>
For each search tool call, return a json object with function name and arguments within <tool_call></tool_call> XML tags:
<tool_call>
{"name": search, "arguments": <args-json-object>}
</tool_call>

Figure 10: Prompt used for the Tongyi baseline.

Instruction Aware Retrieval Prompt

Retrieve documents that answer the query, but do NOT retrieve documents about {answer_1}, {answer_2}, ..., {answer_n}, as these answers are already covered. \n\n Query: {query}

Figure 11: Prompt used for instruction-aware retrieval.

Reranking Prompt

You are a relevance assessment assistant. \n Determine whether the following document is relevant to the question.\n\n Respond with ONLY 'Yes' or 'No'. No explanation.\n\n f"Question: {query}\n\n f"Document Title: {title}\n f"Document Text: {text}\n\n Relevant:

Figure 12: Prompt used for LLM reranking with Qwen3-30B.

Query Expansion Prompt

Your task is to expand the original query with relevant terms and concepts from the documents. The new query will be used by a retrieval model to retrieve a new set of documents that captures new perspectives. \n Original Query: {query} \n Documents: {documents_text} \n Instructions: 1. Create a new question using relevant terms and concepts from the documents \n 2. DO NOT answer the question \n 3. Your response should be one sentence \n 4. Your response should end with a question mark \n New Query:

Figure 13: Prompt used for the query expansion baseline.

SearchR1 Example Trajectory

<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, if you find you lack some knowledge, you can call a search engine by <search> query </search> and it will return the top searched results between <information> and </information>. You can search as many times as your want. If you find no further external knowledge needed, you can directly provide the answer inside <answer> and </answer>, without detailed illustrations. For example, <answer> Beijing </answer>. Question: What manga was drawn by Ryoichi Ikegami?
<|im_end|>
<|im_start|>assistant
<think>I need to find the manga drawn by Ryoichi Ikegami. I'll search for it.</think>
<search>Ryoichi Ikegami drawn manga</search>
<information>Doc 1(Title: "Ryoichi Ikegami") Ryoichi Ikegami is a Japanese manga artist. After graduating from junior high school he moved to Osaka and drew manga while working as a billboard sign painter debuting at the age of 17 writing rental comics. In 1966 he published a story called in the gekiga magazine Garo that caught the eye of fellow Garo contributor, manga artist Shigeru Mizuki, who offered him a job as his assistant. Ikegami accepted and moved to Tokyo and worked as Mizuki's assistant for two and a half years. In 2001, he won the Shogakukan Manga Award for general manga as the artist of "Heat".
Doc 2(Title: "Mikyo") Mikyo Mikyo
Doc 3(Title: "Manga iconography") Manga iconography Japanese manga has developed its own visual language or iconography for expressing emotion and other internal character states. This drawing style has also migrated into anime, as many manga stories are adapted into television shows and films. While this article addresses styles from both types of output, the emphasis here is on the manga origins for these styles. The popular and recognizable style of manga is very distinctive. Emphasis is often placed on line over form, and the storytelling and panel placement differ from those in American comics.
</information>
<think>I found out that Ryoichi Ikegami is a Japanese manga artist. Now I need to find which manga he drew.</think>
<search>Manga drawn by Ryoichi Ikegami</search>
<information>Doc 1(Title: "Manga") Manga Manga (Japanese: 漫画) are comics or graphic novel s originating from Japan. Most manga conform to a style developed in Japan in the late 19th century, and the form has a long prehistory in earlier Japanese art. The term "manga" is used in Japan to refer to both comics and cartooning. Outside of Japan, the word is typically used to refer to comics originally published in the country. In Japan, people of all ages read manga.
Doc 2(Title: "Ryoichi Ikegami") Ryoichi Ikegami is a Japanese manga artist. After graduating from junior high school he moved to Osaka and drew manga while working as a billboard sign painter debuting at the age of 17 writing rental comics. In 1966 he published a story called in the gekiga magazine Garo that caught the eye of fellow Garo contributor, manga artist Shigeru Mizuki, who offered him a job as his assistant. Ikegami accepted and moved to Tokyo and worked as Mizuki's assistant for two and a half years. In 2001, he won the Shogakukan Manga Award for general manga as the artist of "Heat".
Doc 3(Title: "Manga iconography") Manga iconography Japanese manga has developed its own visual language or iconography for expressing emotion and other internal character states. This drawing style has also migrated into anime, as many manga stories are adapted into television shows and films. While this article addresses styles from both types of output, the emphasis here is on the manga origins for these styles. The popular and recognizable style of manga is very distinctive. Emphasis is often placed on line over form, and the storytelling and panel placement differ from those in American comics.
</information>
<think>I found out that Ryoichi Ikegami is a Japanese manga artist. Now I need to find which manga he drew. I'll search for his notable works.</think>
<search>Ryoichi Ikegami notable works</search>
<information>Doc 1(Title: "Mikyo") Mikyo Mikyo
Doc 2(Title: "Ryoichi Ikegami") Ryoichi Ikegami is a Japanese manga artist. After graduating from junior high school he moved to Osaka and drew manga while working as a billboard sign painter debuting at the age of 17 writing rental comics. In 1966 he published a story called in the gekiga magazine Garo that caught the eye of fellow Garo contributor, manga artist Shigeru Mizuki, who offered him a job as his assistant. Ikegami accepted and moved to Tokyo and worked as Mizuki's assistant for two and a half years. In 2001, he won the Shogakukan Manga Award for general manga as the artist of "Heat".
Doc 3(Title: "Baiko") Baiko baiko
</information>
<think>I found out that Ryoichi Ikegami drew the manga "Heat". Now I can provide the answer.</think>
<answer> Heat </answer>

