DeepNote: Note-Centric Deep Retrieval-Augmented Generation

Ruobing Wang[1,2], Qingfei Zhao[1,2], Yukun Yan[3†], Daren Zha[1], Yuxuan Chen[4], Shi Yu[3], Zhenghao Liu[5], Yixuan Wang[3], Shuo Wang[3], Xu Han[3], Zhiyuan Liu[3], Maosong Sun[3†]

[1]Institute of Information Engineering, Chinese Academy of Sciences;
[2]School of Cyber Security, University of Chinese Academy of Sciences;
[3]Department of Computer Science and Technology, Institute for AI, Tsinghua University;
[4]South China University of Technology; [5] Northeastern University
{wangruobing}@iie.ac.cn

Abstract

Retrieval-Augmented Generation (RAG) mitigates factual errors and hallucinations in Large Language Models (LLMs) for question-answering (QA) by incorporating external knowledge. However, existing adaptive RAG methods rely on LLMs to predict retrieval timing and directly use retrieved information for generation, often failing to reflect real information needs and fully leverage retrieved knowledge. We develop **DeepNote**, an adaptive RAG framework that achieves in-depth and robust exploration of knowledge sources through note-centric adaptive retrieval. DeepNote employs notes as carriers for refining and accumulating knowledge. During in-depth exploration, it uses these notes to determine retrieval timing, formulate retrieval queries, and iteratively assess knowledge growth, ultimately leveraging the best note for answer generation. Extensive experiments and analyses demonstrate that DeepNote significantly outperforms all baselines (+10.2% to +20.1%) and exhibits the ability to gather knowledge with both high density and quality. Additionally, DPO further improves the performance of DeepNote. The code and data are available at `https://github.com/thunlp/DeepNote`.

1 Introduction

Large Language Models (LLMs) (OpenAI, 2023; Touvron et al., 2023) capture versatile knowledge (Shultz et al., 2024) through billions of parameters, boosting performance in question-answering (QA) tasks. However, even state-of-the-art LLMs can encounter hallucinations (Chen et al., 2023) and factual errors (Mallen et al., 2023; Min et al., 2023). Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) is a widely used technique that leverages external non-parameterized knowledge resources to help LLMs push their inherent parameter knowledge boundaries to mitigate these issues. However, Vanilla RAG usually fails to gather sufficient information for complex QA tasks w.r.t. long-form QA (Stelmakh et al., 2022; Lyu et al., 2024), and multi-hop QA (Yang et al., 2018). These complex QA tasks often involve broad or in-depth information retrieval needs, which may not be explicitly reflected in the initial query or easily fulfilled in a single retrieval attempt.

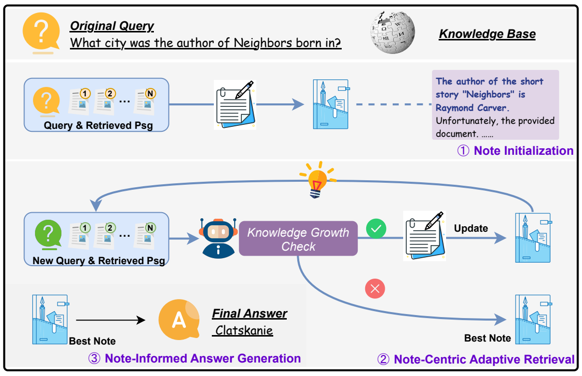

Figure 1: **Illustration of DeepNote.** DeepNote fully integrates knowledge retrieved across multiple iterations using notes as the knowledge carrier and employs the best note to formulate retrieval decisions.

Recently, several works (Jiang et al., 2023; Asai et al., 2024) have proposed adaptive RAG (ARAG), which enables adaptively capture more valuable knowledge for answering complex questions. Despite their success, they still have two limitations. ***First***, each retrieval triggers an immediate generation. This approach may cause each output segment to reflect limited knowledge from a specific retrieval iteration, neglecting the integration and interaction of information across different retrieval iterations. ***Second***, they leverage LLMs to actively predict retrieval timing; however, differences between the LLMs' internal cognition and the actual retrieval needs may lead to missing key knowledge.

To address them, we present **DeepNote**, an ARAG framework that utilizes notes as knowledge

†Corresponding authors

carriers to deeply and robustly explore knowledge bases for answering complex questions. DeepNote comprises three key processes: note initialization, note-centric adaptive retrieval, and note-informed answer generation. **As depicted in Figure 1**, in the note initialization process, we first construct an initial note as the starting point for adaptive retrieval, treating it as the best note. In the note-centric adaptive retrieval process, we continuously use the best note to guide the system in making optimal forward retrieval decisions, and update the note with newly retrieved information from a view of knowledge growth. During each retrieval iteration, the model is encouraged to review and compare the latest note with the best note. In the answer generation process, the system leverages the best note to generate comprehensive and accurate answers.

Extensive empirical experiments conducted on five datasets (including both complex and simple QA), demonstrate that DeepNote can effectively, robustly, and deeply explore knowledge bases. The overall performance of DeepNote significantly surpasses that of Vanilla RAG (up to +20.1%) and a range of previous mainstream methods (up to +10.2%), confirming its superiority. We also develop an automated fine-tuning data construction pipeline and a training dataset, DNAlign, to enhance the model's instruction-following capabilities across multiple task stages and align with high-quality response preferences. Empirical results on Llama3.1-8B and Qwen2.5-7B indicate that performing DPO with DNAlign further improves our framework's performance across all datasets. Additionally, multi-dimensional analysis demonstrates that our framework can gather high-quality and comprehensive information with higher knowledge density, while effectively balancing retrieval efficiency and performance.

2 Related Work

2.1 Retrieval-Augmented Generation (RAG)

Through knowledge augmentation, RAG (Ram et al., 2023; Lewis et al., 2020; Guu et al., 2020) helps LLMs mitigate issues such as hallucinated outputs (Chen et al., 2023; Zuccon et al., 2023), out-of-date knowledge and long-tail knowledge gaps (He et al., 2023; Kandpal et al., 2023), while extending LLMs beyond their knowledge boundaries (Yin et al., 2023b). In QA tasks (Baek et al., 2023; Siriwardhana et al., 2023; Voorhees, 1999), Vanilla RAG typically employs a retriever (Karpukhin et al., 2020) to fetch external knowledge from the corpus and incorporates it as text into the input space of LLMs, thereby enhancing the quality of answer. Some previous methods (Yu et al., 2023; Izacard et al., 2023) adopt a single-step RAG method, where the retrieved passages are processed for knowledge refinement before generating the final answer. However, they fail to directly retrieve sufficient information, especially in complex QA tasks. One line of studies (Trivedi et al., 2023; Borgeaud et al., 2022; Ram et al., 2023; Press et al., 2023; Wang et al., 2024) attempt multi-step RAG during generation to alleviate this issue. Another line of recent studies (Jiang et al., 2023; Yao et al., 2023; Asai et al., 2024; Jeong et al., 2024) propose ARAG systems, which can automatically determine "*when and what to retrieve*" via various feedbacks. However, they may fail to actively predict true retrieval needs and timing through the LLM's parametric cognition and lack interaction with knowledge retrieved across multiple iterations. Therefore, our work aims to establish a note-centric adaptive RAG that fully integrates knowledge retrieved across multiple iterations and uses the best note to guide retrieval decisions.

2.2 Fine-Tuning for RAG

Fine-tuning is widely used to improve the capabilities of LLM-augmented components in RAG systems (de Luis Balaguer et al., 2024). Early methods of fine-tuning to enhance LLM-based components in RAG primarily focused on training the retriever and the generator (Ke et al., 2024; Lin et al., 2024). Recent RAG methods have shifted toward modular designs (Gao et al., 2023b). Particularly in complex QA tasks, adaptive RAG often requires base models to follow intricate instructions (Yin et al., 2023a; Xu et al., 2024) to enable the functionality of diverse components (Asai et al., 2024). Classic alignment training methods include supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF). However, SFT lacks negative feedback and is prone to overfitting. Recently, Rafailov et al. proposed a more efficient reinforcement learning algorithm, direct preference optimization (DPO), which aligns response preferences and enhances the model's instruction-following ability by learning the differences between positive and negative sample pairs. In our work, we focus on using DPO to enhance the model's capability in multiple processes.

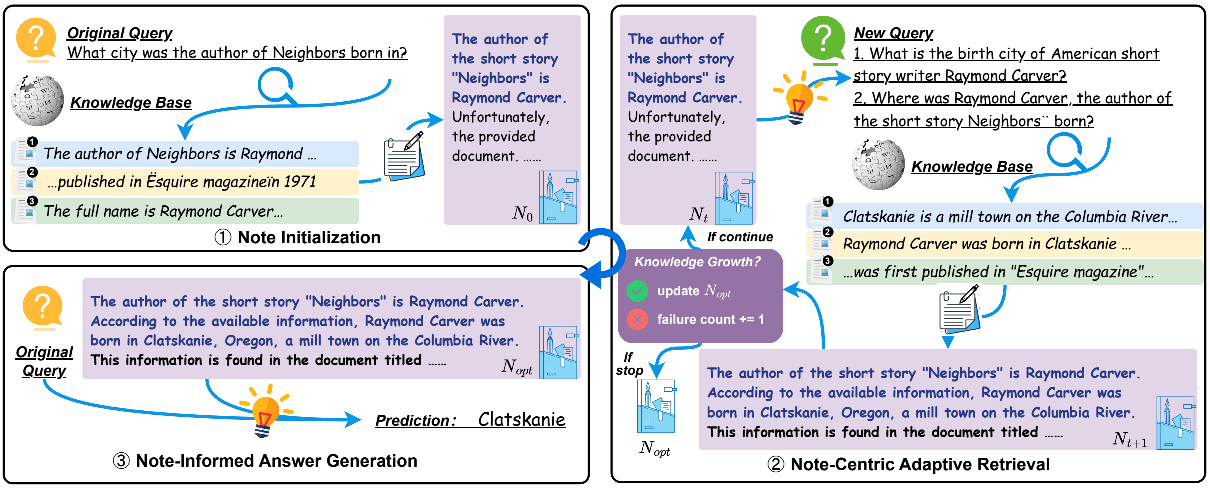

Figure 2: **Overview of DeepNote.** DeepNote consists of three processes: Note Initialization, Note-Centric Adaptive Retrieval, and Note-Informed Answer Generation. We employ a note-centric strategy to formulate retrieval decisions (including "when and what to retrieve"), accumulate knowledge, and generate answers.

3 Methodology

In this section, we first introduce three key processes (§ 3.1, § 3.2, and § 3.3) of **DeepNote**, with an overview illustrated in Figure 2. We then introduce our training dataset DNAlign, its automated construction pipeline (§ 3.4), and the training process (§ 3.5).

