# Technique Appendix for Submission 12525

**Authors of Submission 12525**

## Contents



Preprint. Under review.

# 1 Practical Considerations and Societal Impact

## 1.1 Limitations

Despite our efforts to improve the LLM's ability to reason and search, our work has several limitations. Some of these limitations are shared across many existing RAG systems. We highlight them not only as directions for future work but also in the hope of inspiring broader exploration within the community to advance the development of more effective retrieval-augmented reasoning systems.

**First**, our method currently focuses on textual inputs and outputs. Extending reasoning-augmented search to multimodal scenarios, e.g., incorporating images or structured tables, remains an important direction for future work. We plan to integrate EXSEARCH with large vision-language models and extend the text-only retrieval documents to a multimodal corpus.

**Second**, in line with prior work, we deliberately avoid hardcoding heuristics such as fixed query decomposition rules or predefined in-context learning demonstrations. However, similar to most existing approaches, our model performs retrieval at every reasoning step, regardless of necessity. Since LLMs pre-trained on large web data have parameterized extensive world knowledge, for some simpler queries, this fixed retrieval strategy may be redundant. An important next step is to develop more adaptive strategies that allow the model to decide *when* to retrieve based on context, rather than always retrieving at each step.

**Third**, our training and evaluation are based on rule-based, answer-centric metrics such as Exact Match and F1 score. While effective for benchmark evaluation, these metrics may not fully capture performance in more open-ended or exploratory tasks, especially for long-form generation. As mentioned in our paper, in future work, we aim to explore more open-domain setups and alternative supervision signals beyond gold-standard answers, such as LLM-as-the-judge.

Overall, the proposed EXSEARCH makes progress in the RAG research area by effectively teaching LLMs to interleave reasoning and search within a self-improving process. However, it remains an initial step. We believe addressing the above limitations will be crucial for building more general and robust search-augmented reasoning systems.

## 1.2 Ethics Statement

The research conducted in this paper centers around the development of a reasoning-augmented search framework. The proposed method enables Language Models (LLMs) to dynamically retrieve and reason over external information. In the process of conducting this research, we have adhered to ethical standards to ensure the integrity and validity of our work. All the questions used in this study were obtained from existing benchmarks, ensuring a high level of transparency and reproducibility in our experimental procedure. To support our retrieval system, we used an open-source corpus, specifically Wikipedia. This ensures that our research utilizes publicly accessible and freely available data, minimizing potential bias and promoting fairness.

We have made every effort to ensure that our study does not involve human subjects, private data, or any content that may cause harm to individuals or social groups. No part of this work includes deceptive practices or intentional misuse of information. We are committed to conducting and presenting this research with integrity and social responsibility. We intend to release our code and implementation details to support open research, following the NeurIPS submission policy, and aim to facilitate further study in the information retrieval and retrieval-augmented generation areas.

### 1.3 Societal Impacts

Our work introduces EXSEARCH, a reasoning-augmented search framework that interleaves multi-step reasoning with dynamic document retrieval. The primary goal is to improve the factual accuracy and interpretability of LLMs in open-domain question answering and knowledge-intensive tasks. The model presented in this work requires explicit grounding in external sources, and our framework emphasizes verifiable retrieval and intermediate reasoning steps, which improve transparency. These properties support downstream mitigation efforts, such as traceable generation or retrieval auditing. Additionally, we note that our current implementation does not involve user data or personalized profiles, thus mitigating privacy concerns.

This approach offers several key benefits. *Improved Transparency*: By grounding answers in external sources and providing intermediate reasoning steps, EXSEARCH allows for better traceability of the model's decision-making process, which is crucial for understanding and verifying outputs; *Enhanced Trustworthiness*: The framework's reliance on verifiable external retrieval reduces the risk of generating hallucinated information, contributing to more reliable and factually accurate responses; and *Broader Applicability*: With its focus on factual grounding and reasoning, EXSEARCH can be applied to a wide range of knowledge-intensive applications, including scientific research, legal analysis, and education, where accuracy and clarity are essential.

While this research is foundational and not tied to specific applications or deployments, we acknowledge the broader risks associated with enhancing the factual accuracy and coherence of LLMs. Specifically, the improved text generation capabilities of LLMs may be misused in malicious contexts or scenarios, such as: (i) generating more convincing disinformation; (ii) fabricating plausible but incorrect content by selectively retrieving or combining real documents; and (iii) enabling better-targeted persuasive text (e.g., phishing, political propaganda). In summary, while EXSEARCH could be misused in unintended settings, its design principles, combined with the potential for verifiable and auditable generation, offer avenues for responsible deployment. While we do not foresee immediate or direct societal harm, we encourage future work to explore safeguards, such as automated monitoring and ethical auditing mechanisms, in high-stakes applications.

## 2 Comparison with Prior Work

In this work, the proposed method enables LLMs to perform interleaved reasoning and search (also referred to as agentic search in this work) and optimizes their capability for this pattern through a self-incentivized framework. Below, we compare our method with previous iterative retrieval methods and self-learning methods in detail.

### 2.1 Comparing with Previous Iterative Retrieval Method.

In this paragraph, we systematically compare EXSEARCH with previous work that incorporates iterative retrieval techniques, especially those integrated with LLMs, highlighting our fundamental innovations. Most previous iterative retrieval approaches rely on a modular multi-stage pipeline, where distinct models are trained separately for sub-tasks such as query rewriting, retrieval, and evidence aggregation. For example, For example, Self-RAG [1] and RankRAG [2] train LLMs for document relevance judgment; ADELIE [3] adopts LLMs for knowledge extraction; while other work [4, 5] cascades multiple specialized components in sequence. Despite their progress, this modular design overlooks the end-to-end optimization, leading to potential misalignment among the training objectives in different stages. More recently, several works attempt to prompt powerful, extremely large models (i.e., GPT-3.5 or Qwen-32B) to iteratively interact with retrieval engines through in-context learning [6, 7]. However, they are limited by using predefined demonstrations or in-context learning examples. While simplifying system design, these approaches overlook improving the LLM's intrinsic reasoning capability for adaptive retrieval and generation. In contrast, EXSEARCH trains a *unified* LLM to reason over the evolving context, adaptively decide what information to retrieve, optionally re-rank retrieved documents, extract fine-grained supporting evidence, and generate the final answer, all within an end-to-end learning framework. Unlike previous work, EXSEARCH aligns the training signals of these actions through a coherent learning objective, improving the LLM in an end-to-end manner.

Table 1: Main notation used in this work.

| Symbol | Description |
|---|---|
| $x$ | The initial input query. |
| $y$ | The ground-truth answer corresponding to the initial query $x$. |
| $\theta$ | The parameters of the large language model (LLM). |
| $\mathcal{R}$ | The external retriever. |
| $i$ | The index of the $i$-th step in the reasoning and retrieval process. |
| $x_i$ | The sub-query generated at the $i$-th step. |
| $\boldsymbol{d}_i$ | The set of documents retrieved in the $i$-th step, i.e., $\boldsymbol{d}_i = \mathcal{R}(x_i) = \{\boldsymbol{d}_{i,j} \mid j \in [K]\}$. |
| $e_i$ | The fine-grained evidence extracted from the retrieved documents $\boldsymbol{d}_i$. |
| $t$ | The index of the $t$-th training iteration. |
| $\boldsymbol{z}$ | The full reasoning trajectory, consisting of interleaved sub-queries, retrieved documents, and extracted evidence, i.e., $\boldsymbol{z} = \{(x_i, \boldsymbol{d}_i, e_i) \mid i \in [\lvert\boldsymbol{z}\rvert]\}$ |

*The most contemporary work to* EXSEARCH *is Search-R1 [8], developed independently around the same time*. Search-R1 applies the proximal policy optimization (PPO [9]) algorithm to encourage LLMs to issue multiple search queries during reasoning. However, EXSEARCH differs from Search-R1 in three critical aspects: (i) *Reasoning Process*: EXSEARCH enables richer retrieval actions, including query generation, optional re-ranking of retrieved documents, and fine-grained evidence extraction, while Search-R1 only conducts iterative query decomposition without reflection or re-ranking; (ii) *Training Algorithm*: EXSEARCH treats search trajectories as latent variables and optimizes a variational evidence lower bound via a Generalized Expectation-Maximization algorithm [10], achieving stable and progressive self-improvement. In contrast, Search-R1 adopts an online on-policy reinforcement learning framework, which often suffers from sample inefficiency and training instability [11–13]; and (iii) *Reward Signal*: EXSEARCH introduces a trajectory-level training signal, evaluating the quality of the entire search trajectory based on the likelihood of generating the correct answer, rather than relying solely on a binary outcome-based reward as in Search-R1.

### 2.2 Comparison with Previous Self-training Methods.

Recent studies have explored self-training frameworks where a model is iteratively trained on its generated data [14, 15]. For example, some work [16, 14] allows the model to first generate a solution and fine-tune it on the generated trajectories, showing promising results on mathematical reasoning tasks. Similar ideas have also been applied in combination with REINFORCE Leave-One-Out (RLOO) methods [14, 17, 18]. More recently, other work [15, 19] proposes self-rewarding methods, where the LLM itself is used via LLM-as-a-Judge prompting to provide its own rewards during an iterative DPO [20] training process. While effective in closed-book settings, these approaches overlook a critical limitation: **the inherent limitation of parametric knowledge in LLMs** [21, 22]. Without external retrieval, the model cannot dynamically access relevant information when it is missing from its internal memory. In contrast, this work focuses on agentic search, which aims to enable the LLM to interleave dynamic retrieval within the reasoning process and further reflect on the retrieved content at a fine-grained level (See the § 2.1 in the main body of the paper). Additionally, we provide a theoretically grounded analysis of the advantages of our method, as discussed in this Appendix 3.2 and Appendix 3.3. We also introduce an extended resource, EXSEARCH-Zoo, which supports multiple model families and richer reasoning actions.

## 3 Detailed Theoretic Analysis

Due to space constraints in the main body of this paper, we include the detailed version of the theoretical derivations and analyses of EXSEARCH here. Below, we first formulate the search and reasoning process in EXSEARCH. We then introduce how the generalized expectation–maximization technique is leveraged to improve the LLM's capability in EXSEARCH through a self-improving loop (§ 3.1). Additionally, we prove that the resulting training procedure is convergent. Table 1 lists the main notation used in the paper.

### 3.1 Skeleton Derivations for EXSEARCH

**Reviewing Agentic Search Procedure** In this work, the proposed EXSEARCH is inspired by the *Exploratory Search* paradigm [23], which models information-seeking as a dynamically unfolding process where search queries are iteratively refined based on intermediate results. EXSEARCH simulates this by interleaving three core actions:

(i) ***thinking***: The LLM generates a query $x_i$ based on the current context $x$ and the accumulated search trajectory $\boldsymbol{z}_{<i}$, formulated as:

$$x_i = p(x_i \mid x, \boldsymbol{z}_{<i}; \theta) \tag{1}$$

(ii) ***search***: A retrieval module $\mathcal{R}$ retrieves the top-$K$ documents $\boldsymbol{d}_i$ relevant to the query $x_i$:

$$\boldsymbol{d}_i = \mathcal{R}(x_i) \tag{2}$$

(iii) ***recording***: The LLM reflects on the retrieved documents $\boldsymbol{d}_i$ and extracts evidence $e_i$ conditioned on $x$ and $\boldsymbol{d}_i$:

$$e_i = p(e_i \mid x, \boldsymbol{d}_i; \theta) \tag{3}$$

In this step, the model focuses solely on the current sub-query and its associated documents, reducing computational cost by limiting the context to the most relevant information.

Formally, the reasoning-augmented search process is modeled as a sequence $\boldsymbol{z} = \{(x_i, \boldsymbol{d}_i, e_i) \mid i \in [|\boldsymbol{z}|]\}$, with the joint likelihood:

$$p(\boldsymbol{z} \mid x; \theta) = \prod_{i=1}^{|\boldsymbol{z}|} p\left((x_i, \boldsymbol{d}_i, e_i) \mid x, \boldsymbol{z}_{<i}; \theta\right) \tag{4}$$

After the interleaved search and reasoning process, the LLM aggregates information from $\boldsymbol{z}$ to generate the final answer $y \sim p(y \mid x, \boldsymbol{z}; \theta)$.