Figure 14: Example inference trajectory for the SearchR1 baseline.

Tongyi Example Trajectory


```
{
  "type": "reasoning",
  "tool_name": null,
  "arguments": null,
  "output": "The API expects a single query string, not an array. So we need to provide a single query; perhaps we can combine multiple queries separated by newline or something? However specification says: \"Perform a search for the top 10 results for each query in one call.\" That suggests it can handle array of query strings. But error says 'query' must be a string. Perhaps the wrapper expects a string named 'query', but earlier spec says Array of query strings. Let's try sending a single query that includes multiple terms separated by semicolons.\n\nWe'll do one query: \"Ryoichi Ikegami manga list\". Let's try."
 },

 {
  "type": "tool_call",
  "tool_name": "search",
  "arguments": "{\"query\": \"Ryoichi Ikegami manga list\"}",
  "output": "A search for 'Ryoichi Ikegami manga list' found 5 results:\n\n## Web Results\n1. <docid: 1494481__1> [Ryoichi Ikegami He became a professor at Osaka Uni...]\nRyoichi Ikegami He became a professor at Osaka University of Arts in 2005. Ikegami has worked on several popular series, such as \"Mai, the Psychic Girl\" with writer Kazuya Kudo, \"Crying Freeman\", with writer Kazuo Koike, as well as \"Sanctuary\" and \"Heat\" with writer Sho Fumimura. He also wrote and drew \"\", a manga version of Spider-Man and collaborated with Garon Tsuchiya for the manga \"BOX\" (BOX 暗い箱). His most recent work is \"Begin\" currently serialized in Big Comic Superior.\n\n2. <docid: 1494481__0> [Ryoichi Ikegami is a Japanese manga artist. After ...]\nRyoichi Ikegami is a Japanese manga artist. After graduating from junior high school he moved to Osaka and drew manga while working as a billboard sign painter debuting at the age of 17 writing rental comics. In 1966 he published a story called in the gekiga magazine Garo that caught the eye of fellow Garo contributor, manga artist Shigeru Mizuki, who offered him a job as his assistant. Ikegami accepted and moved to Tokyo and worked as Mizuki's assistant for two and a half years. In 2001, he won the Shogakukan Manga Award for general manga as the artist of \"Heat\".\n\n3. <docid: 5676940__0> [Ryoko Yamagishi is a Japanese manga artist. She is...]\nRyoko Yamagishi is a Japanese manga artist. She is one of the Year 24 Group, a collection of female artists who innovated (girls') manga throughout the 1970s. Her major works include and \"Terpsichora\", both of which have won manga awards. Biography. Ryoko Yamagishi was born on September 24, 1947, in Kamisunagawa, Hokkaido, Japan. As a child, she studied ballet, which plays a part in many of her works. When she read the manga of Machiko Satonaka in 1964, she decided to pursue becoming a manga artist.\n\n4. <docid: 66090580__0> [Ryū to Ichigo is a Japanese shogi -themed manga se...]\nRyū to Ichigo is a Japanese shogi -themed manga series written and illustrated by Mitsuharu Yanamoto. It has been serialized in Shogakukan's \"Weekly Shōnen Sunday\" since May 2020. Publication. \"Ryū to Ichigo\" is written and illustrated by Mitsuharu Yanamoto. The series began in Shogakukan's \"Weekly Shōnen Sunday\" on May 20, 2020. Shogakukan has collected is chapters into individual \"tankōbon\" volumes. The first volume was released on August 18, 2020, with a promotional video posted on the same day. As of June 17, 2021, four volumes have been released.\n\n5. <docid: 65637267__0> [Ryoko (manga) Ryoko (stylized as RYOKO) is a Japan...]\nRyoko (manga) Ryoko (stylized as RYOKO) is a Japanese manga series written and illustrated by Kaito Mitsuhashi. It was serialized in Shogakukan's \"Weekly Shōnen Sunday\" from October 2016 to June 2019, with its chapters collected in five \"tankōbon\" volumes. Publication. \"Ryoko\" is written and illustrated by Kaito Mitsuhashi. Mitsuhashi, at 21, won the Shin Shedai Sunday Award (New Generation Sunday Award) for his work \"Ryoko Meet Meal!!\" in December 2015."
 },
 {
  "type": "reasoning",
  "tool_name": null,
  "arguments": null,
  "output": "The search result #1 is likely a summary of his works: \"Mai, the Psychic Girl\", \"Crying Freeman\", \"Sanctuary\", \"Heat\", manga version of Spider-Man, \"BOX\", \"Begin\". Also maybe \"Kaidan Restaurant\"? Not sure.\n\nBetter to open result #1."
 },
```