3.1 Note Initialization

To enhance the model's awareness of useful knowledge while minimizing noise during adaptive exploration, we introduce a note as the knowledge carrier. We start with an original query q_0, then retrieve top-k passages $P_{k,0} = \{p_1, p_2, \ldots, p_k\}$ as references. We observe that since the system fails to foresee the characteristics and aspects of the retrieved knowledge, a fine-grained note construction approach, where notes are strictly summarized from predefined aspects or domains, often leads to misalignment between the collected knowledge and the actual relevant information. Therefore, we delegate reasoning and decision-making entirely to the LLM, providing only the highest-level objective to facilitate its flexible and comprehensive collection of knowledge that supports answering or reasoning about the q_0. We now formalize this process:

$$N_0 \sim \mathrm{LLM}_{\mathrm{Init}}(\mathrm{Instruct}_{\mathrm{Init}}, q_0 \| P_{k,0}) \qquad (1)$$

where we use the prompt template $\mathrm{Instruct}_{\mathrm{Init}}$ to instruct LLM to generate the initial note N_0. The $\mathrm{LLM}_{\mathrm{Init}}(\cdot)$ denotes the backbone model used in the note initialization.

3.2 Note-Centric Adaptive Retrieval

To effectively and deeply explore the unknown semantic space of the corpus, we develop a note-centric, three-stage adaptive retrieval process.

Query Refinement In this stage, we leverage the distilled knowledge stored in the note to formulate the new query q_t for further retrieval. Specifically, we only have the initial note N_0 as a reference after the note initialization process τ_0. Thus, in iteration τ_1, we regard N_0 as N_{Opt}. In each iteration τ_t, the input consists of the q_0, the list of previously generated queries, and the best note so far. Among them, the best note[1] so far refers to the note selected as the best choice by comparing it with the previous iteration's best note, denoted as N_{Opt}. This recursive comparison process resembles how humans integrate and learn new knowledge, as they tend to formulate new questions based on their existing optimal understanding. Additionally, the list of previously generated queries includes new queries generated in all previous iterations $\tau_{<t}$, denoted as $Q_t^{\mathrm{Pre}} := \{q_1, q_2, \ldots, q_{t-1}\}$. This design stems from our observation that the LLM tends to repeatedly generate highly similar queries if issues raised in earlier iterations remain unresolved. To prevent the system from getting trapped in localized exploration, we introduce Q^{Pre} to eliminate the generation of redundant or ineffective queries. To

[1]The generation of the best note N_{Opt} is a recursive process, where N_{Opt} in the current iteration τ_t is defined using the best note N_{Opt} from the iteration τ_{t-1} along with other variables. Therefore, we provide a detailed definition of N_{Opt} in the adaptive retrieval decision stage in § 3.2.

sum up, the process can be formalized as follows:

$$q_t \sim \text{LLM}_{\text{QR}}(\text{Instruct}_{\text{QR}}, q_0 \| N_{\text{Opt}} \| Q_t^{\text{Pre}}) \quad (2)$$

Equation (2) clearly illustrates the process of generating new queries q_t for further retrieval in iteration τ_t, where $t \geq 1$. The $\text{Instruct}_{\text{QR}}$ and $\text{LLM}_{\text{QR}}(\cdot)$ represent the prompt template and backbone model of the process in the query refinement stage.

Knowledge Accumulation Our goal is to leverage new queries to explore potential query-relevant semantic subspaces within the corpus for knowledge accumulation. We guide the LLM from a view of "*how to foster stable and effective knowledge growth*" for complex information collection, refinement, and updating. Specifically, we first use a new query q_t to retrieve top-k passages $P_{k,t}$. Next, we construct a note-updating workflow informed by multi-dimensional guidance.

$$N_t \sim \text{LLM}_{\text{KA}}(\text{Instruct}_{\text{KA}}, q_0 \| N_{\text{Opt}} \| P_{k,t}) \quad (3)$$

Equation (3) presents the process of note updating for knowledge accumulation using the model LLM_{KA}. The $\text{Instruct}_{\text{KA}}$ denotes the prompt template, where we provide a detailed workflow. In this workflow, we require that the knowledge incorporated into updated notes N_t remains faithful to the retrieved passages $P_{k,t}$, meaning that the collected information should follow their style and, whenever possible, use direct excerpts. This strategy aims to minimize the introduction of parametric knowledge over deep iterative processes, which could otherwise lead to knowledge bias after multiple iterations. Furthermore, we enforce knowledge validity, ensuring that the collected knowledge contributes to solving the q_0. This allows the system to remain focused on the q_0 throughout multiple iterations, mitigating noise interference. Additionally, to avoid the accumulation of redundant knowledge over iterations, we perform a semantic review to assess whether the collected information is already present in N_{Opt}.

Adaptive Retrieval Decision An intuition is that retrieving relevant information from a corpus has an inherent upper bound. Moreover, we observe that the model, limited by its ability to follow instructions, does not always accumulate knowledge effectively and may occasionally introduce noise. Therefore, we focus on two key aspects in this stage. First, we determine whether to employ the next retrieval iteration by assessing whether the note updating leads to knowledge gain, achieving the adaptive retrieval process. Second, we identify the best note so far to improve retrieval decision, new query generation, and note update in the next iteration τ_{t+1}. Specifically, we first guide the LLM to carefully review the content of the updated note N_t and the best note so far N_{Opt}, then assess their knowledge to get a status value V_t:

$$\begin{gathered} V_t \sim \text{LLM}_{\text{ARD}}(\text{Instruct}_{\text{ARD}}, q_0 \| N_{\text{Opt}}), \\ V_t \in \{\texttt{True}, \texttt{False}\} \end{gathered} \quad (4)$$

where the LLM_{ARD} and the $\text{Instruct}_{\text{ARD}}$ refer to the backbone model and the prompt template in the assessment process. In the assessment workflow, we have also designed multi-dimensional evaluation criteria, including 1) whether the content contains key information directly related to q_0, 2) whether the content has multiple aspects and sufficient details, and 3) whether the content is practical enough. Next, we adopt V_t to determine whether to update the best note N_{Opt}. If $V_t = \texttt{True}$, the updated note N_t generated in the current iteration τ_t is designated as the best note N_{Opt}. If $V_t = \texttt{False}$, the content of the best note N_{Opt} remains unchanged.

3.3 Note-Informed Answer Generation

Adaptive Stop Condition If the LLM determines that an updated note N_t is inferior to the best note N_{Opt}, the update is considered unsuccessful. Such a failed update indicates that the exploration has not contributed new knowledge and suggests low marginal returns from further retrieval. Based on this, we define two stopping criteria for adaptive retrieval. First, we set a threshold for the number of failure updates, termed "max failure"; once this limit is reached, the iteration terminates. Second, we impose a maximum number of iterations, termed "max step".

Task-Oriented Generation After terminating the iteration τ_t, we input the N_{Opt} from the final iteration along with the q_0 into the LLM to generate the final answer. Due to the varying output styles of different question-answering tasks, we have customized generation instructions for each task (more details in Appendix B.1).

$$\alpha \sim \text{LLM}_{\text{Ans}}(\text{Instruct}_{\text{Ans}}, q_0 \| N_{\text{Opt}}) \quad (5)$$

In Equation (5), $\text{Instruct}_{\text{Ans}}$ denotes the prompt template set of the task-oriented generation process, which includes a series of task-oriented instructions, and LLM_{Ans} indicates the backbone model in task-oriented generation stage.

3.4 Data Construction for Training

Previous studies have found that using state-of-the-art LLMs for automated sample annotation has high human correspondence (Liu et al., 2023; Fu et al., 2024). Therefore, we employ GPT-4o-mini for automated annotation for DPO training. We developed an automated data construction pipeline and carefully curated a small but high-quality training dataset for multi-task training, named **DNAlign**. This dataset $\mathcal{D}$ stems from four key task stages, including note initialization data $\mathcal{D}_{\text{Init}}$, query refinement data $\mathcal{D}_{\text{QR}}$, knowledge accumulation data $\mathcal{D}_{\text{KA}}$, and task-oriented generation data $\mathcal{D}_{\text{Ans}}$, which can be formulated as $\{x, y^+, y^-\} \sim \mathcal{D} = \langle \mathcal{D}_{\text{Init}}, \mathcal{D}_{\text{QR}}, \mathcal{D}_{\text{KA}}, \mathcal{D}_{\text{Ans}} \rangle$. We provide a detailed description of the construction process and the statistics of DNAlign in Appendix D.

3.5 Preference Optimization through DPO

To enhance the instruction-following ability of the models used in each stage of DeepNote and align with higher-quality response preferences, we employ DPO to train the backbone models used in multiple stages, marked as M_{DN}. The training data comes from DNAlign.

$$\mathcal{L}_{DPO}(M^{\theta}_{\text{DN}}; M^{ref}_{\text{DN}}) = -\mathbb{E}_{\{x,y^+,y^-\}\sim\mathcal{D}}[log\sigma \\ [\beta log\frac{M^{\theta}_{\text{DN}}(y^+|x)}{M^{ref}_{\text{DN}}(y^+|x)} - \beta log\frac{M^{\theta}_{\text{DN}}(y^-|x)}{M^{ref}_{\text{DN}}(y^-|x)}]] \tag{6}$$

Equation (6) defines the training objective, where M^{θ}_{DN} and M^{ref}_{DN} represent trained model and reference model frozen during training.

4 Experimental Setup

In this section, we detail the experimental settings and summarize them in Appendix C.

4.1 Datasets & Metrics & Corpora

Multi-hop QA task includes three challenging datasets: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (2WikiMQA) (Ho et al., 2020), and MusiQue (Trivedi et al., 2022). They require the RAG system to retrieve multi-hop knowledge and provide accurate answers through multi-hop reasoning. For the evaluation data and retrieval corpus, we use the versions released by Trivedi et al. (2023). For evaluation metrics, we follow Jiang et al. in using F1-Score (f1) and Exact Match (em). Moreover, we also add Accuracy (acc.), a common metric for QA systems evaluation (Vu and Moschitti, 2020).

Long-form QA task requires the system to gather diverse information and generate comprehensive answers. We select the ASQA (Stelmakh et al., 2022) dataset to evaluate the system's ability to explore a wide range of relevant knowledge in response to the vague original question. Specifically, we use the ASQA dataset with 948 queries recompiled by ALCE (Gao et al., 2023a) for evaluation and apply ALCE's official evaluation metrics, involving String Exact Match (str-em) and String Hit Rate (str-hit).

Short-form QA task aims to gather factual and commonsense information to produce brief responses, with relatively simple retrieval and reasoning requirements. We select StrategyQA (Geva et al., 2021) to evaluate the system's performance and robustness on simpler tasks. It requires the system to retrieve commonsense details and output a `Yes/No` answer. We follow the test set from previous work (Srivastava et al., 2023), randomly sampling 500 samples for evaluation, with accuracy (acc.) as the evaluation metric.