**Training Objective.** The goal of EXSEARCH is to improve the LLM's ability to generate the correct answer $y$ after reasoning. The training objective is formulated as:

$$\log p(y \mid x; \theta) = \log \sum\nolimits_{\boldsymbol{z}} p(y, \boldsymbol{z} \mid x; \theta) \tag{5}$$

Here, $\boldsymbol{z} = \{(x_i, \boldsymbol{d}_i, e_i) \mid i \in [|\boldsymbol{z}|]\}$ represents a sequence of the *thinking*, *search*, and *recording* actions. In EXSEARCH, we introduce a proposal distribution $q(\boldsymbol{z} \mid x)$ to approximate the sampling space of $\boldsymbol{z}$ and apply **Jensen's inequality** to the marginal log-likelihood in Eq. 5:

$$\begin{aligned} &\log \sum\nolimits_{\boldsymbol{z}} q(\boldsymbol{z} \mid x) \frac{p(y, \boldsymbol{z} \mid x; \theta)}{q(\boldsymbol{z} \mid x)} \\ \geq &\sum\nolimits_{\boldsymbol{z}} q(\boldsymbol{z} \mid x) \log \frac{p(y, \boldsymbol{z} \mid x; \theta)}{q(\boldsymbol{z} \mid x)} \\ = &\mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z}|x)} \left[\log \frac{p(y, \boldsymbol{z} \mid x; \theta)}{q(\boldsymbol{z} \mid x)}\right] \end{aligned} \tag{6}$$

The right-hand side represents the variational evidence lower bound (ELBO) of $\log p(y \mid x; \theta)$, and the bound becomes tight if $q(\boldsymbol{z} \mid x)$ approximates the true posterior distribution $p(\boldsymbol{z} \mid y, x; \theta)$. Thus, we can iteratively estimate $p(\boldsymbol{z} \mid y, x; \theta)$ and maximize this ELBO using the generalized expectation-maximization algorithm to progressively improve the LLM.

**E-step: Trajectory Exploration.** In the E-step, we estimate the distribution over reasoning trajectories $\boldsymbol{z}$ by sampling from the LLM $\theta$. In the $t$-th iteration, we approximate the distribution $q(\boldsymbol{z} \mid x) \approx p(\boldsymbol{z} \mid x, y; \theta)$, yielding the following form for the ELBO:

$$\text{ELBO} = \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z}|x,y;\theta^t)} \left[\log p(y, \boldsymbol{z} \mid x; \theta)\right] + \mathcal{H}(p(\boldsymbol{z} \mid x, y; \theta^t)) \tag{7}$$

Here, the entropy term $\mathcal{H}(p(\boldsymbol{z} \mid x, y; \theta^t))$ is constant with respect to $\theta$. Since direct sampling from the posterior is intractable, we apply importance sampling [24, 25], where the distribution $p(\boldsymbol{z} \mid x; \theta^t)$ is easier to sample from, and each sample is assigned an importance weight:

$$w(\boldsymbol{z}) = \frac{p(\boldsymbol{z} \mid x, y; \theta^t)}{p(\boldsymbol{z} \mid x; \theta^t)} \tag{8}$$

This allows us to rewrite the ELBO as:

$$\text{ELBO} = \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|x;\theta^t)}\left[w(\boldsymbol{z})\log p(y,\boldsymbol{z}\mid x;\theta)\right] + c, \tag{9}$$

where $c$ is a constant.

**M-step: Re-weighted Trajectory Learning.** In the M-step, we update the model parameters $\theta$ by maximizing the ELBO from Eq. 9. The objective becomes:

$$\theta = \arg\max_{\theta} \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|x;\theta^t)}\left[w(\boldsymbol{z})\log p(y,\boldsymbol{z}\mid x;\theta)\right] \tag{10}$$

The overall training process is performed using stochastic gradient descent, with gradients computed as:

$$\nabla_\theta \text{ELBO}(\theta) = -\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|x;\theta^i)}\left[w(\boldsymbol{z})\nabla_\theta(\mathcal{L}_\mathcal{R} + \mathcal{L}_\mathcal{A})\right] \tag{11}$$

### 3.2 Convergence Analysis

Below, we analyze the convergence behavior of ExSearch using the generalized expectation-maximization algorithm, providing a more detailed explanation than in the main body of the paper. We show that the training objective $\log p(y \mid x;\theta)$ is non-decreasing after each training iteration and progressively converges to a stationary point due to its upper-bounded property. To provide a tighter characterization of convergence, we interpret the optimization gap as a KL divergence between importance-weighted sampling and the true posterior.

**Lemma 3.1 (Monotonic Improvement).** *At each iteration $t \in \mathbb{Z}^+$, the training objective of the LLM satisfies:*

$$\log p(y \mid x;\theta^{t+1}) \geq \log p(y \mid x;\theta^t). \tag{12}$$

*Proof.* Reviewing the EM-style training in our method, the main concept involves introducing a tractable evidence lower bound (ELBO) and progressively improving it to optimize $\log p(y|x;\theta)$. In the E-step of the $t$-th iteration, we sample trajectories $\boldsymbol{z}$ from the current model as $\boldsymbol{z} \sim p(\boldsymbol{z} \mid x;\theta^t)$ and assign each a weight $w(\boldsymbol{z}) = \frac{p(\boldsymbol{z}|x,y;\theta^t)}{p(\boldsymbol{z}|x;\theta^t)} \propto p(y \mid x,\boldsymbol{z};\theta^t)$. We define the ELBO as:

$$\text{ELBO}(\theta,\theta^t) := \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|x;\theta^t)}\left[w(\boldsymbol{z})\log p(y,\boldsymbol{z}\mid x;\theta)\right] + c. \tag{13}$$

In the M-step, we update the model by maximizing this ELBO:

$$\theta^{t+1} = \arg\max_{\theta} \text{ELBO}(\theta,\theta^t), \tag{14}$$

which guarantees that: $\text{ELBO}(\theta^{t+1},\theta^t) \geq \text{ELBO}(\theta^t,\theta^t)$. Meanwhile, since the ELBO indicates the evidence lower bound for the marginal distribution $\log p(y \mid x;\theta^{t+1})$, it holds that: $\log p(y \mid x;\theta^{t+1}) \geq \text{ELBO}(\theta^{t+1},\theta^t)$, and $\text{ELBO}(\theta^t,\theta^t) = \log p(y \mid x;\theta^t)$. By combining these two equations, we have:

$$\log p(y \mid x;\theta^{t+1}) \geq \log p(y \mid x;\theta^t). \tag{15}$$

□

Therefore, we have completed the proof of non-decreasing improvement for each training iteration.

**Lemma 3.2 (Boundedness).** *The sequence $\{\log p(y \mid x;\theta^t)\}_{t=1}^{\infty}$ is upper-bounded.*

*Proof.* Since $p(y \mid x;\theta) \in [0,1]$, we naturally have $\log p(y \mid x;\theta) \leq 0$ for all $\theta$. □

**Theorem 3.3 (Convergence of ExSearch).** *By Lemma 3.1 and Lemma 3.2, the sequence $\{\log p(y \mid x;\theta^t)\}$ is non-decreasing and upper-bounded. By the Monotone Convergence Theorem [26], it converges to a finite limit.*

**Remark 3.1 (Tightness via KL Divergence).** *The ELBO can be interpreted as a tight bound of $\log p(y \mid x;\theta)$ with the following identity:*

$$\log p(y \mid x;\theta) = \textit{ELBO}(\theta,\theta^t) + \textit{KL}(q^*(\boldsymbol{z}) \parallel p(\boldsymbol{z} \mid x,y;\theta)), \tag{16}$$

*where $q^*(\boldsymbol{z}) \propto w(\boldsymbol{z}) \cdot p(\boldsymbol{z} \mid x;\theta^t)$ is the induced sampling distribution. Therefore, under mild regularity conditions (e.g., bounded support, continuity of log-likelihood), as $q^*$ approaches the true posterior, the KL term vanishes, and the ELBO becomes tight. This strengthens the convergence result and characterizes the optimization gap.*

### 3.3 Rethink What Does the LLM Learn Within EXSEARCH?

In § 3.1 and main body of our paper, we demonstrate that the answer log-likelihood (i.e., $p(y|x, \boldsymbol{z}; \theta)$) serves as an end-to-end training signal in our self-incentivized training process, guiding the model to reason and search. Given the primary goal of information retrieval, especially in downstream tasks like retrieval-augmented generation, we typically train the model to generate accurate answers for users and evaluate its performance using correctness-oriented metrics, such as exact match score or accuracy, thereby ensuring the factuality. However, a natural question arises:

> *Does our self-improving framework also maximize the commonly used downstream metrics, such as accuracy, and why does it work or not?*

In addition to the strong empirical results shown in our experiments, we provide a more interpretable and theoretical analysis in this section. Below, we first briefly highlight the learning objective introduced in § 3.1, denoted as the ***vanilla objective***, which uses $w(\boldsymbol{z}) \propto p(y \mid x, \boldsymbol{z}; \theta)$ as the training signal. We then introduce a ***goal-oriented learning objective***, where the LLM is trained to maximize an evaluation metric using a similar Expectation-Maximization algorithm. Finally, we relate these two objectives and show their consistency in model training and optimization, illustrating why the vanilla objective aligns with maximizing the expected metric and downstream task performance.

#### 3.3.1 Reviewing Vanilla Learning Objective.

The vanilla learning objective is defined as ELBO $= \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z}|x;\theta^t)} [w(\boldsymbol{z}) \log p(y, \boldsymbol{z} \mid x; \theta)]$. Here, $w(\boldsymbol{z})$ is the weighting function given by $w(\boldsymbol{z}) = \frac{p(\boldsymbol{z}|x,y;\theta^t)}{p(\boldsymbol{z}|x;\theta^t)}$. This weighting function is derived from the ratio of the posterior distribution $p(\boldsymbol{z} \mid x, y; \theta^t)$ to the prior $p(\boldsymbol{z} \mid x; \theta^t)$, which reflects how well the trajectory $\boldsymbol{z}$ supports the final answer $y$.

#### 3.3.2 Introducing Goal-oriented Objective

We now replace the weighting function $w(\boldsymbol{z})$ with a evaluation metric $r(y)$ (*also widely known as the reward function*), which evaluates the quality of the final answer $y$. Formally, given an evaluation metric $r(y)$ that rates the quality of an answer $y$, our goal is to optimize the model parameters $\theta$ to improve the expected performance. We define the expected learning objective under the model as:

$$\mathcal{J}(\theta) = \mathbb{E}_{y \sim p(y|x;\theta)}[r(y)] = \sum_y r(y) p(y \mid x) \tag{17}$$

In EXSEARCH, since the model first generates a reasoning trajectory $\boldsymbol{z}$ and then outputs an answer $y$, we can rewrite Eq. 17 as:

$$\mathcal{J}(\theta) = \mathbb{E}_{y \sim p(y|x;\theta)}[r(y)] = \sum_y r(y) \sum_{\boldsymbol{z}} p(\boldsymbol{z}, y \mid x) = \sum_{\boldsymbol{z}, y} r(y) p(y, \boldsymbol{z} \mid x). \tag{18}$$

Here, $p(y, \boldsymbol{z} \mid x)$ denotes the LLM generating a reasoning path $\boldsymbol{z}$ followed by a final answer $y$. Marginalizing over all possible $(\boldsymbol{z}, y)$ is typically intractable due to the large action space of the LLM. We now derive a variational surrogate for optimizing such a goal-oriented objective through a tractable lower bound.

**Proposition 1** (Variational Lower Bound as a Proxy for Metric Maximization)**.** *Given a non-negative evaluation metric function $r$, let $\mathcal{J}(\theta) := \mathbb{E}_{\boldsymbol{z}, y \sim p(\boldsymbol{z}, y|x;\theta)}[r(y)]$ be the expected metric. We can introduce a proposal distribution $q(\boldsymbol{z}, y)$ over the $(\boldsymbol{z}, y)$ space to construct a more tractable evidence lower bound:*

$$\textit{ELBO}(\theta, q) := \sum_{\boldsymbol{z}, y} q(\boldsymbol{z}, y \mid x) \log \frac{r(y) p(\boldsymbol{z}, y \mid x; \theta)}{q(\boldsymbol{z}, y \mid x)}, \tag{19}$$

*which is a function only related to $q$ and $\theta$. It satisfies:*

$$\left\| \arg\max_\theta \textit{ELBO}(\theta, q) - \arg\max_\theta \mathcal{J}(\theta) \right\| \leq c \cdot \left( \textit{KL}(q \parallel r \cdot p_\theta) \right)^{1/2}. \tag{20}$$

*, where $c$ is a constant. Under mild assumptions, the boundedness (=) holds when $q = q^* \approx r \cdot p_\theta$.*

We provide the detailed proof for this proposition in § 3.3.5. This proposition indicates that we can optimize the ELBO as a proxy to improve the $\mathcal{J}(\theta)$, following a similar Expectation-Maximization algorithm as introduced in vanilla EXSEARCH (§ 3.1). In more details, this involves alternating between the following steps: (i) E-step: sampling $(\boldsymbol{z}, y)$ trajectories; and (ii) M-step: updating the model parameters.

