ThinkNote: Enhancing Knowledge Integration and Utilization of Large Language Models via Constructivist Cognition Modeling

Zhipeng Xu[1], Zhenghao Liu[1*], Yukun Yan[2], Shuo Wang[2], Shi Yu[2], Zheni Zeng[2], Chaojun Xiao[2], Zhiyuan Liu[2], Ge Yu[1] and Chenyan Xiong[3]

[1]School of Computer Science and Engineering, Northeastern University, Shenyang, China
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[3]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, United States

Abstract

Large Language Models (LLMs) have demonstrated strong performance across a wide range of NLP tasks. However, they often exhibit suboptimal behaviors and inconsistencies when exposed to unfamiliar external information, underscoring their limitations in effectively leveraging such knowledge. Inspired by constructivist learning theory, we propose ThinkNote, a novel framework that enhances the external knowledge utilization of LLMs through a two-stage constructivist cognitive modeling process. Specifically, ThinkNote performs knowledge assimilation to align new information with the model's parametric memory, forming a coherent internal representation. It then applies thought accommodation to adapt internal reasoning, thereby promoting more consistent and reliable outputs. Extensive experimental results demonstrate that ThinkNote achieves a 10% improvement over strong baseline methods on various question-answering benchmarks. Further analysis indicates that ThinkNote effectively integrates and utilizes external knowledge to help LLMs generate accurate responses and improve their self-consistency. All data and codes are available at https://github.com/OpenMatch/ThinkNote.

1 Introduction

Large Language Models (LLMs), *e.g.* ChatGPT-4 (Achiam et al., 2023) and LLaMA (Touvron et al., 2023), have shown strong emergent abilities and achieved convincing performance in various NLP tasks (Wei et al., 2022a; Zhao et al., 2023a). Despite these advancements, LLMs often face challenges in effectively leveraging external knowledge to generate accurate responses, especially when encountering information that is noisy or incomplete (Liu et al., 2024; Xie et al., 2024; Asai et al., 2023). Such limitations lead to response inconsistencies and hallucinations, constraining their application in knowledge-intensive scenarios (Ji et al., 2023; Xu et al., 2024a; Wang et al., 2023).

Recent research investigates strategies for enhancing the integration and utilization of external knowledge in LLMs. These efforts aim to develop specialized modules that refine, reflect on, or summarize external knowledge before integration (Yu et al., 2023a; Xu et al., 2023) or filter the external knowledge based on its relevance to the current context (Peng et al., 2023; Zhao et al., 2023b; Asai et al., 2023). While these methods show their effectiveness in helping LLMs acquire new knowledge (Ji et al., 2023; Wei et al., 2024), they still treat LLMs as passive recipients of information (Steffe and Gale, 1995; Larochelle et al., 1998) and fail to address the underlying cognitive limitations that lead to insufficient knowledge utilization (Yu et al., 2025). This limitation leads to incorrect responses and can even cause degraded responses on certain tasks (Foulds et al., 2024; Shuster et al., 2021).

To address these limitations, we propose a shift from treating LLMs as passive consumers of knowledge to modeling them as active knowledge constructors, drawing inspiration from constructivist cognitive theories (Steffe and Gale, 1995; Papert, 1980; Piaget and Inhelder, 1972). Constructivism emphasizes that knowledge is not simply absorbed from the environment, but is actively built through experience, reflection, and assimilation of new information into existing mental frameworks (von Glasersfeld, 1984). Under this paradigm, external knowledge is not directly fused into the input context of LLMs but transformed and recontextualized based on the model's current understanding of the task or question (Schunk, 2012). Such a cognitively grounded design encourages models to resolve inconsistencies, discard irrelevant or misleading information, and synthesize new insights, similar to how humans build reliable understanding from noisy data (Bruner, 1990).

In this paper, we propose ThinkNote, a

* indicates corresponding author.

Findings of the Association for Computational Linguistics: EACL 2026, pages 211–229
March 24-29, 2026

cognition-inspired framework to activate LLMs to effectively integrate and utilize external knowledge by mimicking the cognitive processes of assimilation and accommodation from constructivism (Steffe and Gale, 1995). Specifically, THINKNOTE incorporates knowledge assimilation to transform external information into coherent insights and understanding, extending beyond query-specific notes (Yu et al., 2023a) to enrich the parametric knowledge of LLMs. The thought accommodation then updates the parametric memory of LLMs by incorporating the assimilated knowledge to refine and enhance the internal reasoning chain of LLMs. This two-step cognitive process is prompted to mimic different learning behaviors of humans, where we configure THINKNOTE using four instructions to process external knowledge by associating familiar knowledge, anchoring unfamiliar information, extracting logical reasoning, or identifying counterfactuals.

Our experiments on a range of knowledge-intensive question-answering tasks demonstrate the effectiveness of THINKNOTE. Compared with directly incorporating external knowledge, THINKNOTE consistently achieves around a 10% performance improvement across backbone LLMs of different scales. Moreover, THINKNOTE exhibits stronger robustness when facing noisy, incomplete, or partially misleading external knowledge, suggesting that actively constructing and reorganizing knowledge is more reliable than passive conditioning on retrieved evidence. Further analysis reveals that THINKNOTE leads to more concentrated information accumulation (Fan et al., 2025) on a small set of salient tokens, indicating its ability to identify and leverage critical elements of external knowledge during response generation.

2 Related Work

Large Language Models (LLMs) consistently face stability challenges when utilizing external knowledge (Zhao et al., 2023b; Asai et al., 2023). While LLMs are exposed to extensive corpora during training and demonstrate strong generalization across various tasks (Zhao et al., 2023a; Touvron et al., 2023), their performance declines when encountering unfamiliar, ambiguous, or contradictory information (Ji et al., 2023; Liu et al., 2024; Foulds et al., 2024). These limitations illustrate a notable gap between the availability of external knowledge and the model's capacity to effectively integrate and apply it, especially evident in knowledge-intensive tasks (Yu et al., 2023b; Shi et al., 2024; Jiang et al., 2023).