4.2 Baselines & LLMs

We extensively compare five types of baselines: 1) LLMs without Retrieval, which directly feeds queries into LLMs to output answers; 2) Vanilla RAG (Vanilla), which employs one-time retrieval and directly inputs the retrieved passages along with the query to generate an answer; 3) Single-Step RAG (SSRAG), which involves additional processing of the retrieved knowledge, such as summarization, based on Vanilla RAG; 4) Multi-Step RAG (MSRAG), which employs multiple retrievals; 5) Adaptive RAG (ARAG), which leverages an adaptive forward exploration strategy to retrieve knowledge to enhance answer quality. For SSRAG, we use Vanilla RAG, Chain-of-note (CoN) as counterparties. For MSRAG, we select RAT for comparison. For ARAG, we select three recent mainstream methods for comparison, including FLARE, Self-RAG, and ReAct. Additionally, we conduct experiments on a series of LLMs, including GPT-4o (Hurst et al., 2024) (OpenAI `gpt-4o-mini-0718`), Qwen2.5-7b (Yang et al., 2024), Llama3.1-70B-Instruct and Llama3.1-8B (Dubey et al., 2024).

4.3 Retrievers

We conduct experiments on all multi-hop datasets using two types of retrievers: BM25, implemented in Elasticsearch as the sparse retriever, and bge-

Methods & LLMs	Multi-hop												Long-form			Short-form
	HotpotQA				2WikiMQA				MusiQue				ASQA			StrategyQA
	acc.	f1	em	avg.	acc.	f1	em	avg.	acc.	f1	em	avg.	str-em	str-hit	avg.	acc.
LLMs without Retrieval																
Qwen2.5-7B-Instruct	19.2	25.7	18.2	21.0	25.0	29.0	24.2	26.1	2.8	9.8	2.4	5.0	24.9	8.3	12.7	67.2
Llama3.1-8B-Instruct	22.6	27.7	22.0	24.1	29.2	32.5	28.2	30.0	3.2	9.2	3.2	5.2	32.4	10.2	15.9	69.2
GPT-4o-mini	31.8	39.3	29.8	33.6	30.6	33.9	27.2	30.6	7.8	16.0	5.8	9.9	34.1	9.4	17.8	73.8
Llama3.1-70B-Instruct	32.2	40.9	30.8	34.6	34.8	38.0	31.4	34.7	7.4	13.0	5.6	8.7	41.4	14.4	21.5	75.2
Vanilla RAG (Vanilla)																
Qwen2.5-7B-Instruct	37.4	44.0	33.6	38.3	33.2	36.3	31.8	33.8	7.6	12.5	5.6	8.6	42.1	15.9	22.2	68.4
Llama3.1-8B-Instruct	37.6	46.4	35.0	39.7	33.4	36.3	32.0	33.9	6.8	12.1	6.0	8.3	39.3	13.3	20.3	71.4
GPT-4o-mini	44.0	52.2	40.0	45.4	40.4	44.4	39.2	41.3	10.6	17.3	7.6	11.8	44.3	17.5	24.5	71.2
Llama3.1-70B-Instruct	44.6	53.6	42.2	46.8	45.2	47.0	42.8	45.0	11.6	17.5	9.2	12.8	42.0	15.3	23.4	73.8
Baselines with GPT-4o-mini																
FLARE (Jiang et al., 2023)	45.8	52.9	39.2	46.0	54.8	53.6	42.4	50.3	18.6	24.9	15.6	19.7	36.8	9.9	23.4	70.0
Self-RAG (Asai et al., 2024)	43.8	53.0	41.8	46.2	35.8	40.4	33.6	36.6	11.6	19.7	10.2	13.8	42.6	16.7	24.4	68.4
CoN (Yu et al., 2023)	50.2	56.8	42.6	49.9	53.8	53.0	42.8	49.9	18.6	26.1	14.4	19.7	32.8	6.9	19.8	75.2
RAT (Wang et al., 2024)	52.0	58.3	43.6	51.3	50.8	60.0	40.0	50.3	25.2	33.5	21.0	26.6	35.7	11.4	23.6	60.2
ReAct (Yao et al., 2023)	56.0	56.8	40.4	51.1	63.6	52.6	35.6	50.6	27.0	29.3	16.6	24.3	39.4	15.1	27.3	72.0
DeepNote (Ours)																
DeepNote $_{\text{Qwen2.5-7B-Instruct}}$	50.6	59.2	48.0	52.6	50.0	51.4	41.8	47.7	14.6	19.8	11.6	15.3	44.4	19.4	26.4	71.6
+DPO $_{\text{Qwen2.5-7B-Instruct}}$	49.0	58.1	46.6	51.2	55.4	55.7	44.6	51.9	15.4	21.9	11.4	16.2	47.2	21.7	28.4	72.8
DeepNote $_{\text{Llama3.1-8B-Instruct}}$	48.0	54.3	41.2	47.8	58.0	58.2	48.2	54.8	17.0	21.3	13.2	17.2	43.4	17.9	26.2	70.8
+DPO $_{\text{Llama3.1-8B-Instruct}}$	54.6	58.9	44.0	52.5	63.8	60.5	47.4	57.2	24.4	27.3	14.4	22.0	46.4	19.8	29.4	74.2
DeepNote $_{\text{GPT-4o-mini}}$	56.8	64.3	50.2	57.1	66.2	63.7	52.6	60.8	24.8	31.3	18.4	24.8	**48.6**	**23.1**	**32.2**	**76.4**
DeepNote $_{\text{Llama3.1-70B-Instruct}}$	**59.2**	**67.2**	**54.2**	**60.2**	**72.4**	**67.1**	**55.8**	**65.1**	**32.6**	**35.0**	**23.0**	**30.2**	44.2	16.6	30.3	75.4
$\Delta_{\text{DeepNote}\rightarrow\text{Vanilla}}$	14.6↑	13.6↑	12.0↑	13.4↑	27.2↑	20.1↑	13.0↑	20.1↑	21.0↑	17.5↑	13.8↑	17.4↑	4.3↑	5.6↑	7.6↑	5.2↑

Table 1: **Results (%) of overall performance.** "**Bold**" denotes the highest value. Meanwhile, the symbol "↑" indicates the increase in our highest value compared to the Vanilla baseline under the same backbone model setting.

base-en-v1.5 as the dense retriever. For ASQA and StrategyQA, we employ the dense retriever GTR-XXL (Ni et al., 2022) following Gao et al., and we use the corpus provided by ALCE. In addition, we evaluate the performance of our framework under various top-k settings, top-$k \in \{3, 5, 7\}$, with a default of 5 (more results in Appendix A.4).

4.4 Implementation Details

Our method conducts all inference and data construction under a zero-shot setting, and we align the prompts for generation within the same dataset (cf. Appendix B). In practice, we utilize the vLLM (Kwon et al., 2023) inference acceleration tool to speed up the inference of local open-source models. Since our approach involves an adaptive iterative process, we also employ various iteration halt condition recipes to conduct a thorough analysis of our framework's performance and robustness (cf. Appendix A.2). During DPO training, we perform full parameter fine-tuning on 8×A100 GPUs, using a batch size of 8, a learning rate of 5e-7, and β set to 0.1, training the model for one epoch.

5 Results and Analysis

5.1 Overall Performance

The overall performance of DeepNote in three types of QA tasks is shown in Table 1.

Vanilla RAG struggles to meet complex retrieval demands, while DeepNote shows significant improvement in complex QA tasks. As shown in Table 1, we observe that Vanilla RAG performs well on relatively simple short-generation tasks but shows poor performance on complex multi-hop QA, highlighting that simple one-time retrieval fails to meet the demands of complex retrieval and reasoning. In contrast, DeepNote demonstrates significant performance improvements over Vanilla RAG on all datasets, regardless of whether using industry-leading closed-source models or small-size parameter open-source models. Our framework achieves a notable improvement by up to 20.1%, which confirms the effectiveness and importance of the deep exploration of our framework.

Even with information refinement, the single-step RAG remains limited by the knowledge boundary due to the one-time retrieval. DeepNote significantly outperforms the SSRAG method, CoN, on all complex QA tasks, while also showing performance advantages on simple short-form QA tasks. This trend indicates that although CoN summarizes retrieved documents to reduce noise, it still has a knowledge boundary. Furthermore, we find that the performance of CoN decreases significantly on long-form tasks compared to other tasks. This suggests that note-centric adaptive exploration fosters more effective and stable knowledge growth

Methods	HotpotQA				2WikiMQA				MusiQue				ASQA			StrategyQA
	acc.	f1	em	avg.	acc.	f1	em	avg.	acc.	f1	em	avg.	str-em	str-hit	avg.	acc.
GPT-4o-mini																
DeepNote	56.8	64.3	50.2	57.1	66.2	63.7	52.6	60.8	24.8	31.3	18.4	24.8	48.6	23.1	32.2	76.4
w/o Adap. Retrieval	47.0	54.6	41.4	47.7	46.2	48.8	43.4	46.1	14.2	20.8	10.8	15.3	47.1	21.0	27.8	74.8
w/o Adap. Retrieval & Init. Note	44.0	52.2	40.0	45.4	40.4	44.4	39.2	41.3	10.6	17.3	7.6	11.8	44.3	17.5	24.5	71.2
Llama3.1-70B-Instruct																
DeepNote	59.2	67.2	54.2	60.2	72.4	67.1	55.8	65.1	32.6	35.0	23.0	30.2	44.2	16.6	30.3	75.4
w/o Adap. Retrieval	42.6	51.0	39.8	44.5	38.8	40.0	36.6	38.5	12.6	16.3	10.4	13.1	42.4	15.5	23.7	73.8
w/o Adap. Retrieval & Init. Note	44.6	53.6	42.2	46.8	45.2	47.0	42.8	45.0	11.6	17.5	9.2	12.8	42.0	15.3	23.4	73.8

Table 2: **Results (%) of the ablation study.** The "w/o Adap. Retrieval" denotes that DeepNote employs only the initial note without adaptive retrieval; the "w/o Adap. Retrieval & Init. Note" means DeepNote employs neither adaptive retrieval nor initial note, which degenerates into Vanilla RAG. The "avg." denotes the arithmetic mean. "Blue", "light purple" and "dark purple" represent the highest, second highest, and lowest values.

than CoN while avoiding knowledge loss.