**E-step: Sampling Trajectories.** To approximate the true posterior distribution $q^*$, we sample $(\boldsymbol{z}, y)$ from the current model $p(\boldsymbol{z}, y \mid x; \theta^t)$ using an importance sampling strategy. The corresponding importance weight is formulated as $\frac{q^*(\boldsymbol{z},y|x)}{p(\boldsymbol{z},y|x;\theta^t)} = r(y)$, which is obtained using the evaluation metric.

$$\begin{aligned}\text{ELBO} &= \mathbb{E}_{(\boldsymbol{z},y)\sim q^*(\boldsymbol{z},y|x)}[\log p(\boldsymbol{z}, y \mid x; \theta)] + c \\ &= \mathbb{E}_{(\boldsymbol{z},y)\sim p(\boldsymbol{z},y|x;\theta^t)}[\frac{q^*(\boldsymbol{z}, y)}{p(\boldsymbol{z}, y \mid x; \theta^t)} \log p(\boldsymbol{z}, y \mid x; \theta)] + c \\ &= \mathbb{E}_{(\boldsymbol{z},y)\sim p(\boldsymbol{z},y|x;\theta^t)}[r(y) \log p(\boldsymbol{z}, y \mid x; \theta)] + c\end{aligned} \tag{21}$$

**M-step: Update the Model Parameters.** Given the weighted samples, we update $\theta$ by maximizing the weighted log-likelihood:

$$\theta = \arg\max_{\theta} \mathbb{E}_{(\boldsymbol{z},y)\sim p(\cdot|\theta^t)}[r(y) \log p(\boldsymbol{z}, y \mid x; \theta)]. \tag{22}$$

By alternating between the above E-step and M-step, we can train the LLM to maximize the given metric. This can be seen as a reward-weighted generalization of expectation-maximization [27]. This formulation naturally integrates downstream evaluation metric (via $r(y)$) into a likelihood-based training framework.

### 3.3.3 Relating Vanilla Learning Objective with Goal-oriented Objective.

In the goal-oriented objective, the evaluation metric $r(y)$ directly evaluates the quality of the answer $y$ to return a training signal. Similarly, the training signal $w(\boldsymbol{z}) \propto p(y \mid x, \boldsymbol{z}; \theta)$ represents the probability of generating a correct answer, and the higher the probability of generating the correct answer, the greater the expected metric. Therefore, we have $w(\boldsymbol{z}) \propto r(y)$.

Reviewing the optimization objectives in the *vanilla objective* and the *goal-oriented objective*: In the vanilla objective, we have $\theta = \arg\max_\theta \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|x;\theta^t)}[w(\boldsymbol{z}) \log p(y, \boldsymbol{z} \mid x; \theta)]$. In the goal-oriented objective, we have $\hat{\theta} = \arg\max_\theta \mathcal{J}(\theta) \approx \arg\max_\theta \mathbb{E}_{(\boldsymbol{z},y)\sim p(\cdot|\theta^t)}[r(y) \log p(\boldsymbol{z}, y \mid x; \theta)]$. Thus, when $w(\boldsymbol{z}) \propto r(y)$, we have $\theta \approx \hat{\theta}$. That is, under mild assumptions, the two optimization objectives are theoretically equivalent.

*In summary*, for a non-negative metric or reward $r(\cdot)$ that encourages the model to generate high-quality, correct answers, training with the vanilla objective also maximizes that the evaluation metric in downstream tasks.

### 3.3.4 Experimental Results

To validate the theoretical analysis presented above, we implement the goal-oriented objective to train the LLM. Following prior work, such as DeepSeek-R1 [28], we adopt two rule-based metrics, namely Exact Match (EM) and Accuracy (Acc.), as the function $r(y)$ in Proposition 1. These metrics are commonly used in open-domain QA and provide direct, interpretable supervision.

Figure 1 presents the performance of models trained using both the vanilla objective and the extended goal-oriented objective. We observe that models trained with different objectives (e.g., answer log-likelihood or specific metrics) converge to similar performances on the corresponding metric. For instance, the curves labeled *vanilla* and *w/ Exact Match* on the left side of Figure 1 both achieve a score of 50 in exact match after 5 iterations of training. A similar trend can also be found on the right side of Figure 1, where both the curves labeled *vanilla* and *w/ Accuracy* achieve a score of 53 in accuracy. This observation supports and aligns with the theoretical analysis presented above.

Additionally, we observe that the model trained using the vanilla answer log-likelihood converges faster and achieves the best average performance across both exact match and accuracy metrics. We analyze that there are two potential reasons. **First**, the $p(y \mid \boldsymbol{z}, x; \theta)$ is proportional to exact match and

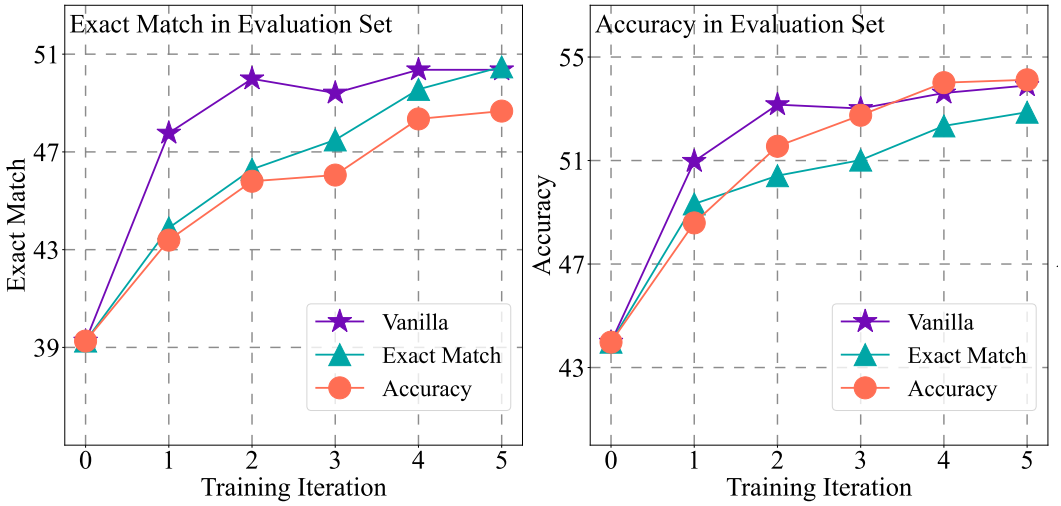


Figure 1: Comparing vanilla training process in our method with variants trained using Exact Match or Accuracy as the training signal.

accuracy, which enhances the probability of generating correct answers, thus improving both metrics. **Second**, compared to rule-based metric that only evaluate the final result, $p(y \mid \boldsymbol{z}, x; \theta)$ provides a denser and continuous signal. This offers more effective guidance on how intermediate steps in the reasoning process influence the final outcome, allowing the model to fine-tune its reasoning trajectory toward the optimal solution.

### 3.3.5 Detailed Derivation for Proposition 1

Let $r(y) \geq 0$ be an evaluation metric, and define the expected objective as $\mathcal{J}(\theta) := \mathbb{E}_{(\boldsymbol{z},y)\sim p(\boldsymbol{z},y|x;\theta)}[r(y)]$. For any proposal distribution $q(\boldsymbol{z}, y)$ over reasoning-answer trajectories, we can construct a tractable evidence lower bound (ELBO) as:

$$\text{ELBO}(\theta, q) := \sum_{\boldsymbol{z},y} q(\boldsymbol{z}, y \mid x) \log \frac{r(y)p(\boldsymbol{z}, y \mid x; \theta)}{q(\boldsymbol{z}, y \mid x)}. \tag{23}$$

To prove Proposition 1, we show that there exists a constant $c > 0$ such that:

$$\left\| \arg\max_{\theta} \text{ELBO}(\theta, q) - \arg\max_{\theta} \mathcal{J}(\theta) \right\| \leq c \cdot \left(\text{KL}(q \parallel r \cdot p_\theta)\right)^{1/2}, \tag{24}$$

where $r \cdot p_\theta$ denotes the unnormalized target distribution $q^*(\boldsymbol{z}, y) \propto r(y)p(\boldsymbol{z}, y \mid x; \theta)$.

*Proof.* We first apply Jensen's inequality to construct the ELBO:

$$\begin{aligned} \mathcal{J}(\theta) &= \sum_{\boldsymbol{z},y} r(y)p(\boldsymbol{z}, y \mid x; \theta) = \sum_{\boldsymbol{z},y} q(\boldsymbol{z}, y \mid x) \cdot \frac{r(y)p(\boldsymbol{z}, y \mid x; \theta)}{q(\boldsymbol{z}, y \mid x)} \\ &\geq \exp\left( \sum_{\boldsymbol{z},y} q(\boldsymbol{z}, y \mid x) \log \frac{r(y)p(\boldsymbol{z}, y \mid x; \theta)}{q(\boldsymbol{z}, y \mid x)} \right) = \exp(\text{ELBO}(\theta, q)). \end{aligned} \tag{25}$$

Taking logarithms (which preserves ordering due to its monotonicity), we obtain $\log \mathcal{J}(\theta) \geq \text{ELBO}(\theta, q)$ and note that $\arg\max_\theta \mathcal{J}(\theta) = \arg\max_\theta \log \mathcal{J}(\theta)$ due to the monotonicity of $\log(\cdot)$. To relate the maximization of $\text{ELBO}(\theta, q)$ and $\mathcal{J}(\theta)$, we begin by denoting $\theta^* := \arg\max_\theta \log \mathcal{J}(\theta) = \arg\max_\theta \mathcal{J}(\theta)$ and $\hat{\theta} := \arg\max_\theta \text{ELBO}(\theta, q)$. We aim to bound the distance $\|\theta^* - \hat{\theta}\|$.

We assume the objective function $\log \mathcal{J}(\theta)$ is $L$-smooth, i.e., it has $L$-Lipschitz continuous gradients[1]. This implies that for any $\theta_1, \theta_2$:

$$\log \mathcal{J}(\theta_1) \leq \log \mathcal{J}(\theta_2) + \nabla \log \mathcal{J}(\theta_2)^\top (\theta_1 - \theta_2) + \frac{L}{2} \|\theta_1 - \theta_2\|^2. \tag{26}$$

[1] https://en.wikipedia.org/wiki/Lipschitz_continuity

Applying this inequality with $\theta_1 = \theta^*$ and $\theta_2 = \hat{\theta}$, and using the fact that $\nabla \text{ELBO}(\hat{\theta}, q) = 0$ at the maximizer $\hat{\theta}$, we can write:

$$\begin{aligned} \log \mathcal{J}(\theta^*) &\leq \log \mathcal{J}(\hat{\theta}) + \nabla \log \mathcal{J}(\hat{\theta})^\top (\theta^* - \hat{\theta}) + \frac{L}{2} \|\theta^* - \hat{\theta}\|^2 \\ &\leq \log \mathcal{J}(\hat{\theta}) + \frac{L}{2} \|\theta^* - \hat{\theta}\|^2, \end{aligned} \tag{27}$$

where the last inequality assumes that the inner product term vanishes at first-order optimality. We now relate $\log \mathcal{J}(\hat{\theta})$ and $\text{ELBO}(\hat{\theta}, q)$ via:

$$\log \mathcal{J}(\hat{\theta}) - \text{ELBO}(\hat{\theta}, q) = \text{KL}(q \,\|\, r(y) \cdot p(\boldsymbol{z}, y \mid x; \hat{\theta})) + \log \mathcal{Z} \tag{28}$$

where $\mathcal{Z} = \sum_{\boldsymbol{z}, y} r(y) p(\boldsymbol{z}, y \mid x; \hat{\theta}) = \mathcal{J}(\hat{\theta})$, so the gap equals:

$$\log \mathcal{J}(\hat{\theta}) - \text{ELBO}(\hat{\theta}, q) = \text{KL}(q \,\|\, q^*) \tag{29}$$

with $q^*(\boldsymbol{z}, y) \propto r(y) p(\boldsymbol{z}, y \mid x; \hat{\theta})$.