To mitigate these limitations, some methods attempt to enhance knowledge integration by optimizing the interface between external information and the model's input context (Cuconasu et al., 2024; Gao et al., 2023b). These methods typically aim to filter or reformat noisy inputs through knowledge summarization (Xu et al., 2023), reflection (Asai et al., 2023), or multi-step verification (Yu et al., 2023a; Peng et al., 2023). While these approaches have advanced the ability of LLMs to integrate and utilize external information, they often push the burden of knowledge alignment to external modules (Wang et al., 2025), where the LLMs remain conceptualized as passive processors, failing to reorganize or evaluate new information based on their current understanding (Madaan et al., 2023). This design choice limits their ability to support adaptive knowledge integration, especially in scenarios that require the model to actively construct and revise its understanding.

More recent efforts explore internal modifications to how LLMs represent, revise, or adapt knowledge. These include work on memory editing (Meng et al., 2022; Trivedi et al., 2023), belief tracking (Wilie et al., 2024), and metacognitive strategies such as self-verification and response calibration (Madaan et al., 2023; Liu et al., 2024). While promising, these approaches are often fragmented and tool-like: they provide mechanisms to intervene in model behavior without modeling the interaction between the model and external knowledge (Collins et al., 2024). There is still limited consensus on what constitutes effective knowledge integration at the cognitive level, and most current methods operate under engineering-driven constraints rather than principles drawn from learning theory or epistemology (Floridi and Nobre, 2024).

3 Methodology

In this section, we introduce our THINKNOTE framework, grounded in constructivist cognitive theory (Steffe and Gale, 1995), which aims to emulate the cognitive process of knowledge learning employed by human learners to interact with external knowledge. We first describe the overall workflow of THINKNOTE framework (Sec. 3.1) and then elaborate on the specific configuration strategies used to regulate learning behavior (Sec. 3.2).

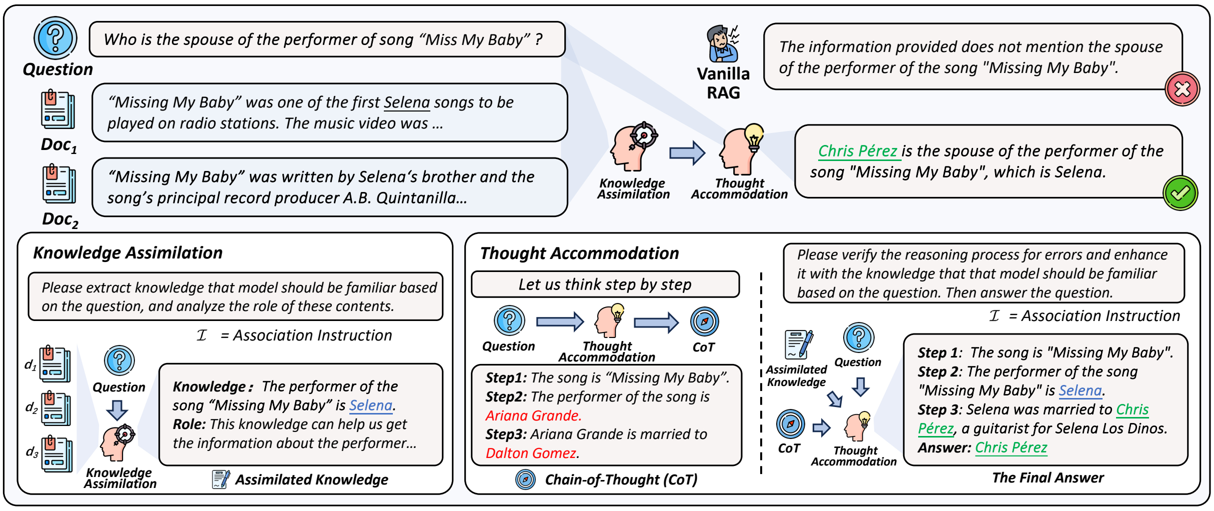


Figure 1: The Overview of the THINKNOTE Workflow. The framework follows a two-stage process of knowledge assimilation and thought accommodation to integrate external knowledge into model reasoning.

3.1 Enhancing Knowledge Integration and Utilization in LLMs via THINKNOTE

As shown in Figure 1, THINKNOTE is a cognition-inspired framework designed to enhance the ability of LLMs to integrate and utilize external knowledge to generate accurate responses, which draws on the cognitive principles of constructivist learning theory (Steffe and Gale, 1995). This framework simulates a two-stage cognitive process: THINKNOTE first conducts knowledge assimilation to align new information with the model's parametric memory, and then applies thought accommodation to adapt internal reasoning, thereby promoting more consistent and reliable outputs.

Knowledge Assimilation. Humans activate relevant memory (which is referred to as schema in constructivist cognitive theory) to interpret the meaning and function of external information and construct new understanding, which is called knowledge assimilation (Steffe and Gale, 1995; Moreno and Mayer, 1999). The knowledge assimilation stage in THINKNOTE simulates this process, driving the LLMs to align and integrate external knowledge with their internal parametric memory and point out the function of knowledge for problem-solving. Specifically, given a query q and a set of external knowledge $\mathcal{D}$, THINKNOTE guides the foundational model $\mathcal{M}$ to generates an assimilated rationale $\mathcal{T}$, which can be expressed as:

$$\mathcal{T} = \mathcal{M}\left(\mathcal{I}_{\text{ka}}, q, \mathcal{D}\right), \tag{1}$$

where $\mathcal{I}_{ka}$ indicates the instruction to mimic different learning behaviors. The assimilated knowledge $\mathcal{T}$ extends beyond knowledge summarization (Yu et al., 2023a) and serves to contextualize the knowledge within the problem scenarios, enabling the model to not only recall relevant facts but also explain how and why these facts matter.