DeepNote enables more effective and robust knowledge exploration and accumulation. Compared to the MSRAG and ARAG, DeepNote shows great performance advantages across all QA tasks, demonstrating its superiority and generalization. We provide an in-depth analysis of the reasons behind this advantage. First, multi-step RAG (i.e. RAT) often introduces noise due to indiscriminate retrieval (Asai et al., 2024). On the other hand, ARAG relies on limited retrieval data or previously generated segments to determine the next retrieval strategies. The difference is that we use a note-centric approach to continuously accumulate knowledge from the perspective of information growth while avoiding noise during the adaptive iteration process. The best note is used to make the next retrieval decision. This enables the system to ensure knowledge growth during exploration and make more effective and robust retrieval decisions based on the best knowledge.

DPO effectively improves the model's ability to follow instructions in multi-stage tasks, leading to further performance gains of our framework. We find that DPO significantly improves the overall performance of DeepNote in most cases. Specifically, DPO improves the in-domain performance of our DeepNote by up to 4.2%. This improvement also generalized to more challenging out-of-domain multi-hop QA data (i.e., MusiQue) and other types of out-of-domain tasks (i.e., long-form and short-form QA tasks), with an improvement of up to 4.8%. Importantly, we achieve broad performance improvements by training on data from a single dataset, 2WikiMQA. These results validate the effectiveness and generalization of our automated data construction pipeline, DNAlign training data, and multi-task training strategy.

5.2 Ablation Study

In the ablation study, we validate the effectiveness of the note-centric adaptive retrieval process and note initialization. Table 2 presents the main results of our ablation experiments, with additional results provided in Appendix A.1.

We find that DeepNote significantly outperforms "w/o Adap. Retrieval", particularly on multi-hop datasets where the performance gap is more pronounced. These results validate the effectiveness of our note-centric adaptive retrieval process, which enables stable knowledge accumulation. Notably, since the adaptive process is intrinsically built on notes, the initialization note and adaptive retrieval are interdependent. Therefore, we further compare DeepNote with "w/o Adap. Retrieval & Init. Note", which reveals that the initial note generally achieves superior performance over Vanilla RAG in most cases, though occasional performance degradation occurs. This suggests that the initial note is effective, but its performance can be unstable due to the inherent one-time summarization and refinement of information.

5.3 Analysis

Knowledge Density and Performance Analysis We conduct an in-depth analysis of how different processes in our framework affect the density of collected knowledge. In Figure 3, we refer to the retrieved documents or notes used in the final answers by Vanilla, initial note alone, and DeepNote as "References". The portions of the "Reference" relevant to answering the original query are termed "Evidence". Specifically, we employ the model used in the answer generation stage to identify the "Evidence". Based on this, we also calculate the proportion of "Evidence" token length within the "Reference", referred to as knowledge density. We

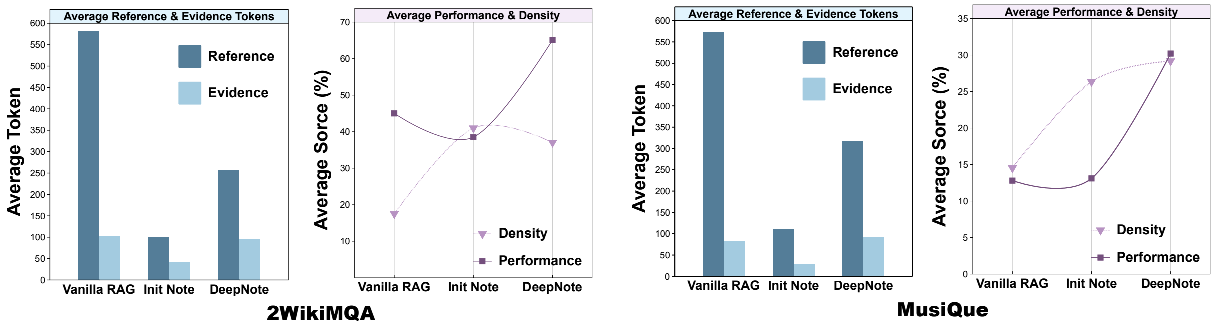

Figure 3: **Knowledge Density Comparision on Llama3.1-70B-Instruct.** The "Init Note" means that the initial note. We calculated the arithmetic mean of token length, density, and performance.

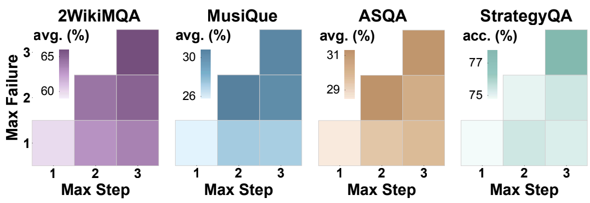

Figure 4: **Performance on different adaptive hyper-parameters** with Llama3.1-70B-Instruct.

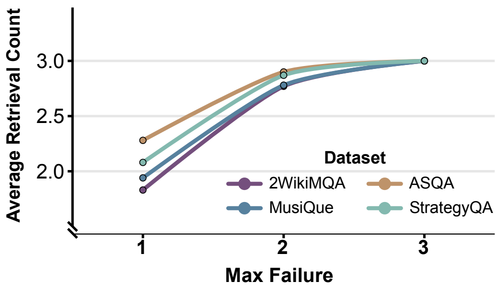

Figure 5: **Retrieval efficiency on different adaptive hyper-parameters** with Llama3.1-70B-Instruct.

find that the references in Vanilla are very lengthy but have low knowledge density, indicating significant noise in these references. The initial note improves knowledge density by summarizing and refining the information retrieved in a single pass. However, this increase in density is mainly due to the sharp reduction in the total token length of the references. In Figure 3 and Table 2, we find that the initial note refines knowledge and reduces noise, thereby enhancing performance in most cases, although instability may arise due to the reduced total knowledge volume. In contrast, our framework achieves a knowledge density comparable to the initial note and significantly higher than Vanilla, while showing substantial performance improvement. This suggests that note-centric adaptive retrieval can gather more comprehensive, refined, and accurate knowledge while minimizing noise.

Efficiency and Performance Trade-off Using DeepNote, researchers can adjust the failure update threshold and total iteration threshold to control exploration depth. In Figures 4 and 5, we investigate the impact of the adaptive stop threshold on both performance and retrieval counts. Figure 4 suggests that performance improves as the total iteration threshold increases, while the maximum update failure threshold remains constant. This improvement arises from relaxing the total iteration constraint, which facilitates deeper exploration through additional retrieval attempts. Conversely, when the total iteration threshold is fixed, increasing the update failure threshold also enhances performance by allowing greater tolerance for errors during exploration. Notably, competitive performance is achieved when the two thresholds are set to similar values. In Figure 5, we further show the total number of retrievals used during the adaptive retrieval process (excluding the retrievals in the note initialization). We find that increasing the threshold requires more retrieval counts, accompanied by diminishing marginal returns. Therefore, when balancing retrieval efficiency and performance, it is advisable to choose a moderate or lower failure threshold and set the total iteration threshold slightly higher than it.

6 Conclusion

In this work, we identify two limitations in the existing studies and develop a novel ARAG framework–**DeepNote**. DeepNote uses notes as knowledge carriers for stable knowledge growth and devises optimal retrieval strategies based on the best available knowledge. Extensive empirical experiments, ablation studies, and multi-dimensional analyses confirm the superiority of DeepNote across various question-answering tasks and its flexibility in balancing retrieval efficiency and performance.

7 Limitations

Experiments demonstrate that DeepNote significantly advances RAG systems in tackling complex problems through robust and superior deep knowledge exploration and continuous information accumulation. However, certain limitations still warrant attention. First, this work focuses on single-source retrieval; future efforts should explore dynamic knowledge integration in multi-source settings. Second, existing datasets prioritize early-stage exploration gains, leaving the performance of DeepNote in long-chain tasks unexplored. Building long-chain datasets could better align models with high-quality responses in later iterations.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. 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*. OpenReview.net.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *CoRR*, abs/2306.04136.

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

Yuyan Chen, Qiang Fu, Yichen Yuan, Zhihao Wen, Ge Fan, Dayiheng Liu, Dongmei Zhang, Zhixu Li, and Yanghua Xiao. 2023. Hallucination detection: Robustly discerning reliable answers in large language models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 245–255. ACM.

Maria Angels de Luis Balaguer, Vinamra Benara, Renato Luiz de Freitas Cunha, Roberto de M. Estevão Filho, Todd Hendry, Daniel Holstein, Jennifer Marsman, Nick Mecklenburg, Sara Malvar, Leonardo O. Nunes, Rafael Padilha, Morris Sharp, Bruno Silva, Swati Sharma, Vijay Aski, and Ranveer Chandra. 2024. RAG vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture. *CoRR*, abs/2401.08406.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. Gptscore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 6556–6576. Association for Computational Linguistics.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023a. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6465–6488. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Trans. Assoc. Comput. Linguistics*, 9:346–361.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Hangfeng He, Hongming Zhang, and Dan Roth. 2023. Rethinking with retrieval: Faithful large language model inference. *CoRR*, abs/2301.00303.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll L. Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, and Dane Sherburn. 2024. Gpt-4o system card. *CoRR*, abs/2410.21276.

Gautier Izacard, Patrick S. H. Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 7036–7050. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7969–7992. Association for Computational Linguistics.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Zixuan Ke, Weize Kong, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Bridging the preference gap between retrievers and llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10438–10451. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.

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

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. RA-DIT: retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2511–2522. Association for Computational Linguistics.

Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. 2024. CRUD-RAG: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *CoRR*, abs/2401.17043.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12076–12100. Association for Computational Linguistics.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 9844–9855. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Trans. Assoc. Comput. Linguistics*, 11:1316–1331.

Thomas R. Shultz, Jamie Wise, and Ardavan Salehi Nobandegani. 2024. GPT-4 understands discourse at least as well as humans do. *CoRR*, abs/2403.17196.

Shamane Siriwardhana, Rivindu Weerasekera, Tharindu Kaluarachchi, Elliott Wen, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *Trans. Assoc. Comput. Linguistics*, 11:1–17.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew K. Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakas, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartlomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, Cèsar Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodolà, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan J. Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, François Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocon, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse H. Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, José Hernández-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory W. Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Senel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, María José Ramírez-Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael I. Ivanitskiy, Michael Starritt, Michael Strube, Michal Swedrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T., Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Milkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima (Shammie) Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay V. Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Trans. Mach. Learn. Res.*, 2023.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 8273–8288. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. 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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics.

Ellen M. Voorhees. 1999. The TREC-8 question answering track report. In *Proceedings of The Eighth Text REtrieval Conference, TREC 1999, Gaithersburg, Maryland, USA, November 17-19, 1999*, volume 500-246 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Thuy Vu and Alessandro Moschitti. 2020. Ava: an automatic evaluation approach to question answering systems. *arXiv preprint arXiv:2005.00705*.

Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. 2024. RAT: retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *CoRR*, abs/2403.05313.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Wenpeng Yin, Qinyuan Ye, Pengfei Liu, Xiang Ren, and Hinrich Schütze. 2023a. Llm-driven instruction following: Progresses and concerns. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 19–25.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023b. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8653–8665. Association for Computational Linguistics.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *CoRR*, abs/2311.09210.

Guido Zuccon, Bevan Koopman, and Razia Shaik. 2023. Chatgpt hallucinates when attributing answers. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP 2023, Beijing, China, November 26-28, 2023*, pages 46–51. ACM.

Appendix

A Additional Experimental Results

A.1 Ablation Study

Table 5 presents more ablation results across all models and datasets. We observe that on complex QA datasets (including multi-hop and long-form QA tasks), the performance with adaptive retrieval significantly surpasses that without adaptive retrieval, confirming the effectiveness of our note-centric adaptive retrieval. However, on the simpler StrategyQA dataset, the advantage diminishes, as straightforward reasoning tasks inherently require less retrieval.

A.2 Adaptive Hyper-Parameter Analysis

In Table 4, we present the impact of different hyper-parameters on DeepNote's performance across all datasets and models. We employ six sets of hyper-parameters, $\{\text{max step}, \text{max failure}\} = \{(1,1),(2,1),(2,2),(3,1),(3,2),(3,3)\}$. It is worth mentioning that the max failure value cannot exceed the max step value, as having failure updates exceed the total iteration threshold would render the max failure meaningless. In Table 4, we observe conclusions similar to those in Figure 4. Increasing either max failure or max step can encourage the model to potentially perform deeper retrieval. Comparing the results of the $(2,2)$ and $(3,1)$ hyper-parameter sets, we find that $(2,2)$ often outperforms $(3,1)$ as reaching the max failure limit terminates the iteration, rendering an excessively high max step ineffective. Therefore, we recommend researchers use values for max failure and max step that are close to each other when running DeepNote.

Additionally, we find that models trained with DPO tend to achieve higher performance with smaller hyper-parameter settings. This is partly because the initial iteration of deep exploration typically yields the highest returns, with diminishing marginal gains as exploration continues. Furthermore, since our training data is derived from τ_0 and τ_1, the model effectively learns how to better explore the knowledge base in the early stages.

A.3 Knowledge Density Analysis

Figure 6 presents additional results on knowledge density analysis. The trends and conclusions are consistent with those in Figure 3.

A.4 Impact of Different Top-k Values and Retrievers

The top-k and retriever settings significantly impact the overall performance of RAG systems. In Table 1, we have already presented the main results of DeepNote based on the top-5 settings and the BM25-based retriever. Here, we further investigate the performance of DeepNote under different top-k settings and evaluate its performance on two mainstream types of retrievers.

Table 6 and 7 present the performance of DeepNote with different top-k settings. The results show that on complex datasets, using a higher top-k (i.e., top-7) leads to better performance. On relatively simple commonsense QA datasets, top-5 achieves the best results. This indicates that complex datasets have higher and more intricate retrieval demands. Additionally, across various top-k settings, DeepNote significantly outperforms Vanilla RAG, demonstrating its robustness.

For different retrievers, the results in Table 8 reveal that using dense retrievers achieves higher performance. Overall, DeepNote's performance is similar using both types of retrievers, confirming the robustness of our framework.

B Prompt Details

In this section, we present all the prompts used in our framework.

B.1 Prompts for Inference

For prompt in the inference stage, we present the prompts used in all three key processes: note initialization (Table 10), note-centric adaptive retrieval, and note-informed answer generation. The note-centric adaptive retrieval process consists of multiple stages, including the Query Refinement Stage (Table 11), Knowledge Accumulation Stage (Table 12), and Adaptive Retrieval Decision Stage (Table 13). In addition, due to the varied output style (e.g., long- or short-form generations) of different QA tasks, we tailor the prompts to be task-oriented. For example, multi-hop QA tasks require short and precise outputs, often only a few words, while the knowledge in the best note appears as a long text. Therefore, we guide the LLM to output only key answers without including extraneous words (Table 14). For the long-form QA task, we guide the response style instead of stringent limitations (Table 15). Additionally, since StrategyQA requires the system to provide binary answers (Yes/No), our prompt instructs the model to output only Yes or No as the response (Table 16).

B.2 Prompts for DPO

Only constructing Note Initialization Data (Table 17) and Query Refinement Data (Table 18) require additional prompts. In building Knowledge Accumulation Data, we directly use the $\text{Instruct}_{\text{ARD}}$ from the inference process to determine whether knowledge has increased and construct positive-negative pairs based on this judgment. In building Task-Oriented Generation Data, we use the same prompt as in the inference process and employ task evaluation metrics as supervision signals to select positive-negative pairs.

C Experimental Setup Details

C.1 More Implementation Details

In detail, we reproduce Self-RAG and ReAct via the langchain framework[2]. During the inference stage, we use a temperature value of 0.1. In the data construction phase, we primarily adjust two parameters: temperature and top_p. By combining them pairwise, we use nine parameter sets to construct the training data, temperature $\in \{0.1, 0.5, 0.9\}$ and top_p $\in \{0.1, 0.5, 0.9\}$. Plus, We summarize all experimental settings in Table 9.

[2] https://github.com/langchain-ai

	$\mathcal{D}_{\text{Init}}$	$\mathcal{D}_{\text{QR}}$	$\mathcal{D}_{\text{KA}}$	$\mathcal{D}_{\text{Ans}}$	$\mathcal{D}$
# Sample	1900	1900	1900	300	6000

Table 3: **Statistics of DNAlign Datasets for DPO.**

D Details of Training Dataset Construction

We randomly sampled 15000 samples from the train set of the 2WikiMQA dataset to construct our DNAlign dataset. We present the statistics of DNAlign in Table 3.

D.1 Note Initialization Data

For each sampled instance, we used the original query q_0, the retrieved document $P_{k,0}$, and the prompt template $\text{Instruct}_{\text{Init}}$ to form the input x_{Init}, which was fed into the LLM for the note initialization inference process. To improve the diversity of responses, we configured nine parameter settings (detailed in Appendix C.1) during inference. It is worth mentioning that we also use multiple top-k values to simulate diverse retrieval scenarios in real-world settings. After inference, we employed GPT-4o-mini as the evaluation model to select the positive example y_{Init}^{+} and negative example y_{Init}^{-} from the nine generated initial notes. We filtered out instances that lacked either a positive or a negative example. Finally, the constructed training data for the note initialization process is denoted as $\{x_{\text{Init}}, y_{\text{Init}}^{+}, y_{\text{Init}}^{-}\} \sim \mathcal{D}_{\text{Init}}$.

D.2 Query Refinement Data

We perform inference with the same parameter settings, top-k strategy, and apply the same filtering approach. Notably, this stage requires using the generated output from the initialization note as input, meaning the quality of the initial note affects the quality of the training data at this stage. Based on this, we construct the input x_{QR} using y_{Init}^{+}, q_0, and the prompt template $\text{Instruct}_{\text{QR}}$. We then employ GPT-4o-mini to select positive examples y_{QR}^{+} and negative examples y_{QR}^{-}, forming the dataset $\left\{x_{\text{QR}}, y_{\text{QR}}^{+}, y_{\text{QR}}^{-}\right\} \sim \mathcal{D}_{\text{QR}}$.

D.3 Knowledge Accumulation Data

At this stage, the data enhances the model's ability to update notes and maximize knowledge accumulation. We maintain the same inference parameters, top-k strategy, and filtering strategies. We retrieve the top-k documents, $P_{k,1}$, using the new query labeled y_{QR}^{+}. Next, we use y_{Init}^{+}, q_0, and $P_{k,1}$ as the input. We directly apply the evaluation strategy from the adaptive retrieval decision stage to generate positive and negative labels. We then randomly select one positive and one negative example from the respective sets as the final positive and negative samples. The final dataset is denoted as $\{x_{\text{KA}}, y_{\text{KA}}^{+}, y_{\text{KA}}^{-}\} \sim \mathcal{D}_{\text{KA}}$.

D.4 Task-Oriented Generation Data

After obtaining a high-quality note, we aim to align the system's response style for specific tasks. We employ the inference process of Vanilla RAG to generate answers and use the task evaluation metric to identify positive and negative examples. We apply the same parameters, top-k strategy, and positive-negative pairs selection strategy as in the knowledge accumulation stage. The dataset can be formulated as: $\{x_{\text{Ans}}, y_{\text{Ans}}^{+}, y_{\text{Ans}}^{-}\} \sim \mathcal{D}_{\text{Ans}}$.

E Case Study

In Tables 19 and 20, we present examples of DeepNote and conduct a case study. Given the query "*Where was the place of death of Anna Of Pomerania's father?*", Vanilla RAG and Self-RAG failed to explore effective information and outputted the response "No information." DeepNote, after the second note update, identified the key information about her father. Following the third update, it not only located his place of death but also found the time of her father's death within the same paragraph, ultimately outputting the correct information: "*Stettin*." Importantly, we observe that our answer not only includes the correct response but also expands on closely related knowledge: "*Stettin (also known as Szczecin in Polish)*". This demonstrates DeepNote's superior knowledge integration capability and the ability to maintain logical coherence during the integration process.

Additionally, Table 21 presents a highly challenging question, i.e., "*A man who played in the 1986 FIFA world cup played for what team during the 1982 Scottish League Cup Final?*". This case illustrates that errors are mainly due to the inability to retrieve relevant information.