Putting it all together, we can obtain the following equation:

$$\log \mathcal{J}(\theta^*) - \text{ELBO}(\hat{\theta}, q) \leq \frac{L}{2} \|\theta^* - \hat{\theta}\|^2 + \text{KL}(q \,\|\, q^*) \tag{30}$$

By rearranging this, we have:

$$\|\theta^* - \hat{\theta}\|^2 \leq \frac{2}{L} \cdot \Big( \log \mathcal{J}(\theta^*) - \text{ELBO}(\hat{\theta}, q) \Big) \leq \frac{2}{L} \cdot \text{KL}(q \,\|\, r \cdot p_{\hat{\theta}}) \tag{31}$$

Taking square roots gives:

$$\|\theta^* - \hat{\theta}\| \leq c \cdot \big( \text{KL}(q \,\|\, r \cdot p_{\hat{\theta}}) \big)^{1/2}, \tag{32}$$

where $c = \sqrt{\frac{2}{L}}$. This completes the proof. □

*In summary*, this proposition suggests that we can maximize ELBO as a surrogate for $\mathcal{J}$. Below, we examine when this bound is tight and show that optimizing ELBO under a specific posterior $q^*$ is equivalent to maximizing the expected objective.

**Lemma 3.4 (Tightness of the Lower Bound).** *The lower bound in Eq. 25 becomes tight if and only if the proposal distribution satisfies:*

$$q(\boldsymbol{z}, y \mid x) = q^*(\boldsymbol{z}, y \mid x) \propto r(y) \cdot p(\boldsymbol{z}, y \mid x; \theta) \tag{33}$$

*Proof.* Equality in *Jensen's inequality* holds when the log argument is constant over $\boldsymbol{z}, y$:

$$\frac{r(y) p(\boldsymbol{z}, y \mid x; \theta)}{q(\boldsymbol{z}, y \mid x)} = \text{const.}, \quad \forall \boldsymbol{z}, y. \tag{34}$$

Solving for $q(\boldsymbol{z}, y \mid x)$ yields:

$$q^*(\boldsymbol{z}, y \mid x) = \frac{r(y) p(\boldsymbol{z}, y \mid x; \theta)}{\mathcal{Z}}, \quad \mathcal{Z} = \sum_{\boldsymbol{z}, y} r(y) p(\boldsymbol{z}, y \mid x; \theta) \tag{35}$$

i.e., the normalized metric-weighted joint distribution. □

**Lemma 3.5 (Optimality of ELBO Maximization).** *If $q = q^*(\boldsymbol{z}, y \mid x) \propto r(y) p(\boldsymbol{z}, y \mid x; \theta)$, then:*

$$\arg\max_{\theta} \textit{ELBO}(\theta, q^*) = \arg\max_{\theta} \log \mathcal{J}(\theta) \tag{36}$$

*Proof.* From Lemma 3.4, if $q = q^*$, then:

$$\text{ELBO}(\theta, q^*) = \log \mathcal{J}(\theta) \Longrightarrow \arg\max_{\theta} \text{ELBO}(\theta, q^*) = \arg\max_{\theta} \mathcal{J}(\theta) \tag{37}$$

Thus, maximizing the ELBO is equivalent to maximizing the expected objective. □

# 4 More Experiment Details

## 4.1 Experimental Datasets

Following prior work [29–31, 2], we evaluate our method on a wide range of knowledge-intensive benchmarks, including Natural Questions (NQ) [32], HotpotQA [33], MuSiQue [34], and 2WikiMultihopQA (2WikiQA) [35]. Table 2 summarizes key statistics of these experimental datasets.

## 4.2 Evaluation Metrics

Following previous studies [36, 37, 2], we use the following metrics from KILT [38] for evaluation: *F1*, *Exact Match* (EM), and *Accuracy* (Acc.). *Exact Match (EM)* checks whether the predicted string exactly matches the ground truth. *Accuracy* checks whether the ground truth answer is included in the generated answer, often referred to as cover-Exact Match. *F1 Score* measures the overlap between the generated answer and the ground truth answer. It represents the harmonic mean of token-level precision and recall between these two sequences.

## 4.3 Baselines

For a comprehensive evaluation, we compare ExSearch with a range of competitive baselines, categorized into three groups based on their use of retrieval and reasoning integration strategies: (i) **Direct Reasoning without Retrieval**; (ii) **Advanced Retrieval-Augmented Generation**; and (iii) **Iterative Retrieval-Augmented Generation**.

**Direct Reasoning without Retrieval.** These methods rely solely on the LLM's internal parametric knowledge to reason over the input and generate answers, without incorporating any external information. We evaluate several recently released models, including both closed-source models, such as GPT-4o [39] and GPT-3.5 [40], as well as strong open-source models, such as DeepSeek-R1 [28], Qwen2.5 [41], QwQ-32B [42], LLaMA-3.3-70B [43], and Mistral-8x7B [44]. All of these models exhibit strong instruction-following and chain-of-thought reasoning capabilities, achieving remarkable performance on a wide range of natural language processing tasks.

**Advanced Retrieval-Augmented Generation.** These models retrieve relevant documents from an external corpus, followed by optional document filtering mechanisms such as re-ranking or summarization, and concatenate the useful information into the LLM's input context for answer generation. Specifically, we evaluate several widely used approaches, including: (i) **ChatQA** [45] and **RankRAG** [2], which unify various knowledge-intensive tasks (e.g., knowledge-grounded dialogue, document re-ranking, question answering) into a single framework, training LLMs on large-scale datasets; (ii) **InstructRAG** [46], which inserts retrieved documents into the LLM's input and trains the model to generate chains of thought to identify useful content for answer generation; (iii) **RetRobust** [47], which trains the LLM to generate accurate answers conditioned on documents containing both correct and distracting information; (iv) **Recomp** [48], which uses extractive or abstractive summarization modules to filter out irrelevant content from retrieved documents, employing a large model (e.g., Flan-UL2, 20B)[2] to generate answers from the remaining information. In our experiments, we use the abstractive summarization module, as it serves as a more competitive baseline than its extractive counterpart.

**Iterative Retrieval-Augmented Generation.** These approaches allow LLMs to actively interact with retrieval modules as needed. For a comprehensive evaluation, we include the following methods: (i) **GenGround** [49], which allows the LLM to iteratively retrieve external documents and refine its generated answer; (ii) **DSPy** [50], a programming framework that enables an LLM to decompose input queries and call external retrievers through structured prompting; (iii) **SearchChain** [36], which guides the LLM to generate a chain of queries and invoke retrieval at each step; (iv) **Iter-RetGen** [6] and **IRCoT** [51], which prompt the LLM to interact with the retriever in an iterative, few-shot manner; (v) **Verify-and-Edit** [52], which adaptively determines when to stop retrieval and finalize the answer based on the generation logits of the LLM; (vi) **Generator-Retriever-Generator** [53] (abbreviated as *Gen-Ret-Gen*), which instructs the LLM to answer a question using both model-generated and

[2] `https://huggingface.co/google/flan-ul2`

Table 2: Statistics of our experimental datasets, where we provide the amount of training and evaluation dataset, the average length of input query (word) as well as the retrieval corpus.

| Experimental benchmarks | Training data size | Query length (Train) | Evaluation data size | Query length (Evaluation) | Retrieval corpus |
|---|---|---|---|---|---|
| Nature Question [32] | 58,622 | 9.21 | 6,489 | 9.16 | Wiki2018 |
| Hotpot QA [33] | 90,185 | 17.85 | 7,384 | 15.63 | Wiki2018 |
| MusiQue QA [34] | 19,938 | 15.96 | 2,417 | 18.11 | Wiki2018 |
| 2WikiMultiHopQA [35] | 167,454 | 12.74 | 12,576 | 11.97 | Wiki2018 |

retrieved documents concatenated as context. All of the above methods are implemented on GPT-3.5, following their officially released reproducible settings. In addition, we include the following open-source baselines: (i) **Search-o1** [7], which prompts the LLM to interleave query decomposition and document retrieval steps iteratively; (ii) **Search-R1** [8], which trains the LLM to use external search engines via outcome rewards and Proximal Policy Optimization (PPO [9]); (iii) **Self-RAG** [1], which trains an LLM to retrieve documents on demand, assess their relevance, and generate final answers. This method is fine-tuned on 170k synthetic examples generated by a proprietary LLM.

### 4.4 Data Collection for Warm-Up

**Data Collection Procedure.** To construct high-quality training trajectories that reflect realistic search behavior, we design a two-stage process: (1) first, we use GPT-4o to simulate step-by-step reasoning traces in the form of interleaved sub-queries; (2) then, each sub-query is paired with a retrieved document, simulating the retrieval process of a real system rather than directly relying on officially annotated golden documents.

To synthesize pseudo training trajectories, we leverage existing datasets, such as HotpotQA [33], to generate pseudo training data for our method, as it is a widely used multi-hop QA dataset similar to the setting of our agentic search method. In HotpotQA, each example consists of a complex multi-hop query, a final answer, and a set of officially annotated sub-queries along with their corresponding supporting documents. For each question, we prompt GPT-4o to simulate a step-by-step reasoning process that interleaves sub-query generation, document retrieval, and intermediate evidence extraction. Specifically, we instruct the model to act as an intelligent search agent. Given the original multi-hop question, its gold answer, and the full set of supporting Wikipedia passages, the model is asked to restructure the reasoning path into an interleaved sequence of three special operations:

- `<THINK>` to denote a generated sub-query;
- `<SEARCH>` to indicate the citation ID of the passage used to answer the sub-query; and
- `<RECORD>` to provide the answer to the sub-query.

The output continues in this format until the final answer is reached, which is prefixed by a `<Final>` tag. An example is included in the prompt to illustrate the expected format. This approach generates high-quality, interpretable reasoning trajectories that align with the structure of our method. All synthesized outputs are included in the supplementary material for reference.

```
You are an intelligent search agent that can simulate the question-
answering process based on my question and answer.

Given an open-domain query about Wikipedia, I have already marked the
correct answer at the end of the question and provided all the reference
Wikipedia passages needed to answer the question.
Your task is to reformat my provided question and references into a
detailed question-answering process.
Specifically, there should be three types of special tokens in your
output:
1. <THINK>, followed by a sub-query
2. <SEARCH>, followed by the citation ID
3. <RECORD>, followed by the answer to the sub-query
```

````
Since this is a multi-hop question, your output should interleave the `<
THINK>`, `<SEARCH>`, and `<RECORD>` tokens until you reach the final
answer.
Please start with a special token `<Final>` followed by the final answer.

Here is a concrete example to demonstrate the output format:
```example
Question: Which magazine was started first, Arthur's Magazine or First
for Women? (Answer: Arthur's Magazine)
Reference:
[1] Arthur's Magazine | Arthur's Magazine (1844-1846) was an American
literary periodical published in Philadelphia in the 19th century. Edited
 by T.S. Arthur, it featured works by Edgar A. Poe, J.H. Ingraham, Sarah
Josepha Hale, Thomas G. Spear, and others. In May 1846, it was merged
into "Godey's Lady's Book".
[2] First for Women | First for Women is a women's magazine published by
Bauer Media Group in the USA. The magazine was started in 1989. It is
based in Englewood Cliffs, New Jersey. In 2011, the circulation of the
magazine was 1,310,696 copies.
Your Output:
<THINK> When did the magazine "Arthur's Magazine" start?
<SEARCH> [1]
<RECORD> 1844
<THINK> When did the magazine "First for Women" start?
<SEARCH> [2]
<RECORD> 1989
<Final> Arthur's Magazine
```

Starting below, for the question "{question}", please complete your
output following the above requirements.

Question: {question}
Reference:
{golden doc}
Your Output:
````

After obtaining the simulated reasoning trajectories from GPT-4o, we pair each generated sub-query (<THINK> entry) with a retrieved document. To maintain consistency with our main experiments, we use the same retrieval setup, i.e., ColBERTv2.0 as the retriever and the 2018 Wikipedia dump as the retrieval corpus. For each sub-query, we first locate its corresponding gold supporting document (provided in the official HotpotQA dataset) and use it as a reference. We then apply ColBERTv2.0 to retrieve the most similar document from the full corpus based on its proximity to the gold reference. This ensures that the retrieved documents closely reflect what a real system would retrieve while remaining grounded in the original supervision signal. The resulting data consists of interleaved sub-queries and paired retrieved documents, effectively mimicking real multi-hop retrieval trajectories. All synthesized trajectories and retrieval pairs are included in the supplementary material.