Thought Accommodation. The thought accommodation refers to the adjustment of internal memory when new information challenges prior understanding in constructivist theory (Steffe and Gale, 1995; Larochelle et al., 1998). In THINKNOTE, this process is simulated by guiding the model $\mathcal{M}$ to revise its initial reasoning based on assimilated knowledge $\mathcal{T}$. Given a query q, THINKNOTE first conducts self-inquiry to generate a preliminary chain-of-thought $\mathcal{R}$ using its parametric memory:

$$\mathcal{R} = \mathcal{M}\left(\mathcal{I}_{\text{CoT}}, q\right). \tag{2}$$

Here $\mathcal{I}_{\text{CoT}}$ triggers a chain-of-thought reasoning grounded in prior knowledge, which may lack context or accuracy due to outdated or incomplete parametric memory. To address this, THINKNOTE guides the model $\mathcal{M}$ to revise the chain-of-thought $\mathcal{R}$ using the assimilated knowledge $\mathcal{T}$, producing the final output as:

$$y = \mathcal{M}\left(\mathcal{I}_{ta}, q, \mathcal{T}, \mathcal{R}\right), \tag{3}$$

where $\mathcal{I}_{ta}$ instructs the model $\mathcal{M}$ to align its reasoning with assimilated external knowledge. Through this thought accommodation process, THINKNOTE enables the model to approximate cognitive disequilibrium regulation, detecting mismatches between external knowledge and internal reasoning.

3.2 Configuring ThinkNote by Mimicking Different Human Learning Behaviors

Learning behaviors are goal-directed mental operations used to enhance knowledge utilization and problem-solving. ThinkNote employs flexible instructions $\mathcal{I}_{ka}$ and $\mathcal{I}_{ta}$ to simulate diverse human learning behaviors at each stage of the cognitive process, supporting both knowledge assimilation and thought accommodation. The instruction is instantiated in one of four constructivist modes: Anchoring, Association, Reasoning, and Reflection. The details of these instruction templates are shown in Appendix A.5.

Association. The association behavior maps new facts to known entities. Rather than introducing novel concepts, it reinforces and elaborates on familiar knowledge. For instance, when encountering information about electric vehicles, the model relates it to existing knowledge in their parametric memory of "batteries" and "charging stations", thereby deepening its understanding by associating these new contents to previously learned concepts (Steffe and Gale, 1995).

Anchoring. When humans encounter a new concept like "photosynthesis" in the query, they would start by identifying more basic concepts such as "chlorophyll" or "carbon dioxide" in external knowledge to create reference points in memory. Similarly, the instruction of anchoring behavior guides models to simulate this behavior and help them form an initial understanding by extracting related information from unfamiliar external knowledge. This behavior grounds its comprehension of queries and facilitates the expansion of topic-relevant knowledge (Yu et al., 2023a).

Reasoning. Inspired by Edge et al. (2024), the reasoning behavior guides the model to extract and organize logical relationships found in the external knowledge. This enables LLMs to build coherent and structured explanations of external knowledge. For example, when synthesizing documents on climate change policy, the model infers causal links, such as "carbon tax leads to emission reduction, which in turn impacts long-term economic growth"—to support principled reasoning.

Reflection. This behavior encourages the model to evaluate its outputs by comparing internal knowledge with external knowledge. By treating the external documents as authoritative references, the model can detect inconsistencies and revise its responses. For instance, if the model initially generates an outdated population statistic, reflection allows it to identify and correct the discrepancy based on up-to-date knowledge (Xu et al., 2024b).

4 Experimental Methodology

In this section, we describe the datasets, baselines, and implementation details of our experiments.

Dataset. Our experiment uses six datasets to evaluate the performance of ThinkNote, including NQ (Kwiatkowski et al., 2019), PopQA (Mallen et al., 2023), TriviaQA (Joshi et al., 2017), 2WikiMultiHopQA (Ho et al., 2020) and ASQA (Stelmakh et al., 2022). We use Contriever (Izacard et al., 2021) to obtain the external knowledge for all datasets to ensure fairness, following previous work (Asai et al., 2023; Wei et al., 2024). Due to the cost of inference, we randomly sample a subset of 500 questions from each dataset in our experiments in line with prior work (Trivedi et al., 2023; Yoran et al., 2023).

Evaluation Metrics. Following Asai et al. (2023) and Wei et al. (2024), we use *correctness* (*str-em*), *hit rates* (*hit*), and *recall* (*rec*) to evaluate the question-answering performance on the ASQA task and use *accuracy* (*acc*) for other tasks. All of the evaluation metrics are implemented referring to the ALCE toolkit (Gao et al., 2023a).

Baselines. The baselines consist of vanilla LLMs, LLMs with CoT, and some RAG models. We use Meta-Llama-3-Instruct and ChatGPT-4o-Mini as backbone LLMs to implement baseline models. Detailed descriptions, configurations, and implementation specifics of each baseline model are provided in Appendix A.3.

Vanilla LLMs. We employ two methods to prompt LLMs to answer questions based solely on their parametric memory. First, the Direct QA approach involves directly instructing the LLMs to answer questions. Second, we implement the Chain-of-Thought (Wei et al., 2022b) prompting, which generates a step-by-step reasoning process to arrive at the answer.