LLMs	Max Step	Max Failure	HotpotQA				2WikiMQA				MusiQue				ASQA			StrategyQA
			acc.	f1	em	avg.	acc.	f1	em	avg.	acc.	f1	em	avg.	str-em	str-hit	avg.	acc
Qwen2.5-7B-Instruct	1	1	47.2	56.2	44.4	49.3	45.4	47.4	39.8	44.2	12.2	17.5	9.8	13.2	44.5	19.7	25.8	72.2
	2	1	46.2	54.6	42.8	47.9	47.2	48.7	39.8	45.2	12.8	17.4	9.8	13.3	44.6	19.7	25.9	69.4
	2	2	49.0	57.3	44.8	50.4	48.8	50.0	40.8	46.5	12.2	17.7	9.8	13.2	44.8	19.3	25.8	71.2
	3	1	46.8	55.5	43.6	48.6	45.8	47.6	38.8	44.1	11.8	16.1	8.6	12.2	44.2	19.5	25.3	71.6
	3	2	50.6	59.2	48.0	52.6	50.0	51.4	41.8	47.7	14.6	19.8	11.6	15.3	44.4	19.4	26.4	71.6
	3	3	48.2	57.5	45.6	50.4	51.2	52.0	42.2	48.5	14.6	19.8	11.8	15.4	44.5	19.8	26.6	72.0
Qwen2.5-7B-Instruct+DPO	1	1	46.4	56.7	44.4	49.2	54.0	54.9	45.0	51.3	16.8	23.6	13.6	18.0	46.2	20.7	28.3	70.8
	2	1	46.4	56.8	45.0	49.4	54.4	55.1	45.4	51.6	14.0	21.9	11.6	15.8	47.1	21.2	28.0	70.2
	2	2	47.4	57.3	44.8	49.8	57.4	57.7	48.0	54.4	15.8	23.9	13.0	17.6	47.1	21.8	28.8	70.8
	3	1	47.4	57.4	44.8	49.9	53.2	54.1	43.4	50.2	13.6	21.3	10.2	15.0	47.0	21.9	28.0	70.2
	3	2	49.0	58.1	46.6	51.2	55.4	55.7	44.6	51.9	15.4	21.9	11.4	16.2	47.2	21.7	28.4	72.8
	3	3	46.2	57.3	44.6	49.4	55.2	55.6	45.0	51.9	16.6	23.4	12.6	17.5	47.4	22.7	29.2	70.2
Llama3.1-8B-Instruct	1	1	45.2	52.0	39.8	45.7	54.2	53.8	45.6	51.2	14.4	18.9	11.0	14.8	43.8	18.3	25.6	72.2
	2	1	45.8	52.8	40.8	46.5	53.4	52.9	46.0	50.8	14.8	18.9	11.8	15.2	45.0	18.9	26.4	72.0
	2	2	49.8	56.9	44.6	50.4	53.8	53.6	45.0	50.8	16.0	21.6	12.6	16.7	44.4	18.4	26.5	72.8
	3	1	47.8	54.2	42.6	48.2	54.6	53.0	45.0	50.9	15.0	19.2	11.4	15.2	44.8	19.2	26.4	73.0
	3	2	48.0	54.3	41.2	47.8	58.0	58.2	48.2	54.8	17.0	21.3	13.2	17.2	43.4	17.9	26.2	70.8
	3	3	49.6	56.6	44.8	50.3	57.2	56.3	48.0	53.8	16.2	21.4	12.2	16.6	44.6	18.9	26.7	70.2
Llama3.1-8B-Instruct+DPO	1	1	53.2	60.1	44.8	52.7	60.2	57.3	47.6	55.0	21.6	24.9	13.2	19.9	46.7	20.8	29.1	74.2
	2	1	54.2	58.1	41.2	51.2	61.8	60.0	49.6	57.1	21.8	26.4	15.2	21.1	46.3	20.4	29.3	73.8
	2	2	53.6	58.1	42.6	51.4	63.6	59.9	48.4	57.3	21.4	26.9	15.2	21.2	46.6	20.6	29.5	72.4
	3	1	54.0	58.5	42.8	51.8	64.8	61.9	50.0	58.9	25.4	27.7	15.8	23.0	46.5	19.2	29.6	73.2
	3	2	54.6	58.9	44.0	52.5	63.8	60.5	47.4	57.2	24.4	27.3	14.4	22.0	46.4	19.8	29.4	74.2
	3	3	55.6	59.4	43.0	52.7	65.6	62.3	50.0	59.3	22.4	26.5	14.4	21.1	47.1	20.2	29.5	72.2
GPT-4o-mini	1	1	56.2	63.2	49.8	56.4	60.6	59.8	50.0	56.8	22.0	28.3	16.2	22.2	48.4	22.9	31.2	75.4
	2	1	57.0	64.0	49.2	56.7	64.0	62.6	52.4	59.7	22.4	28.1	16.2	22.2	48.7	22.7	31.2	75.4
	2	2	58.0	64.9	50.0	57.6	65.8	64.3	53.0	61.0	23.4	29.6	17.2	23.4	48.7	22.4	31.5	77.4
	3	1	57.0	63.4	49.0	56.5	63.4	61.6	51.6	58.9	22.8	28.8	16.4	22.7	48.4	21.8	31.0	76.2
	3	2	56.8	64.3	50.2	57.1	66.2	63.7	52.6	60.8	24.8	31.3	18.4	24.8	48.6	23.1	32.2	76.4
	3	3	58.4	65.4	49.8	57.9	64.0	62.3	51.2	59.2	25.6	31.0	19.4	25.3	49.4	23.1	32.6	77.0
Llama3.1-70B-Instruct	1	1	55.6	63.7	50.6	56.6	65.8	61.5	52.4	59.9	27.2	30.4	19.8	25.8	43.8	15.9	28.5	74.8
	2	1	56.8	64.9	51.6	57.8	69.2	64.6	55.2	63.0	28.8	31.7	21.4	27.3	44.5	16.7	29.5	75.4
	2	2	60.2	68.4	54.4	61.0	70.6	66.1	56.2	64.3	33.0	34.9	24.4	30.8	45.1	17.9	31.3	75.0
	3	1	57.8	65.4	52.0	58.4	70.0	65.0	56.0	63.7	28.6	31.7	21.4	27.2	44.7	17.6	29.8	75.2
	3	2	59.2	67.2	54.2	60.2	72.4	67.1	55.8	65.1	32.6	35.0	23.0	30.2	44.2	16.6	30.3	75.4
	3	3	59.6	67.8	53.4	60.3	73.0	67.9	57.2	66.0	32.0	35.8	23.6	30.5	45.3	17.8	31.2	77.8

Table 4: **Results (%) of performance on different adaptive hyper-parameter analysis** of DeepNote on all LLMs and datasets. We have set a total of six sets of hyper-parameters.

Methods	HotpotQA				2WikiMQA				MusiQue				ASQA			StrategyQA
	acc.	f1	em	avg.	acc.	f1	em	avg.	acc.	f1	em	avg.	str-em	str-hit	avg.	acc.
Qwen2.5-7B-instruct																
DeepNote	50.6	59.2	48.0	52.6	50.0	51.4	41.8	47.7	14.6	19.8	11.6	15.3	44.4	19.4	26.4	71.6
w/o Adap. Retrieval	40.2	48.3	37.4	42.0	35.8	39.6	34.6	36.7	8.6	12.7	6.2	9.2	43.9	19.0	24.0	71.2
w/o Adap. Retrieval & Init. Note	37.4	44.0	33.6	38.3	33.2	36.3	31.8	33.8	7.6	12.5	5.6	8.6	42.1	15.9	22.2	68.4
Llama3.1-8B-Instruct																
DeepNote	48.0	54.3	41.2	47.8	58.0	58.2	48.2	54.8	17.0	21.3	13.2	17.2	43.4	17.9	26.2	70.8
w/o Adap. Retrieval	37.6	44.5	33.6	38.6	39.6	41.2	38.0	39.6	8.4	11.9	5.8	8.7	41.3	16.6	22.2	72.2
w/o Adap. Retrieval & Init. Note	37.6	46.4	35.0	39.7	33.4	36.3	32.0	33.9	6.8	12.1	6.0	8.3	39.3	13.3	20.3	71.4
GPT-4o-mini																
DeepNote	56.8	64.3	50.2	57.1	66.2	63.7	52.6	60.8	24.8	31.3	18.4	24.8	48.6	23.1	32.2	76.4
w/o Adap. Retrieval	47.0	54.6	41.4	47.7	46.2	48.8	43.4	46.1	14.2	20.8	10.8	15.3	47.1	21.0	27.8	74.8
w/o Adap. Retrieval & Init. Note	44.0	52.2	40.0	45.4	40.4	44.4	39.2	41.3	10.6	17.3	7.6	11.8	44.3	17.5	24.5	71.2
Llama3.1-70B-Instruct																
DeepNote	59.2	67.2	54.2	60.2	72.4	67.1	55.8	65.1	32.6	35.0	23.0	30.2	44.2	16.6	30.3	75.4
w/o Adap. Retrieval	42.6	51.0	39.8	44.5	38.8	40.0	36.6	38.5	12.6	16.3	10.4	13.1	42.4	15.5	23.7	73.8
w/o Adap. Retrieval & Init. Note	44.6	53.6	42.2	46.8	45.2	47.0	42.8	45.0	11.6	17.5	9.2	12.8	42.0	15.3	23.4	73.8

Table 5: **All results (%) of ablation study.** "Blue", "light purple" and "dark purple" represent the highest, second highest, and lowest values among the results of different top-k, respectively.

Top-k	Methods	HotpotQA				2WikiMQA				MusiQue				ASQA			StrategyQA
		acc.	f1	em	avg.	acc.	f1	em	avg.	acc.	f1	em	avg.	str-em	str-hit	avg.	acc.
Top-3	Vanilla RAG	42.8	51.0	39.0	44.3	39.2	43.0	38.2	40.1	10.2	15.9	7.2	11.1	41.8	16.5	23.1	68.8
	DeepNote	**56.6**	**64.1**	**49.6**	**56.8**	**61.4**	**61.0**	**51.0**	**57.8**	**21.6**	**27.9**	**16.2**	**21.9**	**48.1**	**21.7**	**30.6**	**73.2**
Top-5	Vanilla RAG	44.0	52.2	40.0	45.4	40.4	44.4	39.2	41.3	10.6	17.3	7.6	11.8	44.3	17.5	24.5	71.2
	DeepNote	**56.8**	**64.3**	**50.2**	**57.1**	**66.2**	**63.7**	**52.6**	**60.8**	**24.8**	**31.3**	**18.4**	**24.8**	**48.6**	**23.1**	**32.2**	**76.4**
Top-7	Vanilla RAG	45.2	54.1	41.8	47.0	39.4	43.4	38.2	40.3	10.8	16.3	7.2	11.4	45.4	18.1	25.0	69.8
	DeepNote	**59.0**	**66.0**	**51.4**	**58.8**	**65.0**	**62.8**	**50.4**	**59.4**	**24.8**	**30.4**	**19.2**	**24.8**	**50.0**	**22.4**	**32.4**	**74.2**

Table 6: **Results (%) on different Top-k.** We present the results of DeepNote using GPT-4o-mini as the backbone model. "Blue", "light purple" and "dark purple" represent the highest, second highest, and lowest values among the results of different top-k, respectively. "**Bold**" means the higher value between Vanilla RAG and DeepNote under the same top-k setting.