**Cost for Data Collection.** The primary cost of our data synthesis arises from using GPT-4o to transform human-annotated sub-query paths into simulated search trajectories, following the pattern used in ExSearch. In our main experiment, we generate 1,000 examples, with an average input length of 5,095.29 tokens and an average output length of 732.20 tokens. Based on OpenAI's GPT-4o pricing[3], the total cost for constructing these 1,000 examples is approximately $12.73 for input and $7.30 for output, totaling around $20. *Thus, the cost per example is* $\$\frac{20}{1000} = 0.02$. Token counts are directly obtained from OpenAI's API call messages. See the official OpenAI API documentation[4] for more details.

[3] https://platform.openai.com/docs/pricing

[4] https://platform.openai.com/docs/api-reference/introduction

### 4.5 Implementation Details and Hyperparameter

During the training stage, we trained the models with a learning rate of $2 \times 10^{-6}$, using DeepSpeed Zero 3 for efficient distributed optimization. The batch size was set to 4 for the 3B, 7B, and 8B models, and reduced to 2 for the 24B model due to memory constraints. We applied a linear warm-up (10% of total steps), followed by a cosine learning rate scheduler. All experiments used BF16 mixed-precision training with a sequence length cutoff of 8192 tokens. For each training example, we sampled 2 search trajectories (We also experimented with varying the sampling number, choosing $\{1, 2, 4, 6, 8\}$, but observed no significant difference in final performance). Table 3 summarizes the hyperparameters in our implementation. The models used in our experiments can be directly downloaded from HuggingFace, an open-source platform for machine learning and deep learning.

During the inference stage, the maximal step $T$ for the *thinking* $\rightarrow$ *search* $\rightarrow$ *recording* iteration is set to 5.

Table 3: Experimental Settings for Model Training

| Model | Batch Size | Learning Rate | Cutoff Length | Scheduler | Gradient Accumulation |
|---|---|---|---|---|---|
| Qwen-2.5-3B-instruct | 4 | $2 \times 10^{-6}$ | 8192 tokens | Cosine | 16 |
| Qwen-2.5-7B-instruct | 4 | $2 \times 10^{-6}$ | 8192 tokens | Cosine | 16 |
| Llama-3.2-3B-instruct | 4 | $2 \times 10^{-6}$ | 8192 tokens | Cosine | 16 |
| Llama-3.1-8B-instruct | 4 | $2 \times 10^{-6}$ | 8192 tokens | Cosine | 16 |
| Mistral-7B-instruct-v0.3 | 4 | $2 \times 10^{-6}$ | 8192 tokens | Cosine | 16 |
| Mistral-24B-small-2501 | 2 | $2 \times 10^{-6}$ | 8192 tokens | Cosine | 16 |

The training process of EXSEARCH is relatively straightforward to implement. Skeleton PyTorch code for EXSEARCH is demonstrated below.

```
import torch
import torch.nn.functional as F

def compute_causal_weight(ref_model, tokenizer, trajectory, answer):
    """
    Compute the reward weight w(z) = p(answer | trajectory).

    This function estimates how likely the reference model (at step t) would
    produce the correct answer y given a sampled reasoning trajectory z.

    Args:
        ref_model: The frozen language model at current iteration.
        tokenizer: The tokenizer corresponding to the model.
        trajectory: A list of messages (e.g., in chat format).
        answer: The gold answer string.

    Returns:
        weight (float): Estimated likelihood p(y | x, z), as a scalar reward.
    """
    input_ids = tokenizer.apply_chat_template(trajectory,
                                              return_tensors="pt").input_ids
    label_ids = tokenizer(answer, return_tensors="pt").input_ids

    with torch.no_grad():
        logits = ref_model(input_ids).logits[:, -label_ids.size(1):, :]
        log_probs = F.log_softmax(logits, dim=-1)
        answer_logp = torch.gather(log_probs, 2, label_ids.unsqueeze(-1))
        answer_logp = answer_logp.squeeze(-1)
        weight = answer_logp.sum().exp().item()  # Likelihood as scalar
    return weight
```

```
def M_step_learning(ref_model, input_ids, label_ids, weights):
    """
    Compute re-weighted cross-entropy loss for a given trajectory.

    This function performs the forward computation of the M-step by applying
    a scalar reward to the log-likelihood loss, encouraging trajectories
    that lead to correct answers.

    Args:
        ref_model: The language model to be updated.
        input_ids: The sampled trajectories (token IDs), shape (B, L)
        label_ids: The gold answers (token IDs), shape (B, L)
        weights: Pre-computed weights, shape (B)

    Returns:
        loss (Tensor): A scalar loss value used for gradient update.
    """
    logits = ref_model(input_ids).logits[:, :-1, :]
    labels = label_ids[:, 1:].to(logits.device)

    loss_fct = torch.nn.CrossEntropyLoss(ignore_index=-100, reduction="mean")
    loss = loss_fct(logits.view(-1, logits.size(-1)), labels.view(-1))

    # Apply pre-computed weights
    weights = torch.exp(weights - weights.max()) / weights.sum()
    loss = loss * weights.mean()  # Weight the loss by the average of the weights
    return loss
```

Table 4: Precision@K (K=3,5) for our method (*w/* 8B Llama3.1 and 7B Qwen2.5) and strong baselines.

| **Tasks** | **NQ** | | **HotpotQA** | | **MusiQue** | | **2WikiQA** | | **Avg.** | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Metrics** | P@3 | P@5 | P@3 | P@5 | P@3 | P@5 | P@3 | P@5 | P@3 | P@5 |
| ColBERTv2.0 | 41.36 | 35.97 | 28.71 | 25.20 | 6.41 | 5.44 | 17.96 | 16.10 | 23.61 | 20.68 |
| *Re-ranking* | | | | | | | | | | |
| MonoT5 [54] | 42.44 | 36.73 | 31.48 | 26.86 | 6.62 | 5.85 | 19.27 | 16.93 | 24.95 | 21.59 |
| BGE [55] | 49.55 | 40.29 | 34.50 | 28.67 | 7.70 | 6.50 | 20.45 | 18.00 | 28.05 | 23.36 |
| RankVicuna [56] | 42.68 | 36.97 | 29.49 | 26.71 | 7.09 | 6.02 | 18.75 | 17.71 | 24.50 | 21.85 |
| RankZepyhr [56] | 41.54 | 37.77 | 30.76 | 27.35 | 8.98 | 7.82 | 18.87 | 16.57 | 25.04 | 22.38 |
| *Query decomposition* | | | | | | | | | | |
| Search-o1 [7] | 42.42 | 37.01 | 42.01 | 34.11 | 17.23 | 13.13 | 25.34 | 18.24 | 31.75 | 25.62 |
| Search-r1 [8] | 40.17 | 36.86 | 27.12 | 32.67 | 4.07 | 8.30 | 16.25 | 23.49 | 21.90 | 25.33 |
| Ours-Qwen2.5-7B | 43.50 | 37.58 | 44.47 | 37.49 | 18.85 | 15.91 | 29.60 | 23.33 | 34.11 | 28.58 |
| Ours-Llama3.1-8B | 43.56 | 37.49 | 43.76 | 36.91 | 19.33 | 16.30 | 28.22 | 22.20 | 33.72 | 28.22 |

### 4.6 Supplementary Experimental Results

**Supplementary Results for Retrieval Performance** In the main body of our paper, we have report the recall@K score for the proposed EXSEARCH and strong baselines. Below, we supplements the performance in terms of precision@K score in Table 4. These results further indicate that EXSEARCH, by dynamically expanding the search as reasoning unfolds, can also improve retrieval performance in addition to enhancing end-to-end answer accuracy.

**Supplementary Results for Training Convergence Experiments.** The theoretical analysis in § 3.2 guarantees that our training objective is non-decreasing across iterations. To empirically validate this, we evaluate model checkpoints after each iteration on both the training and test sets. In the main body of our paper, we report the performance in terms of the exact match scores. In this appendix,

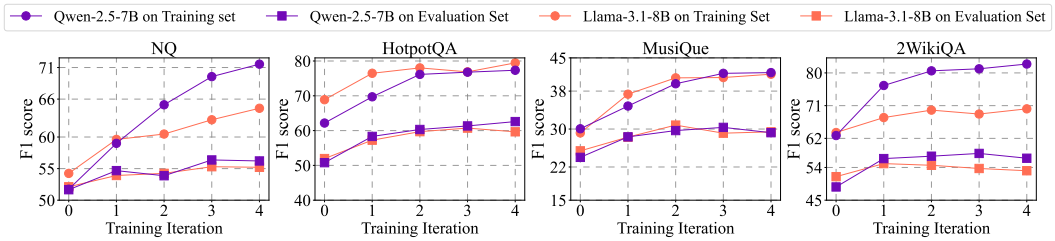


Figure 2: Training convergence of Qwen-2.5-7B and Llama-3.1-8B, where we report the *F1* score for checkpoints in each iteration. The 0th iteration indicates the initial warm-up training.

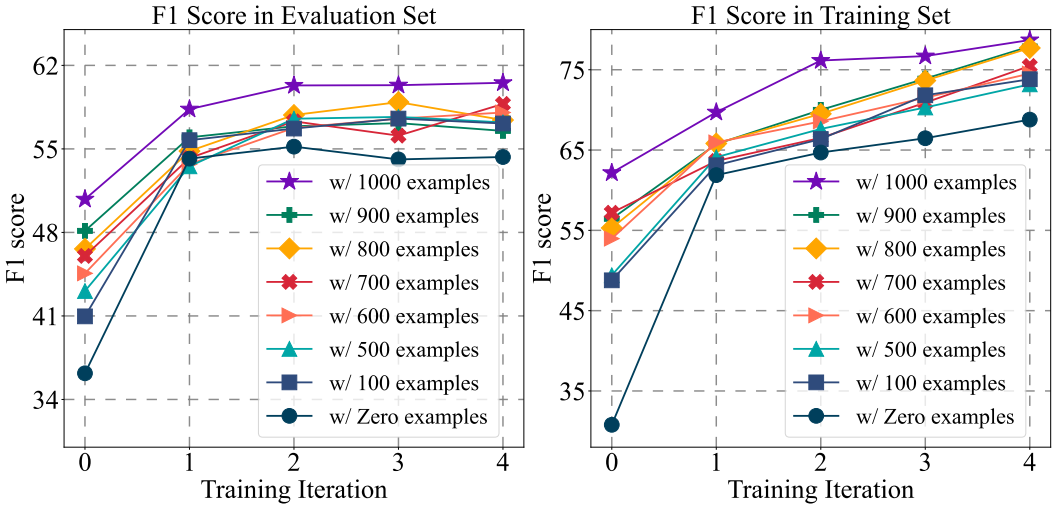


Figure 3: F1 score for ExSearch-Qwen-2.5-7B that was initially empowered by different amounts of warm-up data.

we further supplements the performance in terms of F1 metrics in Figure 2 for a more comprehensive comparison. We observe consistent performance improvement over iterations, eventually stabilizing, confirming the expected non-decreasing convergence. On evaluation sets, models typically peak by the second or third iteration, demonstrating rapid practical convergence. These results align with our theoretical guarantees and highlight the efficiency of our approach.

**Supplementary Results for ExSearch with Cold Start** To investigate the role of warm-up supervision, we train separate models with varying amounts of supervised fine-tuning (SFT) data, denoted by $K$, and apply ExSearch to each independently. The exact match (EM) scores has been reported in the main body of our paper as the primary evaluation metric. In this appendix, the corresponding F1 scores are presented in Figure 3 as a supplementary result. We observe that all models benefit from iterative self-training, with larger $K$ values leading to faster convergence and better final performance. Remarkably, even with only $K = 100$ or no warm-up supervision ($K = 0$), the models still achieve substantial gains, demonstrating the robustness of ExSearch in low-resource and cold-start scenarios. Based on the cost analysis in § 4.4, synthesizing each example costs only \$0.02 on average. Thus, empirically, in our experiment, synthesizing 100 examples incurs an approximate cost of \$2.