RAG Models. We implement three widely used RAG models as baselines, including vanilla RAG, Chain-of-Note (Yu et al., 2023a), and Self-RAG (Asai et al., 2023). The vanilla RAG model directly feeds retrieved passages as the context and asks the LLM to generate the answer (Ram et al., 2023). Chain-of-Note extends the CoT method to summarize the query-related knowledge from external knowledge for producing answers. Self-

Method	LLM	PopQA (*acc*)	NQ (*acc*)	TriviaQA (*acc*)	2WikiMHQA (*acc*)	(*rec*)	ASQA (*str-em*)	(*hit*)	(*rec*)
Vanilla LLMs									
Direct QA	Llama-3-Ins$_{8B}$	24.2	39.2	67.2	43.0	48.1	24.8	5.0	16.1
	Llama-3-Ins$_{70B}$	34.2	54.2	80.4	53.8	60.0	33.3	8.8	20.2
	ChatGPT-4o$_{\text{MINI}}$	32.6	51.0	75.0	47.4	52.3	31.4	7.8	18.5
CoT	Llama-3-Ins$_{8B}$	24.8	44.0	69.4	47.2	51.4	28.8	7.8	25.5
	Llama-3-Ins$_{70B}$	31.6	54.4	80.6	56.4	62.6	36.4	11.2	32.7
	ChatGPT-4o$_{\text{MINI}}$	32.4	53.2	76.2	51.0	55.3	32.4	8.0	21.6
LLM w/ RAG									
Vanilla RAG	Llama-3-Ins$_{8B}$	59.8	56.0	71.2	45.2	49.3	34.2	12.0	27.7
	Llama-3-Ins$_{70B}$	63.4	59.8	75.2	52.2	56.3	38.1	14.6	29.8
	ChatGPT-4o$_{\text{MINI}}$	62.0	64.8	77.8	51.4	56.2	38.9	13.2	28.0
Self-Refined	Self-RAG$_{7B}$	53.8	43.4	66.4	36.6	40.2	28.6	9.0	13.2
	Self-RAG$_{8B}$	54.8	46.2	67.0	37.2	42.6	37.6	15.6	21.4
	Chain-of-Note$_{8B}$	64.2	59.8	73.0	52.0	56.2	42.0	17.8	56.2
	Chain-of-Note$_{70B}$	68.2	67.8	78.4	56.8	61.1	45.0	20.6	54.9
THINKNOTE	Llama-3-Ins$_{8B}$	65.8	62.0	79.8	53.8	60.0	43.4	18.8	57.8
	Llama-3-Ins$_{70B}$	**69.8**	68.4	**85.4**	**63.2**	**68.4**	48.8	22.6	59.9
	ChatGPT-4o$_{\text{MINI}}$	69.8	**71.0**	83.4	61.0	67.4	**51.5**	**24.6**	**61.3**

Table 1: Overall Performance of THINKNOTE and Baselines on Different Knowledge-Intensive Benchmarks. For THINKNOTE, we use the instruction of association learning behavior to guide both knowledge assimilation and thought accommodation and evaluate the overall performance.

RAG (Asai et al., 2023) trains LLMs to employ a self-reflection mechanism to filter out irrelevant evidence effectively.

Implementation Details. We use different language models, including Meta-Llama-3-Instruct-8B, Meta-Llama-3-Instruct-70B, and ChatGPT-4o-Mini as the foundation model to evaluate the performance of the THINKNOTE. Specifically, we deploy Meta-Llama-3-Instruct models using the vLLM (Kwon et al., 2023), and use the OpenAI SDK to call the ChatGPT-4o-Mini API for experiments. The temperature is set to 0.2 for all models.

5 Evaluation Result

In this section, we first present the overall performance of THINKNOTE. We then analyze its effectiveness in leveraging knowledge. Finally, we demonstrate how THINKNOTE enhances the self-consistency of LLMs in answering questions.

5.1 Overall Performance

The overall performance of THINKNOTE on knowledge-intensive tasks is shown in Table 1.

Compared to vanilla RAG models, THINKNOTE delivers over a 10% improvement in performance, demonstrating its effectiveness in enabling LLMs to more accurately integrate and utilize external knowledge for question answering. While self-refinement methods focus on denoising external information to enhance generation, THINKNOTE surpasses these approaches with over a 4% improvement, highlighting the importance of fostering a deeper understanding of knowledge within LLMs rather than merely extracting surface-level content related to queries. Furthermore, THINKNOTE consistently outperforms all baseline models, regardless of the model scale or whether the backbone is black or white-box, indicating strong generalization and robustness across different model configurations.

Compared with vanilla RAG models, the Self-Refined methods exhibit performance degradation on TriviaQA and 2WikiMHQA datasets, showing that filtering query-related contents may risk omitting critical information from external knowledge. Unlike Self-Refined RAG, THINKNOTE employs a two-stage cognitive process that actively acquires essential knowledge from external information, rather than relying solely on LLMs to evaluate the relevance between queries and evidence. By leveraging constructivist cognitive modeling, THINKNOTE offers a promising way to emulate human learning behaviors and demonstrates its effectiveness by consistently achieving improvements across all test scenarios.

Method	Llama-3-Ins-8B PopQA (*acc*)	Llama-3-Ins-8B 2WikiMHQA (*acc*)	Llama-3-Ins-8B ASQA (*str-em*)	Llama-3-Ins-70B PopQA (*acc*)	Llama-3-Ins-70B 2WikiMHQA (*acc*)	Llama-3-Ins-70B ASQA (*str-em*)
Cognitive Modeling Ablation						
THINKNOTE	65.8	53.8	43.4	69.8	63.2	48.8
w/o *Knowledge Assimilation*	58.4	48.2	38.6	61.8	59.6	43.2
w/o *Thought Accommodation*	63.0	46.8	40.2	66.0	57.4	45.7
Learning Behaviors Ablation						
w. Anchoring Instruction	64.8	**54.6**	39.9	69.0	**63.6**	45.8
w. Association Instruction	**65.8**	53.8	**43.4**	**69.8**	63.2	48.8
w. Reasoning Instruction	64.0	52.8	40.3	68.4	59.6	48.3
w. Reflection Instruction	63.2	52.8	42.2	65.2	60.6	**50.2**

Table 2: Ablation Study of THINKNOTE. We examine the contributions of knowledge assimilation and thought accommodation, and vary instruction designs to simulate different learning behaviors. Experiments are conducted with Meta-Llama-3-Instruct-8B and Meta-Llama-3-Instruct-70B.