Top-k	Methods	HotpotQA				2WikiMQA				MusiQue				ASQA			StrategyQA
		acc.	f1	em	avg.	acc.	f1	em	avg.	acc.	f1	em	avg.	str-em	str-hit	avg.	acc.
Top-3	Vanilla RAG	42.2	50.8	40.8	44.6	42.8	44.3	40.0	42.4	11.4	17.4	8.8	12.5	40.1	13.4	22.0	73.2
	DeepNote	**56.2**	**64.1**	**51.4**	**57.2**	**65.4**	**61.6**	**53.0**	**60.0**	**30.2**	**32.7**	**22.4**	**28.4**	**43.4**	**16.7**	**29.5**	**73.8**
Top-5	Vanilla RAG	44.6	53.6	42.2	46.8	45.2	47.0	42.8	45.0	11.6	17.5	9.2	12.8	42.0	15.3	23.4	73.8
	DeepNote	**59.2**	**67.2**	**54.2**	**60.2**	**72.4**	**67.1**	**55.8**	**65.1**	**32.6**	**35.0**	**23.0**	**30.2**	**44.2**	**16.6**	**30.3**	**75.4**
Top-7	Vanilla RAG	45.4	54.7	43.4	47.8	45.8	47.9	44.0	45.9	10.4	17.5	9.0	12.3	43.6	15.8	23.9	**76.4**
	DeepNote	**59.8**	**67.5**	**55.0**	**60.8**	**75.4**	**69.9**	**58.8**	**68.0**	**31.8**	**34.6**	**23.2**	**29.9**	**46.4**	**18.0**	**31.4**	74.4

Table 7: **Results (%) on different Top-k.** We present the results of DeepNote using Llama3.1-70B-Instruct as the backbone model. "Blue", "light purple" and "dark purple" represent the highest, second highest, and lowest values among the results of different top-k, respectively. "**Bold**" means the higher value between Vanilla RAG and DeepNote under the same top-k setting.

Top-k	Retrievers	HotpotQA				2WikiMQA				MusiQue			
		acc.	f1	em	avg.	acc.	f1	em	avg.	acc.	f1	em	avg.
Top-5	BM25	59.2	67.2	54.2	60.2	72.4	67.1	55.8	65.1	32.6	35.0	23.0	30.2
	bge-base-en-v1.5	**61.6**	**68.4**	**55.0**	**61.7**	**75.8**	**69.7**	**59.6**	**68.4**	**33.4**	**37.1**	**26.0**	**32.2**

Table 8: **Results (%) of different retrievers.** We present the results of DeepNote on Llama3.1-70B-Instruct.

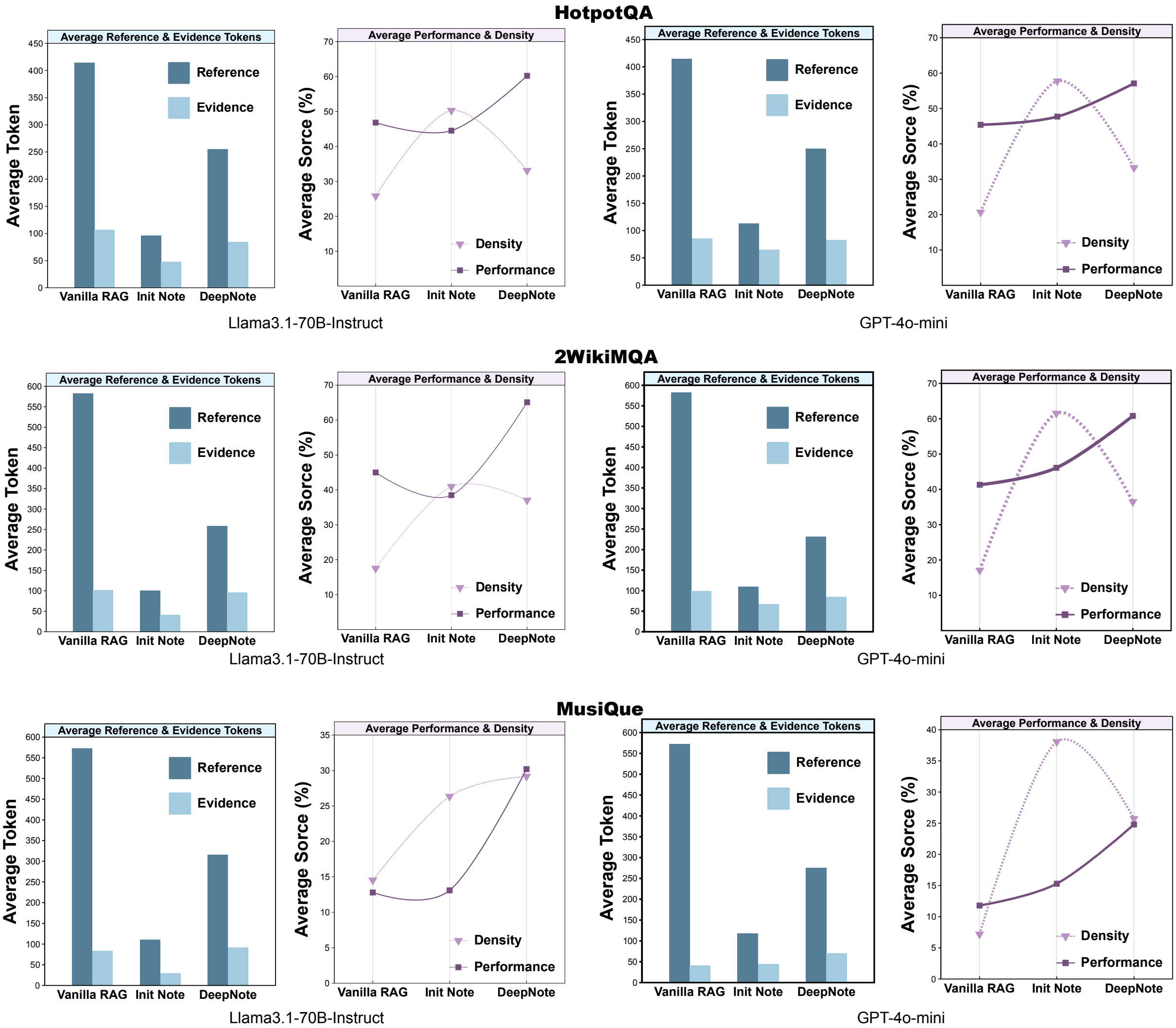

Figure 6: **All results (%) of knowledge density analysis.**

Settings	HotpotQA	2WikiMQA	MusiQue	ASQA	StrategyQA
Dataset statistics					
# Samples used for evaluation	500	500	500	948	500
Evaluation settings					
Metric	Accuracy,F1,EM	Accuracy,F1,EM	Accuracy,F1,EM	Accuracy,F1,EM	Accuracy
Retrieval settings					
Corpus	Trivedi et al., 2023	Trivedi et al., 2023	Trivedi et al., 2023	Wikipedia-2018	Wikipedia-2018
# Documents in Corpus	5233329	139416	430139	21015324	21015324
Retriever	BM25, Dense	BM25, Dense	BM25,Dense	Dense	Dense
Top-k	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7

Table 9: **All experimental settings.** We use bge-base-en-v1.5 as the dense retriever.

Prompt of the Note Initialization Process

Instructions

Based on the provided document content, write a note.

The note should integrate all relevant information from the original text that can help answer the specified question and form a coherent paragraph. Please ensure that the note includes all original text information useful for answering the question.

Question to be answered: `{query}`
Document content: `{refs}`

Please provide the note you wrote:

Table 10: **Prompt of the note initialization process.**

Prompt of the Query Refinement Stage

Instructions

Task: Based on the notes, propose two new questions.

These new questions will be used to retrieve documents to supplement the notes and help answer the original question. The new questions should be concise and include keywords that facilitate retrieval. The new questions should avoid duplication with the existing question list.

Original question: `{query}`
Notes: `{note}`

Existing question list: `{query_log}`

Please provide the note you wrote:

Table 11: **Prompt of the query refinement stage.**

Prompt of the Knowledge Accumulation Stage

Instructions

Task: Based on the retrieved documents, supplement the notes with content not yet included but useful for answering the question.

The supplement should use the original text from the retrieved documents. The added content should include as much information from the retrieved documents as possible.

Question: `{query}`
Retrieved document: `{refs}`
Notes: `{note}`

Please provide the note you wrote:

Table 12: **Prompt of the knowledge accumulation stage.**

Prompt of the Adaptive Retrieval Decision Stage

Instructions

Task: Please help me determine which note is better based on the following evaluation criteria:
1. Contains key information directly related to the question.
2. Completeness of Information: Does it cover all relevant aspects and details?
3. Level of Detail: Does it provide enough detail to understand the issue in depth?
4. Practicality: Does the note offer practical help and solutions?

Please make your judgment adhering strictly to the following rules:
- If Note 2 does not add new meaningful content on top of Note 1, or only adds redundant information, return `json {{"status":"False"}}` directly.
- If Note 2 has significant improvements over Note 1 based on the above criteria, return `json {{"status":"True"}}` directly; otherwise, return `json {{"status":"False"}}`.

Question: `{query}`
Provided Note 1: `{best_note}`
Provided Note 2: `{new_note}`

Based on the above information, make your judgment without explanation and return the result directly.

Table 13: **Prompt of the adaptive retrieval decision stage.**

Prompt of the Note-Informed Answer Generation Process (**Multi-hop QA**)

Instructions

Answer the question based on the given notes. Only give me the answer and do not output any other words.

The following are given notes:
`{note}`
Question: `{query}`

Answer:

Table 14: **Prompt of the note-informed answer generation process (multi-hop QA).**

Prompts of the Note-Informed Answer Generation Process (**ASQA**)

Instructions

Write an accurate, engaging, and concise answer for the given question using only the provided notes. Use an unbiased and journalistic tone.

Question: `{query}`
Notes: `{note}`

Table 15: **Prompt of the note-informed answer generation process (ASQA).**

Prompts of the Note-Informed Answer Generation Process (**StrategyQA**)

Instructions

Answer the question based on the given notes.
Only give me "yes" or "no" as your answer and do not output any other words.

The following are given notes: `{note}`
Question: `{query}`

Answer:

Table 16: **Prompt of the note-informed answer generation process (StrategyQA).**

Prompts of Note Initialization Stage for DPO

Instructions

Task: You will receive a list of notes generated based on a given document content and question. Your task is to evaluate and score these notes based on their quality. Quality refers to: relevance, coherence, completeness in answering the specified question, and accuracy of information.

Question to be answered: {query}
Document content: {refs}
Generated notes: {notes}
Note format: Each note contains "_id" and "content" fields.

Evaluate the generated notes. The highest-scoring note must be factually correct based on the document. If no note is correct, or if there is minimal quality difference between notes, use the same _id for both best and worst.

Output in the following JSON format: json {{"best_id": <_id of the highest-scoring note>, "worst_id": <_id of the lowest-scoring note>}}

Do not include any explanations or additional text.

Table 17: **Prompt of note initialization stage for DPO.**

Prompts of Query Refinement Stage for DPO

Instructions

Task: You will receive a list of new questions generated based on some notes and an existing question list to supplement a given original question. Your task is to evaluate these new questions based on their quality. Quality refers to: relevance, specificity, keyword richness, and non-redundancy. The goal is to identify questions that can retrieve useful information to help answer the original question.