Figure 15: Example inference trajectory for the Tongyi baseline.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction section both outline the main contributions we make, including the retrieve-verify-retrieve (RVR) framework, the experimental results and the advantage of the proposed method over existing baselines.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We describe the proposed framework in Section 3.1, the specific models we use in Section 3.2 under "Implementation Details", and the training hyperparameters in Section 3.2 under "Implementation Details". We describe the datasets in Section 4.1, and baselines in Section 4.2. We include more details on the dataset in Appendix B and the exact prompts used in Appendix D. We will also release the model checkpoints upon publication.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We plan to release the code and models in the near future.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe the datasets in Section 4.1, and provide more details on the dataset in Appendix B. The data splits are the default splits as in the original papers, as described in the two sections mentioned above. The training details (hyperparameters, optimizers) are in Section 3.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide the statistical significance of the experiments directly in the main results tables, including Table 1 and Table 3. Statistical significance is tested using bootstrap resampling with 10,000 trials at $\alpha = 0.05$, as mentioned in Table 1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 3.2 "Implementation Details" provides the machines used and average training wall-clock times per model family on 4 H200 GPUs. Table 2 details inference time per experiment and memory requirements.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Detailed in Appendix E.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Detailed in Appendix E.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release a strong pre-trained language model, image generator, or a scraped dataset.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the original papers of the dataset we use are cited, both in Section 4.1 and Appendix B. QUEST [21] has apache-2.0 and QAMPARI [1] has CC0-1.0 license. We were unable to find a license for WebQuestionsSP [34].

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: We plan to release the model and code in the near future.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We use Qwen3-30B as a verifier, a core and non-standard component of the RVR framework. Its usage is described in Section 3.2 and its impact is analyzed in Section 6 and Appendix C3.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.