5.2 Ablation Study

As shown in Table 2, we conduct ablation studies to evaluate the contribution of each cognitive stage in THINKNOTE. We further analyze the effectiveness of THINKNOTE under different configurations of knowledge learning behavior instructions.

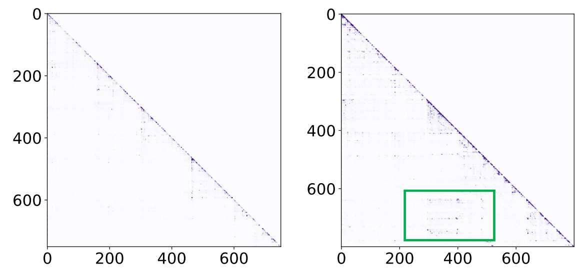
We first evaluate the performance of THINKNOTE variants on three QA tasks: PopQA, 2WikiMHQA, and ASQA. For PopQA, the knowledge assimilation stage in THINKNOTE contributes more significantly to performance improvements than the thought accommodation stage, highlighting that answering questions about less frequent entities heavily depends on acquiring information from external knowledge. In contrast, for 2WikiMHQA and ASQA, both the knowledge assimilation and thought accommodation modules are equally critical to the effectiveness of THINKNOTE, emphasizing the advantages of chain-of-thought reasoning in tackling more complex QA tasks. With increasing model scale, the effectiveness of knowledge assimilation declines, while the advantages of thought accommodation remain steady. This observation implies that although larger LLMs encode more knowledge within their parameters, deliberate reasoning remains essential for effectively tackling these multi-hop QA and long-form QA tasks.

We next examine the impact of knowledge learning behaviors in THINKNOTE by guiding the knowledge assimilation and thought accommodation stages using different sets of instructions, $\mathcal{I}_{ka}$ and $\mathcal{I}_{ta}$, respectively. Among all instruction types, the association learning behavior consistently achieves superior performance across diverse QA tasks. This behavior encourages LLMs to reinforce and extend their internal knowledge representations, rather than relying primarily on the assimilation of unfamiliar information from retrieved documents, as is characteristic of anchoring learning behavior. These findings emphasize the importance of integrating external knowledge through associations with the model's parametric memory. For subsequent experiments, we adopt the association learning behavior instruction to configure THINKNOTE.

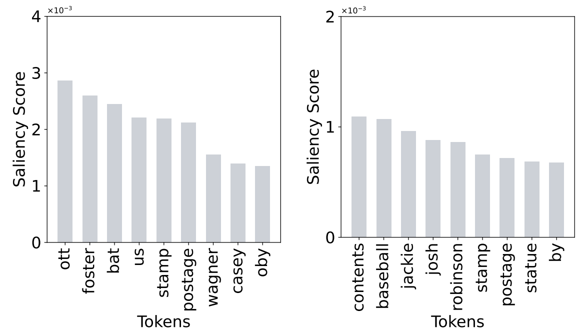
5.3 Information Flow Analysis for THINKNOTE in Knowledge Utilization

We further analyze how knowledge assimilation and thought accommodation influence the reasoning process in Figure 2. Specifically, we sample a case from the NQ dataset and follow Fan et al. (2025) in utilizing the saliency score to examine interactions between tokens, which is also referred to as information flow. Further details on the saliency score are provided in Appendix A.2.

We show the information flow for both Vanilla RAG and THINKNOTE in Figure 2(a) and Figure 2(b), respectively. The Vanilla RAG method demonstrates a dense and intricate pattern of information flow, where a broad range of tokens indiscriminately influence other tokens throughout the sequence. This indicates that Vanilla RAG lacks the mechanism to integrate knowledge or suppress irrelevant information, leading to an over-reliance on incidental associations. In contrast, THINKNOTE guides the information flow to some specific tokens (highlighted with the green box) by modeling cognitive processes. These tokens contribute to the accumulation of information flow as part of a

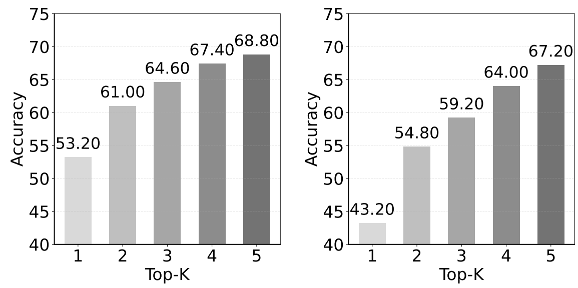


(a) Token-level Information Flow of Vanilla RAG. (b) Token-level Information Flow of THINKNOTE.

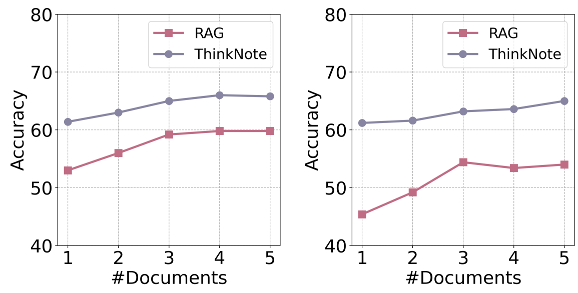


(c) Top Tokens in Information Accumulation as Identified by Vanilla RAG. (d) Top Tokens in Information Accumulation as Identified by THINKNOTE.

Figure 2: Information Flow Analysis for THINKNOTE. We implement both Vanilla RAG and THINKNOTE with Meta-Llama-3-Instruct-8B. Darker-colored points indicate higher saliency scores in information flow analysis.



(a) Retrieval Score (PopQA). (b) Retrieval Score (NQ).