Notes: {notes}

Original question: {query}

Existing question list: {query_log}

New question list: {new_querys}

Question format: Each question contains "_id" and "content" fields.

Evaluate the new question list. The highest-scoring new question must be able to help retrieve relevant information to answer the original question. If no new question can help get useful information, or if there is minimal quality difference between new questions, use the same `_id` for both `best_id` and `worst_id`.

Output in the following format: `json {{"best_id": <_id of the highest-scoring question>, "worst_id": <_id of the lowest-scoring question>}}`

Do not include any explanations or additional text.

Table 18: **Prompt of query refinement stage for DPO.**

Question: Where was the place of death of Anna Of Pomerania's father?
Process of DeepNote: ☞ **Init note:** Unfortunately, the provided document content does not contain any information about Anna Of Pomerania's father or his place of death. The document content appears to be unrelated to Anna Of Pomerania and instead discusses various topics such as a British silent adventure film, the concept of "sins of the father," and the definitions of "place of birth" and "place of origin." ☞ **New queries:** 1. Who was Anna Of Pomerania's father? 2. What is known about the life and death of Anna Of Pomerania's father? ☞ **Updated note:** Unfortunately, the provided document content **does not contain any information about Anna Of Pomerania's father or his place of death.** . . . Sophie of Pomerania, Duchess of Pomerania, we can learn that Sophie's father was Bogislaw IX, Duke of Pomerania, and that he died in 1446. . . . It is worth noting that the documents provided do not mention Anna Of Pomerania at all, and the information about Sophie of Pomerania and her family is not directly related to Anna Of Pomerania . . . ➜ **[Knowledge growth: False]** ☞ **New queries:** 1. What is the name of Anna Of Pomerania's father and where did he die? 2. Who was the parent of Anna Of Pomerania and what was his place of death? ☞ **Updated note:** Based on the retrieved documents, I was **able to find some information related to Anna of Pomerania's family, but not directly about her father's place of death.** . . . From the document "Anna of Pomerania", we know that **Anna of Pomerania's father was Bogislaw XIII, Duke of Pomerania**. Unfortunately, the document does not mention his place of death. . . . ➜ **[Knowledge growth: True]** ☞ **New queries:** 1. What is known about the death of Bogislaw XIII, Duke of Pomerania? 2. Where did Bogislaw XIII, Duke of Pomerania, die? ☞ **Updated note:** From the document "Bogislaw XIII, Duke of Pomerania", we know that **Anna of Pomerania's father was Bogislaw XIII, Duke of Pomerania.** According to the document, **Bogislaw XIII died on 7 March 1606, in Stettin (also known as Szczecin in Polish).** . . . Bogislaw XIII was a prince of Stettin and Wolgast, and a member of the Griffins. He was . . . ➜ **[Knowledge growth: True]** ☞ **Stop the iteration process and output the answer**
Vanilla RAG: No information is given in the passages about Anna Of Pomerania's father. ✗ **Self-RAG:** Not mentioned. ✗
Answer of DeepNote: Stettin (also known as Szczecin in Polish) ✓
Golden Answer: Stettin

Table 19: **An example of DeepNote on the 2WikiMQA dataset (max step $= 3$, max failure $= 2$). Bold blue text** represents all segments related to solving the problem (including both relevant and correct segments as well as relevant but incorrect segments). Green segments denote those used to support correct answers. Red segments represent key incorrect information and erroneous responses collected. The "✓" indicates a correct answer, while "✗" signifies an incorrect answer.

Question: What city was the author of Neighbors born in?
Process of DeepNote: ☞ **Init note:** Unfortunately, the provided document content **does not contain information about the birth city of the author of "Neighbors".** However, based on the available information, I can provide a note that includes the relevant details about the author of "Neighbors". Note: **The author of the short story "Neighbors" is Raymond Carver.** Unfortunately, the provided document content does not mention his birth city. However, it does provide information about the publication history of the short story, which was first published in "Esquire magazine" in 1971 and later included in several collections. ☞ **New queries:** 1. What is the birth city of American short story writer Raymond Carver? 2. Where was Raymond Carver, the author of the short story "Neighbors", born? ☞ **Updated note:** Based on the retrieved documents, I have supplemented the notes with the following content: Note: **The author of the short story "Neighbors" is Raymond Carver.** According to the available information, **Raymond Carver was born in Clatskanie, Oregon,** a mill town on the Columbia River. **This information is found in the document titled "Raymond Carver", which provides details about his birthplace and upbringing.** Although the document "Neighbors (short story)" does not mention his birth city, it does provide information about the publication history of the short story, which was first published in "Esquire magazine"... ➜ **[Knowledge growth: True]** ☞ **New queries:** 1. What is the birthplace of the American author of the short story collection "Will You Please Be Quiet, Please?"? 2. In which city was the author of the short story "Neighbors" raised? ☞ **Updated note:** Based on the retrieved documents, I have supplemented the notes with the following content: Note: The author of the short story "Neighbors" is Raymond Carver. Unfortunately, the retrieved documents do not provide information about his birth city. However, the document "Neighbors (short story)" does provide information about the publication history of the short story, which was first published in "Esquire magazine" in 1971 and later included in several collections, including "Will You Please Be Quiet, Please?" in 1976, "Where I'm Calling From" in 1989, and "Short Cuts" in 1993. Since the retrieved documents **do not provide information about Raymond Carver's birth city,** I was unable to find the answer to the question. However, I can suggest that further research may be needed to find the correct answer. ➜ **[Knowledge growth: False]** ☞ **Stop the iteration process and output the answer**
Vanilla RAG: Not mentioned ✗ **ReAct:** Raymond Carver ✗
Answer of DeepNote: Clatskanie ✓
Golden Answer: Clatskanie

Table 20: **An example of DeepNote on the MusiQue dataset (max step** $= 3$, **max failure** $= 1$). **Bold blue text** represents all segments related to solving the problem (including both relevant and correct segments as well as relevant but incorrect segments). Green segments denote those used to support correct answers. Red segments represent key incorrect information and erroneous responses collected. The "✓" indicates a correct answer, while "✗" signifies an incorrect answer.

Question: A man who played in the 1986 FIFA world cup played for what team during the 1982 Scottish League Cup Final?

Process of DeepNote:

☞ **Init note:** Unfortunately, the provided document content **does not contain information about a specific player who played in the 1986 FIFA World Cup and also played in the 1982 Scottish League Cup Final.** However, based on the available information, we can make an educated guess. Labid Khalifa played in the 1986 FIFA World Cup, but there is no information about him playing in the 1982 Scottish League Cup Final. On the other hand, **Hamish McAlpine played for Dundee United** for twenty years, **between 1966 and 1986, and won the Scottish League Cup twice**, **but it is not specified if he played in the 1986 FIFA World Cup.** However, since he played for Dundee United until 1986, it is possible that he played in the 1986 FIFA World Cup, but this **is not confirmed**. If we assume that Hamish McAlpine played in the 1986 FIFA World Cup, then based on the information provided, he played for Dundee United during the 1982 Scottish League Cup Final.

☞ **New queries:** 1. Which players from the 1986 FIFA World Cup participated in the 1982 Scottish League Cup Final? 2. Who were the Dundee United players that played in the 1986 FIFA World Cup?

☞ **Updated note:** Based on the retrieved documents, I have supplemented the notes with content not yet included but useful for answering the question. Unfortunately, the provided document content **still does not contain direct information about a specific player** who played in the 1986 FIFA World Cup and also played in the 1982 Scottish League Cup Final. However, I have added more information from the retrieved documents to provide a more comprehensive answer. **Labid Khalifa played in the 1986 FIFA World Cup, but there is no information about him playing in the 1982 Scottish League Cup Final.** On the other hand, **Hamish McAlpine played for Dundee United** for twenty years, **between 1966 and 1986**, and won the Scottish League Cup twice, but it is **not specified if he played in the 1986 FIFA World Cup.** However, since he played for Dundee United until 1986, it is possible that he played in the 1986 FIFA World Cup, but this is not confirmed. **Additionally, the 1986 FIFA World Cup squads document may contain information about players who played in the 1986 FIFA World Cup, but the provided text does not specify the players.** If we had access to the full squads document, we might be able to find a player who played in the 1986 FIFA World Cup and also played in the 1982 Scottish League Cup Final. . . . If we assume that Hamish McAlpine played in the 1986 FIFA World Cup, then based on the information provided, he played for Dundee United during the 1982 Scottish League Cup Final. **However, this is still a speculative answer, and we would need more information to confirm it.**. . . ➜ **[Knowledge growth: False]**

☞ **New queries:** 1. Which players participated in both the 1986 FIFA World Cup and the 1982 Scottish League Cup Final for Dundee United? 2. Did Hamish McAlpine play for Scotland in the 1986 FIFA World Cup?

☞ **Updated note:** Based on the retrieved documents, the supplemented notes are as follows: Unfortunately, the provided document content **does not contain information about a specific player** who played in the 1986 FIFA World Cup and also played in the 1982 Scottish League Cup Final. However, **based on the available information**, we can make an educated guess. Labid Khalifa played in the 1986 FIFA World Cup, but there is no information about him playing in the 1982 Scottish League Cup Final. On the other hand, Hamish McAlpine played for Dundee United for twenty years, between 1966 and 1986, and won the Scottish League Cup twice, **but it is not specified if he played in the 1986 FIFA World Cup.** However, **since he played for Dundee United** until 1986, it is **possible that he played in the 1986 FIFA World Cup, but this is not confirmed.** As per the document "Hamish McAlpine", it is mentioned that "He played for Dundee United for twenty years, between 1966 and 1986. Whilst with the club he won the Scottish Football League title once and the Scottish League Cup twice." Although it does not specifically mention the 1982 Scottish League Cup Final, it does mention that he won the Scottish League Cup twice, which suggests that he might have played in the 1982 Scottish League Cup Final. . . . It is worth noting that **the document "1986 FIFA World Cup squads" could potentially provide more information** about the players who played in the 1986 FIFA World Cup, **but unfortunately, the content of this document is not available.** ➜ **[Knowledge growth: False]**

☞ **Stop the iteration process and output the answer**

Answer of DeepNote: Dundee United ✗

Golden Answer: Celtic

Table 21: **Badcase analysis of DeepNote on the HotpotQA dataset (max step** $= 3$, **max failure** $= 2$**).** **Bold blue text** represents all segments related to solving the problem (including both relevant and correct segments as well as relevant but incorrect segments). Green segments denote those used to support correct answers. Red segments represent key incorrect information and erroneous responses collected. The "✓" indicates a correct answer, while "✗" signifies an incorrect answer.