**Supplementary Results for Human Evaluation.** Considering the potential bias of automatic metrics [57], we conduct a human evaluation with three educated individuals assessing the *correctness* of 100 randomly sampled cases from five benchmarks, using a two-scale rating (1 for correct; 0 for incorrect). Each query is paired with the corresponding golden documents and ground truth answers from the original datasets, which serve as references for the human evaluators. We ask at least two annotators to evaluate the same case repeatedly. If there is a discrepancy between two annotators, ask a third annotator to recheck it. The results are presented in Table 5, where we found that our

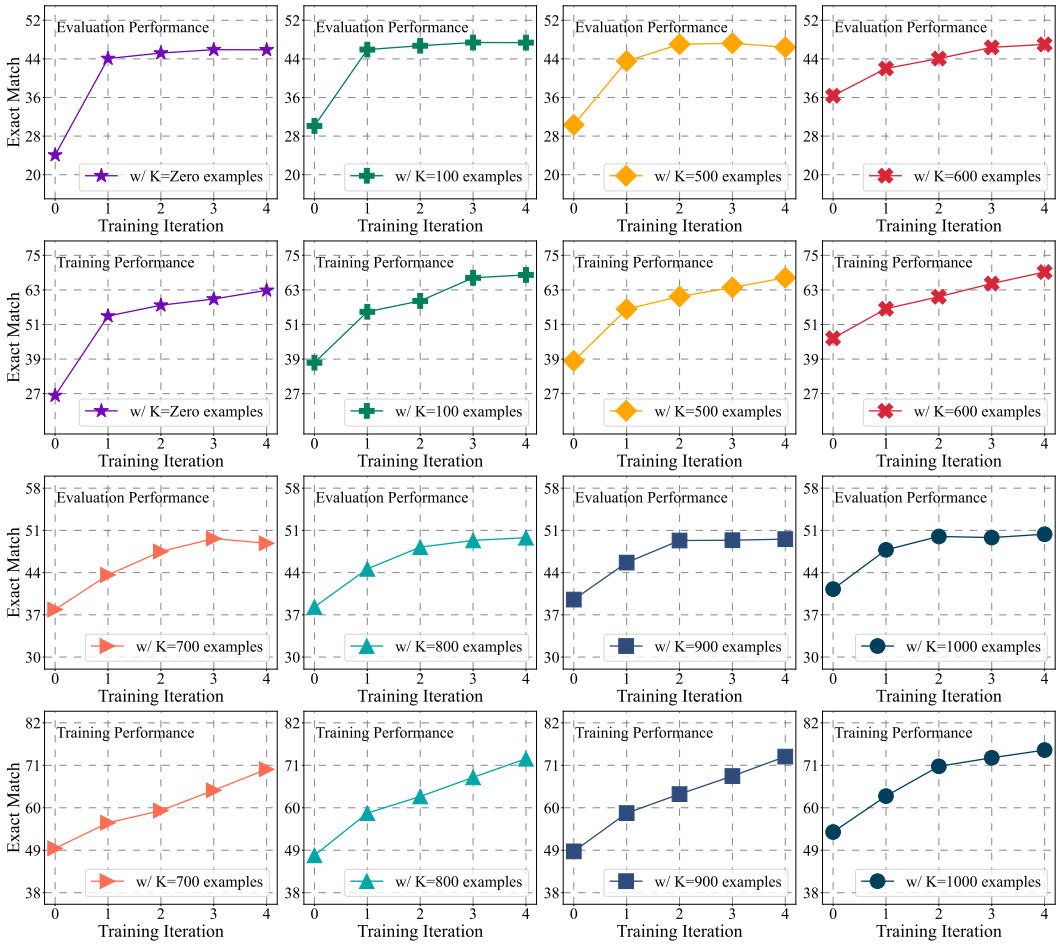


Figure 4: Exact match score for EXSEARCH-Qwen-2.5-7B, which is initially empowered by varying amounts of warm-up data, during the iterative training process in EXSEARCH.

method achieve the highest correctness, further indicating its effectiveness. In our human evaluation, the overall Kappa value is 0.771, demonstrating substantial agreement among the annotators. This indicates the reliability of the evaluation process.

Table 5: Human evaluation on 100 randomly sampled cases.

| | **GPT-4o** | **InstructRAG** | **Search-o1** | **Search-R1** | **EXSEARCH** |
|---|---|---|---|---|---|
| **Correctness** | 48/100 | 40/100 | 46/100 | 50/100 | 54/100 |

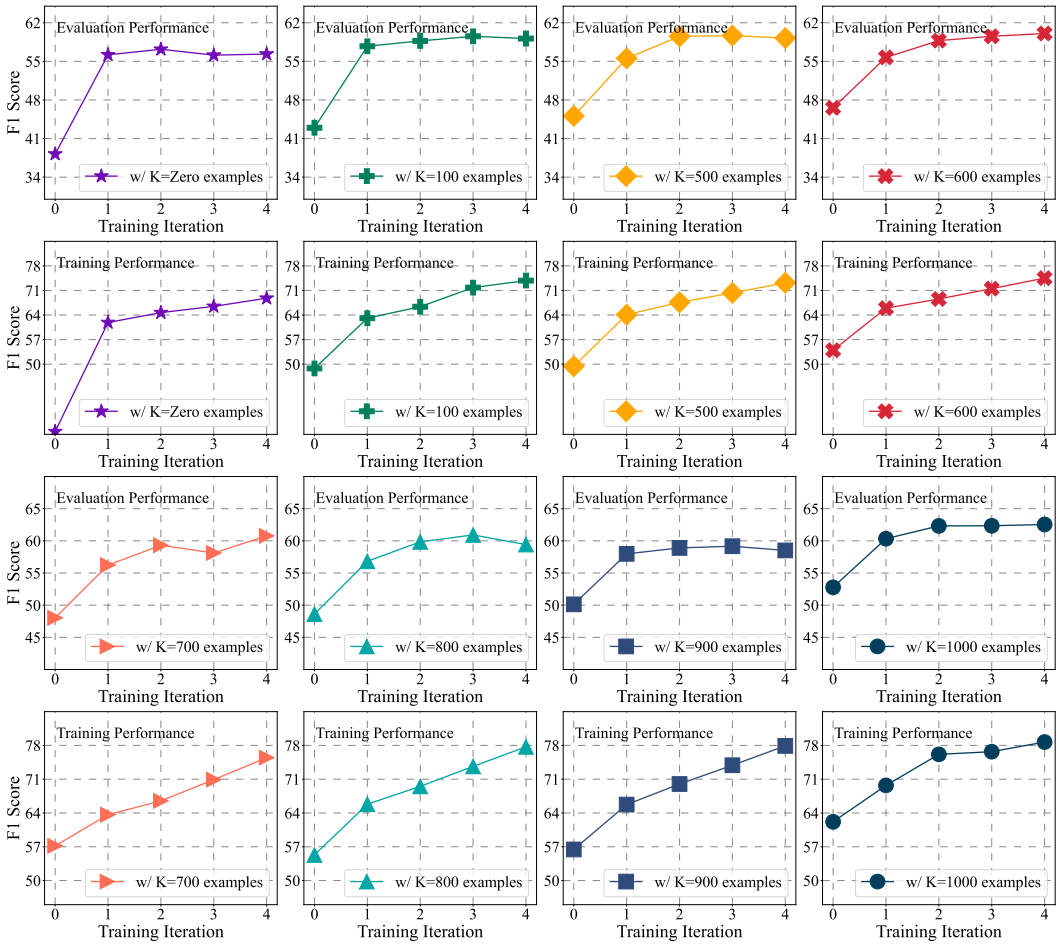


Figure 5: F1 score for EXSEARCH-Qwen-2.5-7B, which is initially empowered by varying amounts of warm-up data, during the iterative training process in EXSEARCH.

Table 6: Experiment results for applying our method to various LLMs.

| Tasks | NQ | | | HotpotQA | | | MuSiQue | | | 2WikiQA | | | Avg. | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metrics | F1 | EM | Acc. | F1 | EM | Acc. | F1 | EM | Acc. | F1 | EM | Acc. | F1 | EM | Acc. |
| Ours-Qwen-2.5-3B | 46.23 | 36.76 | 39.12 | 54.32 | 42.22 | 46.08 | 19.44 | 13.76 | 13.94 | 43.39 | 37.24 | 44.78 | 40.85 | 32.50 | 35.98 |
| Ours-Qwen-2.5-7B | 56.37 | 47.07 | 51.75 | 62.59 | 50.35 | 54.32 | 29.68 | 22.03 | 24.34 | 57.14 | 52.62 | 54.37 | 51.45 | 43.02 | 46.20 |
| Ours-Llama-3.3-3B | 41.42 | 33.49 | 35.17 | 44.12 | 33.53 | 36.14 | 17.64 | 11.23 | 11.82 | 41.73 | 36.28 | 42.22 | 36.23 | 28.63 | 31.34 |
| Ours-Llama-3.1-8B | 55.21 | 43.71 | 50.76 | 60.72 | 47.59 | 53.59 | 30.83 | 20.98 | 24.65 | 54.62 | 47.48 | 54.21 | 50.35 | 39.94 | 45.80 |
| Ours-Mistral-7B-instruct | 56.83 | 45.13 | 52.05 | 59.65 | 50.35 | 54.78 | 30.32 | 23.47 | 24.98 | 53.93 | 47.38 | 54.82 | 50.18 | 41.58 | 46.66 |
| Ours-Mistral-2501-24B | 59.89 | 47.62 | 56.39 | 67.03 | 54.51 | 59.98 | 35.84 | 23.68 | 28.54 | 60.81 | 53.19 | 61.59 | 55.89 | 44.75 | 51.63 |

# 5 EXSEARCH-Zoo: Extending EXSEARCH for Diverse Scenarios

Extensive experiments on a wide range of knowledge-intensive benchmarks demonstrate the state-of-the-art performance of EXSEARCH, as detailed in the main body of the paper. This strong performance motivates us to extend EXSEARCH to more diverse scenarios. We introduce EXSEARCH-Zoo, a comprehensive framework that enhances EXSEARCH along two key dimensions: (i) Diverse backbone LLMs across model families (LLaMA, Qwen, Mistral) and scales (7B-24B parameters); (ii) Extended actions, such as document re-ranking, to enrich the existing reasoning actions (*thinking*, *search*, and *recording*). Below, we describe how EXSEARCH is extended along each of these dimensions.

## 5.1 Diverse Model Families and Scales

We apply EXSEARCH to a range of LLMs with varying parameter sizes and model families. As shown in Table 6, the results exhibit a clear scaling-law pattern. More specifically, we highlight two key observations: First, there is a consistent performance improvement as the model size increases; Second, even models with as few as 3B parameters achieve strong performance when augmented with our method. These findings suggest that EXSEARCH is broadly applicable and scales favorably across different model families.

## 5.2 Extended Retrieval Strategy

In EXSEARCH, when faced with complex information needs, the LLM iteratively infers missing knowledge, evaluates retrieved evidence, and adapts its search strategies as new information is acquired. This behavior is formalized as a reasoning-interleaved search trajectory consisting of three core actions: **think** $\rightarrow$ **seek** $\rightarrow$ **record**, as introduced in § 3.1. Although this pattern is general, for more complex tasks, we may also need to introduce additional actions to improve the end-to-end performance. To illustrate the extensibility of EXSEARCH, we introduce an additional document re-ranking action to the vanilla EXSEARCH framework. The re-ranking step acts as a filtering mechanism, allowing the model to discard irrelevant content and focus on cleaner, more useful evidence before generating intermediate reasoning steps.

**Example: Re-ranking as a Reasoning Action.** We introduce a document re-ranking step between retrieval and evidence selection, resulting in a four-step reasoning pattern: **think** $\rightarrow$ **seek** $\rightarrow$ **rank** $\rightarrow$ **record**. Specifically, *we implement this re-ranking following generative re-ranking techniques [58, 56], where the LLM reads the retrieved documents and autoregressively generates a ranked list of selected identifiers (e.g.,* `[1] > [2] > [3]`*).*

The updated reasoning process consists of: (1) generating a sub-query $x_i$; (2) retrieving candidate documents $\boldsymbol{d}_i$ using the retriever $\mathcal{R}$; (3) re-ranking the retrieved documents to select the most relevant ones, denoted as $\hat{\boldsymbol{d}}_i$; and (4) reflecting on the selected documents $\hat{\boldsymbol{d}}_i$ by extracting an intermediate answer $e_i$ to the sub-query $x_i$.