(c) Accuracy (PopQA). (d) Accuracy (NQ).

Figure 3: Performance of RAG and Retrieval Models with Top-K Document Truncation. We implement both Vanilla RAG and THINKNOTE with Meta-Llama-3-Instruct-8B.

chain-of-thought. This indicates that THINKNOTE can effectively integrate and utilize information to help LLMs generate responses, reflecting a more goal-directed reasoning process, where irrelevant or distracting information is attenuated.

We further analyze the accumulation of information flow in both Vanilla RAG and THINKNOTE in Figure 2(c) and Figure 2(d), highlighting the top 10 tokens with the highest accumulated saliency scores. In this example, the query is: "Who was the first baseball player to be featured on a postage stamp?" The correct answer is "Jackie Robinson". For Vanilla RAG, the tokens with the highest accumulated saliency are generic terms like "stamp" or "postage". While these words are related to the question, they are helpless in answering the question. In contrast, THINKNOTE assigns higher saliency to key content words such as "jackie", "robinson", and "baseball", indicating a more effective focus on the discriminative elements for problem-solving. This highlights its effectiveness in gathering useful information to enhance the knowledge utilization capability of LLMs.

5.4 Effectiveness of THINKNOTE in Knowledge Integration and Utilization

In this subsection, we evaluate the capability of THINKNOTE to utilize external knowledge to generate responses accurately.

As shown in Figure 3, we evaluate the sensitivity of both the vanilla RAG and THINKNOTE to noisy external knowledge by feeding them the top-K retrieved documents. Figures 3(a) and 3(b) show the top-K retrieval accuracy, which reflects the concentration of query-relevant information in documents. As K increases from 1 to 5, retrieval accuracy consistently improves, indicating a higher likelihood of relevant content being included. However, this improvement saturates after $K = 3$, showing that additional documents tend to introduce more noise than useful information. Figures 3(c) and 3(d) compare the performance of THINKNOTE and baseline methods. While the performance of Vanilla RAG remains stable or even degrades when $K \geq 3$, indicating its limited robustness to noisy inputs, THINKNOTE continues to benefit from the additional documents. This trend highlights THINKNOTE's superior ability to identify and utilize query-relevant information even in the presence of increased noise.

Method	Has-Answer			Miss-Answer			Internal Knowledge		
	PopQA	NQ	TriviaQA	PopQA	NQ	TriviaQA	PopQA	NQ	TriviaQA
Llama-3-Ins-8B									
Direct QA	34.0	47.5	80.1	2.6	21.4	33.8	100.0	100.0	100.0
Vanilla RAG	85.5	76.2	91.7	3.2	6.3	18.0	89.3	79.1	88.7
Chain-of-Note	93.9	85.6	95.0	3.8	8.2	15.8	95.9	83.2	89.0
ThinkNote	90.1	80.4	97.0	8.3	25.8	36.0	99.2	92.9	97.3
Llama-3-Ins-70B									
Direct QA	45.6	63.9	91.4	9.0	33.3	51.8	100.0	100.0	100.0
Vanilla RAG	89.0	83.6	95.6	7.1	12.6	22.3	84.8	82.7	87.8
Chain-of-Note	96.2	90.0	97.5	6.4	14.5	28.8	93.6	85.2	90.0
ThinkNote	95.9	88.3	97.5	12.2	33.3	54.0	93.6	94.1	98.0

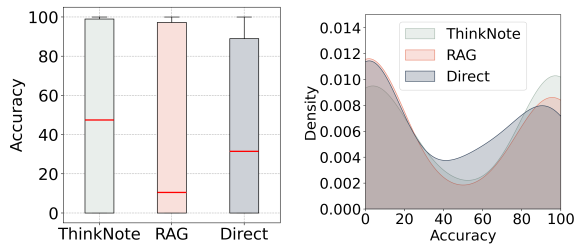
Table 3: Experimental Results on Evaluating the Knowledge Usage Ability of Different RAG Models. Results are reported under three scenarios, evaluating the capabilities of models in external knowledge utilization, robustness to missing evidence, and knowledge conflict resolution.

In Table 3, we then conduct three testing scenarios to evaluate the effectiveness of ThinkNote: Has-Answer, Miss-Answer, and Internal Knowledge. The Has-Answer scenario comprises queries where the external information contains the correct answers, assessing the model's capability to effectively utilize external knowledge. Conversely, the Miss-Answer scenario involves queries where the external information fails to provide the correct answers. The Internal Knowledge scenario evaluates the ability of LLMs to resolve conflicts between internal and external knowledge.

As shown in the evaluation results, Chain-of-Note and ThinkNote methods significantly outperform Vanilla RAG, highlighting their ability to extract critical information for answering questions. In the Miss-Answer scenario, ThinkNote presents strong effectiveness by doubling the accuracy of Chain-of-Note while maintaining the performance of Direct QA. This highlights the robustness of ThinkNote in mitigating the negative impact of noise in external knowledge. Moreover, in the Internal Knowledge scenario, ThinkNote achieves the highest accuracy, illustrating its ability to alleviate conflicts between the external knowledge and the parametric memory of LLMs.

5.5 Effectiveness of ThinkNote in Improving Self-Consistency of LLMs

In this subsection, we explore the self-consistency of ThinkNote and baseline models in answering questions, as shown in Figure 4. Specifically, we conduct 100 sampling iterations to calculate the ratio of correction. The accuracy reflects the model's tendency to generate correct or incorrect answers consistently. We compare three methods: Direct



(a) The Accuracy of Sampled Responses. (b) Density of the Accuracy of Correct Answers.

Figure 4: Answering Consistency of Different Methods. We show the performance on the 2WikiMultiHopQA dataset. All models are implemented with Meta-Llama-3-Instruct-8B.