Formally, we represent the full trajectory as a sequence of triplets $\boldsymbol{z} = \{(x_i, \hat{\boldsymbol{d}}_i, e_i)\}_{i=1}^{|\boldsymbol{z}|}$, and define its likelihood conditioned on input $x$ and parameters $\theta$ as:

$$
\begin{aligned}
p(\boldsymbol{z} \mid x; \theta) &= \prod\nolimits_{i=1}^{|\boldsymbol{z}|} p((x_i, \hat{\boldsymbol{d}}_i, e_i) \mid x, \boldsymbol{z}_{<i}; \theta) \\
&= \prod\nolimits_{i=1}^{|\boldsymbol{z}|} \underbrace{p(x_i \mid \boldsymbol{z}_{<i}; \theta)}_{\text{thinking}} \cdot \underbrace{p(\hat{\boldsymbol{d}}_i \mid x, \mathcal{R}(x_i); \theta)}_{\text{retrieval and re-ranking}} \cdot \underbrace{p(e_i \mid x_i, \hat{\boldsymbol{d}}_i; \theta)}_{\text{recording}}.
\end{aligned} \tag{38}
$$

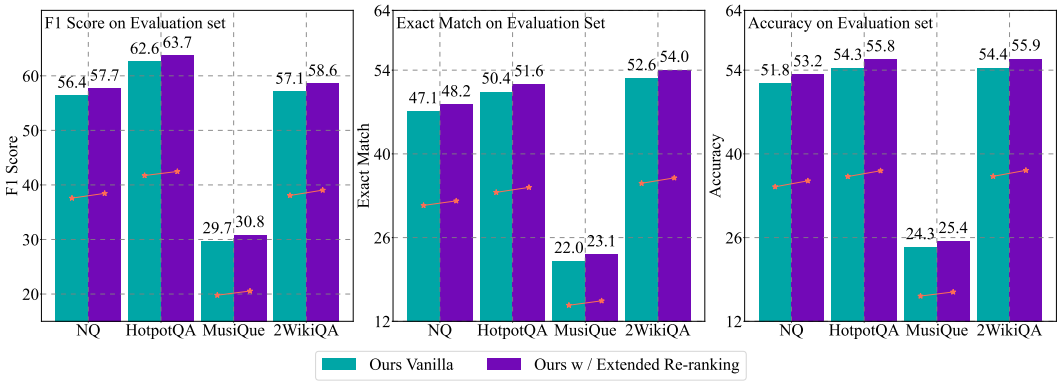


Figure 6: Performance of EXSEARCH when extended with an additional document re-ranking action, where we report the exact match score, F1 score and accuracy for a comprehensive comparison.

After finalizing a full trajectory $\boldsymbol{z}$, the LLM generates a final answer $y$ based on all intermediate reasoning steps via $y \sim p(y \mid x, \boldsymbol{z}; \theta)$.

Compared to the vanilla EXSEARCH, the only modification in the above variant is the additional document re-ranking action to the E-step, while the overall EM training framework remains unchanged. Thus, we can apply the same EM-style optimization in Eq. 10 for this variant, with the learning objective formulated as:

$$\theta = \arg\max_{\theta} \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z}|x;\theta^t)} \left[ w(\boldsymbol{z}) \log p(\boldsymbol{z} \mid x; \theta) + w(\boldsymbol{z}) \log p(y \mid x, \boldsymbol{z}; \theta) \right], \tag{39}$$

where $w(\boldsymbol{z})$ is the importance weight based on how well $\boldsymbol{z}$ supports the correct answer. The log-likelihood decomposes into two components: $\mathcal{L}_{\mathcal{R}} = \sum_{i=1}^{|\boldsymbol{z}|} \log p(x_i \mid \boldsymbol{z}_{<i}; \theta) + \log p(\hat{\boldsymbol{d}}_i \mid x, \mathcal{R}(x_i); \theta) + \log p(e_i \mid x_i, \hat{\boldsymbol{d}}_i; \theta)$ and $\mathcal{L}_{\mathcal{A}} = \log p(y \mid x, \boldsymbol{z}; \theta)$. The former $\mathcal{L}_{\mathcal{R}}$ corresponds to learning iterative search, while the latter $\mathcal{L}_{\mathcal{A}}$ corresponds to learning how to aggregate information for answer generation.

**Experimental Results** Figure 6 reports the results of incorporating re-ranking into the reasoning trajectory of EXSEARCH. We observe that adding a re-ranking step consistently improves performance across all evaluated datasets. For example, on MuSiQue and 2WikiQA, we observe average gains of +1.9 and +2.4 points in F1 score, respectively, indicating that selective document filtering can further enhance the model's ability to reason over relevant information within EXSEARCH. These results highlight the extensibility of our framework: by augmenting the action space with an additional step, we can effectively adapt the search-reasoning loop to more complex settings. This enables us to move beyond fixed action templates and incorporate new reasoning operations based on the specific requirements of downstream tasks. We believe this opens avenues for extending EXSEARCH to specialized tasks, such as retrieval-augmented fact verification, open-domain multi-hop reasoning, or tool-integrated planning, by incorporating additional task-specific actions.

## 6 Prompt and Case Study

### 6.1 System Prompt in EXSEARCH

To enable step-wise reasoning and evidence-aware retrieval, we design a structured system prompt for EXSEARCH that guides the model to simulate an intelligent search agent. The detailed content is shown below. This prompt instructs the model to decompose a complex query into sub-queries (*thinking*), retrieve relevant documents from Wikipedia based on each sub-query (*search*), and extract factual answers from the retrieved content (*recording*). This system prompt aims to encourage the LLM to perform multi-hop reasoning via an interleaved search-and-read loop, rather than attempting to answer the question in a single step. It also enforces an interpretable action trace, allowing us to diagnose the model's behavior at each stage of the reasoning process. Finally, the generation ends with a `<FINAL>` token and the final answer, enabling seamless integration into downstream pipelines and evaluation.

```
You are an intelligent search agent capable of simulating a question-
answering process by actively seeking information from Wikipedia to
answer a given question.

Specifically, given an open-domain query, please iteratively: (1)
Formulate a sub-query to search on Wikipedia; (2) Select useful documents
 from the search results and (3) Extract supporting facts from the
selected documents.
Your output should include three types of special actions corresponding
to the above steps:
(1) <THINK>: Formulate a sub-query.
(2) <SEARCH>: Retrieve and carefully read the documents using the
formulated sub-query.
(3) <RECORD>: Extract the answer to the sub-query from the documents.

Since this is a multi-hop question, your output should interleave <THINK
>, <SEARCH> and <RECORD> actions until reaching the final answer.
Conclude your output with the special token <FINIAL> followed by the
final answer.

Below is the task for you to complete:

<USER QUERY> {THE_INPUT_TASK}
Your Output:
```

### 6.2 Human Evaluation

In order to assess the end-to-end performance of the proposed model, we conducted human evaluations to verify whether the model's outputs align with the ground-truth answers. The evaluation procedure involves presenting human evaluators with a series of tasks, where they are asked to judge whether the model's prediction is consistent with the correct answer. The task provided to the human evaluators is as follows, where placeholders {question}, {model output}, and {answer} are replaced with the actual question, model's response, and the ground-truth answer, respectively. The evaluators are asked to determine whether the model's prediction implies the correct ground-truth answer.

**Guidance Human to Evaluate the Correctness of a Model Output**

In the following task, you are given a question, a model Prediction for the question, and a ground-truth Answer to the question. You should decide whether the model's Prediction implies the Ground-truth Answer.

Question
{question}

Prediction
{model output}

Ground-truth Answer
{answer}

Does the Prediction imply the Ground-truth Answer? Output Yes or No:

To guarantee annotation quality, we ask at least two annotators to evaluate the same questions repeatedly. If there is a discrepancy between the two annotators (i.e., when two annotators give a different correctness score), a third annotator is asked to review and resolve the inconsistency. To quantify the reliability of the annotations, we compute Cohen's Kappa score to measure inter-annotator agreement. A Kappa value close to 1 indicates strong agreement, while a value near 0 suggests random agreement. In our human evaluation, the overall Kappa value is 0.771, demonstrating substantial agreement among the annotators. This indicates that our evaluation process is reliable and that the results can be confidently used to assess the model's performance.

### 6.3 Case Studies

**Good Case: Correct Runtime Case of EXSEARCH.** This example demonstrates a successful reasoning trajectory executed by EXSEARCH for a compositional multi-hop question: *Navarone Garibaldi is the half-brother of a singer who has been married how many times?* The task requires (i) identifying the half-sibling relationship; and (ii) retrieving the number of marriages of the referenced singer. The model first identifies Lisa Marie Presley as Navarone Garibaldi's half-sister, then retrieves the number of times she has been married. It correctly terminates the search after the evidence is obtained and outputs the gold answer, i.e., four.

Below, in each block, the content behind <THINK> and <RECORD> represents model-generated outputs, while <SEARCH> denotes the retrieved document snippets from the external corpus.

```
INPUT QUERY: Navarone Garibaldi is the half-brother of a singer who has
been married how many times? (Golden answer: four times)
```

```
============ Turn 1 ============
<THINK> Who is Navarone Garibaldi's half-brother?
<SEARCH> [0] Title: Navarone Garibaldi. Content: Navarone Garibaldi
Navarone Anthony Garibaldi (born March 1, 1987) is an American musician.
He is the frontman of the band Them Guns. Garibaldi is the son of Marco
Garibaldi and Priscilla Presley, and also half-brother of Lisa Marie
Presley. Navarone Anthony Garibaldi was born to actress Priscilla Presley
 and Marco Garibaldi on March 1, 1987 in Santa Monica, California. With
his mother Priscilla being the former wife of the late Elvis Presley,
media scrutiny was often focused on the family. Garibaldi's half-sister
is Lisa Marie Presley, Elvis's only child. He is the uncle of Riley and
Benjamin Keough and Harper... [2]... [3]...
<RECORD> Lisa Marie Presley
```

```
============ Turn 2 ============
<THINK> How many times has Lisa Marie Presley been married?
<SEARCH> [0] Title: Lisa Marie Presley. Content: Lisa Marie Presley Lisa
Marie Presley (born February 1, 1968) is an American singer-songwriter.
She is the only child of singer and actor Elvis Presley and actress
Priscilla Presley, as well as the sole heir to her father's estate.
Presley has developed a career in the music business and has issued three
 albums. She has been married four times, including to singer Michael
Jackson and actor Nicolas Cage, before marrying music producer Michael
Lockwood, father of her twin girls. Lisa Marie was born on February 1,
1968, to Elvis and Priscilla Presley at Baptist Memorial Hospital in
Memphis, Tennessee, nine
<RECORD> four
```

```
============ Turn 3 (answer aggregation) ============
<FINAL>
<SEARCH> end search.
The final answer to the input question "Navarone Garibaldi is the half-
brother of a singer who has been married how many times?"
Output: four
```

**Bad Case: Over-Searching due to Missed Reasoning.** We present a concrete example below, where the LLM gives a correct answer but overly relies on the retrieval module. In Turn 1 and Turn 2, the model successfully retrieves the occupations of both individuals via two well-formed sub-queries. However, instead of reasoning over the retrieved facts to answer the question, the model repeatedly issues similar external queries (Turn 3–5), such as *Are there any shared occupations between Maher Sabry and Til Schweiger?* These repeated queries yield no new information and reflect an over-dependence on retrieval, rather than using previously acquired knowledge for local inference. This highlights a limitation in current retrieval-augmented models: they often fail to consolidate retrieved evidence before issuing additional queries, resulting in unnecessary search steps.

```
USER QUERY: What occupations are shared by Maher Sabry and Til Schweiger?
 (Golden answer: director, and producer)
```

```
============ Turn 1 ============
<THINK> What is Maher Sabry's occupation?
<SEARCH> [1] Title: Maher Sabry. Content: Maher Sabry is an Egyptian
theater director, playwright, film director, producer and screenwriter,
poet, writer and cartoonist. A gay activist...; [2]...; [3]...
<RECORD> Maher Sabry's occupations include theater director, playwright,
film director, producer, screenwriter, poet, writer, and cartoonist.
```

```
============ Turn 2 ============
<THINK> What is Til Schweiger's occupation?
<SEARCH> [1] Title: Til Schweiger. Content: Til Schweiger Tilman Valentin
 Til Schweiger (born 19 December 1963) is a German actor, voice actor,
film director, film producer, and film editor. He runs his own production
 company, ...; [2] ... [3] ...
<RECORD> Til Schweiger's occupations include actor, voice actor, film
director, film producer, and film editor.
```

```
============ Turn 3 ============
<THINK> Are there any shared occupations between Maher Sabry and Til
Schweiger?
<SEARCH> [1] Title: Maher Sabry. Content: Maher Sabry is an Egyptian
theater director, playwright, film director, producer and screenwriter,
poet, writer and cartoonist. A gay activist,... [2] ... [3] ...
<RECORD> Maher Sabry's occupations include theater director, playwright,
film director, producer, screenwriter, poet, writer, and cartoonist.
```

```
============ Turn 4 ============
<THINK> Are there any shared occupations between Maher Sabry and Til
Schweiger?
<SEARCH> [1] Title: Maher Sabry. Content: Maher Sabry is an Egyptian
theater director, playwright, film director, producer and screenwriter,
poet, writer and cartoonist. A gay activist,... [2] ... [3] ...
<RECORD> Maher Sabry's occupations include theater director, playwright,
film director, producer, screenwriter, poet, writer, and cartoonist.
```

```
============ Turn 5 ============
<THINK> Are there any shared occupations between Maher Sabry and Til
Schweiger?
<SEARCH> [1] Title: Maher Sabry. Content: Maher Sabry is an Egyptian
theater director, playwright, film director, producer and screenwriter,
poet, writer and cartoonist. A gay activist,... [2]... [3]...
<RECORD> Maher Sabry's occupations include theater director, playwright,
film director, producer, screenwriter, poet, writer, and cartoonist.
```

```
============ Turn 6 ============
<THINK> Are there any shared occupations between Maher Sabry and Til
Schweiger?
<SEARCH> [1] Title: Maher Sabry. Content: Maher Sabry is an Egyptian
theater director, playwright, film director, producer and screenwriter,
poet, writer and cartoonist. A gay activist,... [2] ... [3] ...
<RECORD> Maher Sabry's occupations include theater director, playwright,
film director, producer, screenwriter, poet, writer, and cartoonist.
```

```
============ Turn 7 ============
<THINK> Are there any shared occupations between Maher Sabry and Til
Schweiger?
```

```
<SEARCH> [1] Title: Maher Sabry. Content: Maher Sabry is an Egyptian
theater director, playwright, film director, producer and screenwriter,
poet, writer and cartoonist. A gay activist,... [2] ... [3] ...
<RECORD> Maher Sabry's occupations include theater director, playwright,
film director, producer, screenwriter, poet, writer, and cartoonist.
```

```
============ Turn 8 (Up to the maximum search times) ============
<FINAL> theater director, playwright, film director, producer,
screenwriter
please give the final answer to the input question: What occupations are
shared by Maher Sabry and Til Schweiger?
Playwright, film director, producer, screenwriter
```

**Bad Case: Under-Searching due to Premature Entity Binding.** In this example, the user asks a seemingly straightforward compositional question: *Into Dust is a song by the alternative rock band formed in which city?* The correct answer is Santa Monica, California, as the song *Into Dust* is performed by the alternative rock band Mazzy Star. However, the model prematurely assumes an entity match based on a partial overlap with the query phrase "Into Dust," mistakenly linking it to a different band named *Cities in Dust*, a Canadian punk group. As shown in the first retrieval result, this band has no association with the queried song but is erroneously selected due to superficial string-level similarity and genre proximity. The model then terminates the search after this single retrieval step and outputs Hamilton, Ontario, as the final answer, based solely on the retrieved band's origin. This reflects a classic under-searching failure mode, where the model fails to verify whether the retrieved entity is semantically compatible with the query. In this case, a simple clarification query, such as "Who performed *Into Dust* (song)?" would have revealed the correct band and corresponding city. This example highlights the need for disambiguation-aware reasoning mechanisms that delay answer generation until entity identity is sufficiently grounded.

```
USER QUERY: Into Dust is a song by the alternative rock band formed in
which city? (Golden answer: Santa Monica, California)
```

```
==========
<THINK> Which city was the alternative rock band that performed "Into
Dust" formed in?
<SEARCH> [1] Title: Cities in Dust (band). Content: Week in Toronto.
Independent record label Paper Bag Records signed Cities in Dust in April
 2006, in part due to the onstage antics of Frank. Soon after, their
debut album Night Creatures was released. Cities in Dust appeared at the
World Electronic Music Festival in June 2006. The band was nominated for
three awards at the 2006 Hamilton Music Awards: Record of the Year, New
Artist/Group of the Year, Alternative Recording of the Year. Night
Creatures was named by Exclaim! magazine as one of the Top 10 Punk albums
 of 2006 in Canada. and it received heavy rotation on CBC
[2] ... [3]...
<RECORD> Hamilton, Ontario
```

```
USER QUERY: Into Dust is a song by the alternative rock band formed in
which city?
please give the final answer to the input question: Into Dust is a song
by the alternative rock band formed in which city?
Hamilton, Ontario
```

Table 7: Case Study of Under-Searching Caused by Entity Confusion and Early Termination.

| Aspect | Observation |
|---|---|
| **User Query** | Into Dust is a song by the alternative rock band formed in which city? |
| **Gold Answer**<br>**Retrieved Fact** | Santa Monica, California (Mazzy Star)<br>Hamilton, Ontario (based on Cities in Dust band) |
| **Search Count** | 1 |
| **Entity Linking Error** | Mistook "Into Dust" as a song by "Cities in Dust" |
| **Failure Mode** | Under-searching: Incorrect assumption based on first retrieved entity, no disambiguation step. |
| **Suggested Fix** | Add entity verification query: *"Who performed Into Dust?"* or *"Into Dust band name"* before inferring location. |

Table 8: Case study illustrating over-searching due to lack of intermediate reasoning.

| Aspect | Observation |
|---|---|
| **User Query** | What occupations are shared by Maher Sabry and Til Schweiger? |
| **Query Type** | Compositional: Requires comparing two entities' attributes. |
| **Sub-Queries Issued** | Turn 1: What is Maher Sabry's occupation?<br>Turn 2: What is Til Schweiger's occupation? |
| **Knowledge Retrieved** | All relevant occupations were retrieved correctly for both individuals by Turn 2. |
| **Expected Behavior**<br>**Observed Behavior** | The model should compare two sets of occupations and output the shared ones.<br>Model redundantly issues semantically similar queries in Turn 3–5, such as: *"Are there any shared occupations between Maher Sabry and Til Schweiger?"* |
| **Failure Mode** | Over-searching: The model repeatedly queries despite having sufficient information to answer. |
| **Impact** | Increased search cost, unnecessary API calls, and longer latency with no gain in answer quality. |
| **Insight** | Reasoning over retrieved evidence should be prioritized once coverage is sufficient; redundant querying should be suppressed. |

Table 9: Comparison between a successful multi-hop reasoning trajectory and a failure case due to over-searching. The successful case stops searching after gathering sufficient evidence, while the failure case redundantly queries instead of reasoning.

| Step | Successful Reasoning | Over-Searching Failure |
|---|---|---|
| **Input Query** | *Navarone Garibaldi is the half-brother of a singer who has been married how many times?* | *What occupations are shared by Maher Sabry and Til Schweiger?* |
| **Turn 1** | <THINK> Who is Navarone Garibaldi's half-sister?<br><SEARCH> → Lisa Marie Presley<br><RECORD> Lisa Marie Presley | <THINK> What is Maher Sabry's occupation?<br><SEARCH> → Maher Sabry: [occupations]<br><RECORD> theater director, screenwriter, etc. |
| **Turn 2** | <THINK> How many times has she been married?<br><SEARCH> → [four marriages]<br><RECORD> four | <THINK> What is Til Schweiger's occupation?<br><SEARCH> → Til Schweiger: [occupations]<br><RECORD> actor, producer, etc. |
| **Turn 3** | <FINAL> The answer is: four.<br>Search terminated successfully. | <THINK> Are there shared occupations?<br><SEARCH> Maher Sabry (again)<br><RECORD> (repetition) |
| **Failure Mode** | N/A – correct answer produced with minimal hops. | Over-searching: fails to reason over retrieved evidence, keeps querying. |
| **Insight** | Success relies on using retrieved facts to trigger answer generation. | The model needs an early stop or a reasoning trigger mechanism. |

## References


[1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

[2] Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *arXiv preprint arXiv:2407.02485*, 2024.

[3] Yunjia Qi, Hao Peng, Xiaozhi Wang, Bin Xu, Lei Hou, and Juanzi Li. Adelie: Aligning large language models on information extraction. *arXiv preprint arXiv:2405.05008*, 2024.

[4] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[5] Omar Adjali, Olivier Ferret, Sahar Ghannay, and Hervé Le Borgne. Multi-level information retrieval augmented generation for knowledge-based visual question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

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

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

[8] Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

[9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[10] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 1996.

[11] Peter Henderson, Joshua Romoff, and Joelle Pineau. Where did my optimum go?: An empirical analysis of gradient descent optimization in policy gradient methods. *arXiv preprint arXiv:1810.02525*, 2018.

[12] Shengyi Huang, Rousslan Fernand Julien Dossa, Antonin Raffin, Anssi Kanervisto, and Weixun Wang. The 37 implementation details of proximal policy optimization. In *The ICLR Blog Track 2023*, 2022.

[13] Jian Hu, Siyang Jiang, Seth Austin Harding, Haibin Wu, and Shih-wei Liao. Rethinking the implementation tricks and monotonicity constraint in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2102.03479*, 2021.

[14] Haolin Chen, Yihao Feng, Zuxin Liu, Weiran Yao, Akshara Prabhakar, Shelby Heinecke, Ricky Ho, Phil Mui, Silvio Savarese, Caiming Xiong, et al. Language models are hidden reasoners: Unlocking latent reasoning capabilities via self-rewarding. *arXiv preprint arXiv:2411.04282*, 2024.

[15] Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594*, 2024.

[16] Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T Parisi, et al. Beyond human data: Scaling self-training for problem-solving with language models. *Trans. Mach. Learn. Res.*, 2023.

[17] Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! In *The International Conference on Learning Representations*, 2019.

[18] Junzi Zhang, Jongho Kim, Brendan O'Donoghue, and Stephen Boyd. Sample efficient reinforcement learning with reinforce. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[19] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

[20] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 2024.

[21] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

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

[23] Gary Marchionini. Exploratory search: from finding to understanding. *Commun. ACM*, 49, 2006. ISSN 0001-0782.

[24] Víctor Elvira and Luca Martino. Advances in importance sampling. *arXiv preprint arXiv:2102.05407*, 2021.

[25] Surya T Tokdar and Robert E Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2, 2010.

[26] John Bibby. Axiomatisations of the average and a further generalisation of monotonic sequences. *Glasgow Mathematical Journal*, 1974.

[27] Peter Dayan and Geoffrey E. Hinton. Using expectation-maximization for reinforcement learning. *Neural Computation*, 1997.

[28] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[29] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

[30] Xiaoxi Li, Zhicheng Dou, Yujia Zhou, and Fangchao Liu. Corpuslm: Towards a unified language model on corpus for knowledge-intensive tasks. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*, 2024.

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

[32] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 2019.

[33] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[34] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. In *Transactions of the Association for Computational Linguistics: TACL*, 2022.

[35] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.

[36] Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. Search-in-the-chain: Towards accurate, credible and traceable large language models for knowledge-intensive tasks. In *WWW*, 2024.

[37] Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. Investigating the factual knowledge boundary of large language models with retrieval augmentation. In *ACL*, 2023.

[38] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

[39] OpenAI. GPT-4, 2023.

[40] OpenAI. Introducing ChatGPT, 2022.

[41] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[42] Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, 2024.

[43] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[44] Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *ArXiv*, 2023.

[45] Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. Chatqa: Surpassing gpt-4 on conversational qa and rag. *arXiv preprint arXiv:2401.10225*, 2024.

[46] Zhepei Wei, Wei-Lin Chen, and Yu Meng. InstructRAG: Instructing retrieval-augmented generation via self-synthesized rationales. In *The Thirteenth International Conference on Learning Representations*, 2025.

[47] Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*, 2023.

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

[49] Zhengliang Shi, Weiwei Sun, Shen Gao, Pengjie Ren, Zhumin Chen, and Zhaochun Ren. Generate-then-ground in retrieval-augmented generation for multi-hop question answering. *arXiv preprint arXiv:2406.14891*, 2024.

[50] O. Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, et al. Dspy: Compiling declarative language model calls into state-of-the-art pipelines. In *International Conference on Learning Representations*, 2024.

[51] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada, 2023.

[52] Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. *arXiv preprint arXiv:2305.03268*, 2023.

[53] Abdelrahman Abdallah and Adam Jatowt. Generator-retriever-generator approach for open-domain question answering. *arXiv preprint arXiv:2307.11278*, 2023.

[54] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

[55] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

[56] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*, 2023.

[57] Zhengliang Shi, Weiwei Sun, Shuo Zhang, Zhen Zhang, Pengjie Ren, and Zhaochun Ren. Rade: Reference-assisted dialogue evaluation for open-domain dialogue. *ArXiv*, 2023.

[58] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.