QA, Vanilla RAG, and ThinkNote. For Vanilla RAG and Direct QA, the answer generation module is used for sampling, while ThinkNote utilizes the thought accommodation to generate sampled responses.

We first show the accuracy of ThinkNote and baseline models in Figure 4(a). ThinkNote demonstrates its effectiveness by achieving a median accuracy of around 47%, outperforming the Vanilla RAG (10%) and the Direct QA (31%). Notably, the substantially lower median accuracy of Vanilla RAG compared to Direct QA indicates that incorporating retrieved documents can hinder the LLM's ability to produce correct answers, rather than improve it (Xie et al., 2024).

Figure 4(b) further illustrates the density distribution of answering accuracy. Compared with the Vanilla RAG, our ThinkNote improves self-consistency by reducing the density of examples with accuracy between 30% and 80%. In contrast to vanilla RAG, ThinkNote substantially increases the number of queries answered correctly, highlighting its stronger capability to guide LLMs in effectively leveraging external knowledge to produce reliable answers.

6 Conclusion

We present THINKNOTE, a cognition-inspired framework that empowers LLMs to actively construct knowledge through assimilation and accommodation, rather than passively consuming external information. By mimicking key cognitive processes from constructivist theory, THINKNOTE enables more effective integration and utilization of external knowledge and improves self-consistency in correctly answering questions.

Limitation

THINKNOTE demonstrates its effectiveness in integrating and utilizing external knowledge to enhance the response accuracy of white-box and black-box LLMs. However, the cognitive process modeling requires inference with the LLMs three times to generate a final answer, which brings additional time latency and increases API call costs. Furthermore, the inputs provided to the thought accommodation tend to be lengthy, as they include outputs from knowledge assimilation and chain-of-thought.

Acknowledgments

This work is partly supported by the National Natural Science Foundation of China (No. 62576082 and No. 62461146205) and CCF-Zhipu Large Model Innovation Fund (No. 202403).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *ArXiv Preprint*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of ICLR*.

Jerome Bruner. 1990. *Acts of Meaning*. Harvard University Press.

Katherine M Collins, Ilia Sucholutsky, Umang Bhatt, Kartik Chandra, Lionel Wong, Mina Lee, Cedegao E Zhang, Tan Zhi-Xuan, Mark Ho, Vikash Mansinghka, et al. 2024. Building machines that learn and think with people. *Nature human behaviour*, 8(10):1851–1863.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. In *Proceedings of SIGIR*, pages 719–729.

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

Sinan Fan, Liang Xie, Chen Shen, Ge Teng, Xiaosong Yuan, Xiaofeng Zhang, Chenxi Huang, Wenxiao Wang, Xiaofei He, and Jieping Ye. 2025. Improving complex reasoning with dynamic prompt corruption: A soft prompt optimization approach. In *Proceedings of ICLR*.

Luciano Floridi and Anna C Nobre. 2024. Anthropomorphising machines and computerising minds: the crosswiring of languages between artificial intelligence and brain & cognitive sciences. *Minds and Machines*, 34(1):5.

Philip Feldman Foulds, R James, and Shimei Pan. 2024. Ragged edges: The double-edged sword of retrieval-augmented chatbots. *ArXiv preprint*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023a. Enabling large language models to generate text with citations. In *Proceedings of EMNLP*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. *ArXiv preprint*.

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 COLING*, pages 6609–6625.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *ArXiv Preprint*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, (12):248:1–248:38.

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 EMNLP*, pages 7969–7992.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of ACL*, pages 1601–1611.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS*.

Marie Larochelle, Nadine Bednarz, and James W Garrison. 1998. *Constructivism and education*. Cambridge University Press.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. In *Processing of NeurIPS*, volume 36, pages 46534–46594.

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 ACL*, pages 9802–9822.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Proceedings of NeurIPS*, volume 35, pages 17359–17372.

Roxana Moreno and Richard E Mayer. 1999. Cognitive principles of multimedia learning: The role of modality and contiguity. *Journal of educational psychology*, 91(2):358.

Seymour Papert. 1980. *Children, computers, and powerful ideas*, volume 10. Harvester Eugene, OR, USA.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *ArXiv Preprint*.

Jean Piaget and Bärbel Inhelder. 1972. *The Psychology of the Child*. Basic Books.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, pages 1316–1331.

Dale H. Schunk. 2012. *Learning Theories: An Educational Perspective*. Pearson Education.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. Replug: Retrieval-augmented black-box language models. In *Proceedings of NAACL*, pages 8371–8384.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Proceedings of EMNLP Findings*, pages 3784–3803.

Leslie P Steffe and Jerry Edward Gale. 1995. *Constructivism in Education*. Psychology Press.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of EMNLP*, pages 8273–8288.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *ArXiv preprint*.

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 ACL*, pages 10014–10037.

Ernst von Glasersfeld. 1984. An introduction to radical constructivism. *The invented reality*, 17:17–40.

Boshi Wang, Xiang Yue, and Huan Sun. 2023. Can chatgpt defend its belief in truth? evaluating llm reasoning via debate. In *Findings of EMNLP*, pages 11865–11881.

Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan O Arik. 2025. Astute rag: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. In *Proceedings of ACL*, pages 30553–30571.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Proceedings of NeurIPS*, pages 24824–24837.

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

Bryan Wilie, Samuel Cahyawijaya, Etsuko Ishii, Junxian He, and Pascale Fung. 2024. Belief revision: The adaptability of large language models reasoning. *ArXiv preprint*.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *Proceedings of ICML*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *ArXiv preprint*.

Yunqi Xu, Tianchi Cai, Jiyan Jiang, and Xierui Song. 2024a. Face4rag: Factual consistency evaluation for retrieval augmented generation in chinese. In *Proceedings of KDD*, pages 6083–6094.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024b. Hallucination is inevitable: An innate limitation of large language models. *ArXiv preprint*.

Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. 2023. Answering questions by meta-reasoning over multiple chains of thought. In *Proceedings of EMNLP*, pages 5942–5966.

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

Xiaohan Yu, Zhihan Yang, and Chong Chen. 2025. Unveiling the potential of multimodal retrieval augmented generation with planning. *ArXiv preprint*.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023b. Augmentation-adapted retriever improves generalization of language models as generic plug-in. In *Proceedings of ACL*, pages 2421–2436.

Xuanwang Zhang, Yunze Song, Yidong Wang, Shuyun Tang, Xinfeng Li, Zhengran Zeng, Zhen Wu, Wei Ye, Wenyuan Xu, Yue Zhang, et al. 2024. Raglab: A modular and research-oriented unified framework for retrieval-augmented generation. *ArXiv Preprint*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023a. A survey of large language models. *ArXiv preprint*.

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, and Jianshu Chen. 2023b. Thrust: Adaptively propels large language models with external knowledge. In *Proceedings of NeurIPS*, volume 36, pages 69930–69948.

A Appendix

A.1 License

The 2WikiMultiHopQA and ASQA datasets are released under the CC BY 3.0 license, allowing users to freely use and modify the data with proper attribution. The Natural Questions (NQ) dataset is licensed under CC BY-SA 4.0, enabling users to share and modify the data, provided they attribute the source and distribute any modifications under the same license. The PopQA and TriviaQA datasets are distributed under the MIT License, which allows for free use, modification, and distribution of the data, as long as the original copyright notice is retained. All these licenses permit the datasets to be used for academic purposes.

A.2 Further Details on the Computation of Saliency Scores

We follow Fan et al. (2025) and use the saliency score to reflect the strength and direction of information flow, which refers to how much and in what way one token influences another during the reasoning process. Specifically, for each layer l, the saliency between tokens is defined as:

$$S = \left| \sum_h \left(A_{h,l} \odot \frac{\partial L(x)}{\partial A_{h,l}} \right) \right|, \quad (4)$$

where $A_{h,l}$ denotes the attention matrix of the h-th head in layer l, $L(x)$ is the loss function (cross-entropy), and $\odot$ represents the Hadamard product. This formulation combines attention weights with their gradient-based sensitivities, allowing us to capture both what the model focuses on and how crucial that focus is to the final prediction. We extract $A_{h,l}$ from the 21st layer, using Meta-Llama-3-8B-Instruct as the foundation model.

A.3 More Details of Baseline Models

In this subsection, we describe the implementation details of the baseline models in detail.

For Direct QA, we input questions directly to the foundation model without including any additional prompts. And for Chain-of-Thought models, we follow the setting from Wei et al. (2022b) to prompt LLMs to think step by step. We use the same prompts as Ram et al. (2023) to implement the Vanilla RAG model. The prompt template for Chain-of-Note follows the experimental setting of Yu et al. (2023a). For Self-RAG (Asai et al., 2023), we utilize their official codebase to implement Self-RAG$_{7B}$. The Self-RAG$_{8B}$ model is implemented using RAGLAB (Zhang et al., 2024) and fine-tuned based on Meta-Llama-3-Ins-8B to ensure a fair comparison across our experiments.

A.4 Case Study

We choose one case in Figure 5 to show the effectiveness of THINKNOTE by comparing with Vanilla RAG and Chain-of-Note models. For the question "Where was the director of film Hatchet II born?", both Vanilla RAG and Chain-of-Note models fail to generate correct answers. Specifically, the Vanilla RAG model is misled by the retrieved documents, confusing the directors of "Hatchet II" and "Hatchet III" while answering the given question. In contrast, the Chain-of-Note model analyzes the documents in the reasoning process and generates notes that correctly identify the director as "Adam Green". However, despite capturing this key information, the Chain-of-Note model heavily relies on extracting answers from its notes without developing a deeper understanding of external knowledge or associating it with the parametric memory of LLMs. Unlike both Vanilla RAG and Chain-of-Note methods, THINKNOTE not only emphasizes the film but also identifies its director. Moreover, through using the assimilated rationale generated by the knowledge assimilation, THINKNOTE accurately identifies Adam Green's birthplace. This demonstrates THINKNOTE's strong ability to harmonize external knowledge with parametric memory, significantly improving the quality of the responses.

We then analyze the case in Figure 6. This case is from the NQ dataset and asks about the person pictured on "5000 dollar bill". This is not a typical RAG scenario, but it happens often: the retrieved evidence only provides the background knowledge needed to answer the question while not specifying a specific answer. To generate the correct response, models need to accurately analyze the retrieval evidence and activate the parameter knowledge to conduct the reasoning process. When answering this question, Vanilla RAG is limited by its narrow interpretation of the retrieved evidence, leading it to conclude that only \$1, \$5, \$10, \$20, and \$100 dollar bills existed. The Chain-of-Note model generates brief summaries of each passage, but it struggles to provide deeper insights or draw more comprehensive analyses. In contrast, our THINKNOTE model first deepens current experience based on background knowledge (Step1), then attempts to activate LLMs to use the parametric memory to

Question: Where was the director of film Hatchet II born?
Documents: ... The film would leave behind the usual setting of Honey Island Swamp and instead go for a more suburban setting ... "Hatchet II" was announced in November 2008 ... also making uncredited cameos were director Adam Green... Hatchet III is a 2013 American slasher film written by Adam Green and directed by B. J. McDonnell.

Vanilla RAG

The director of film Hatchet II was not mentioned as being born in any specific location... However, the director of Hatchet III, B. J. McDonnell, was not mentioned at all in the provided documents. Therefore, the answer is unknown.

Chain-of-Note

Reading Note...
Document 2 is about the actor Parry Shen, who has starred in the Hatchet series, including Hatchet, Hatchet II, Hatchet III, and Victor Crowley.
Document 3 is about the Hatchet film series, which consists of four films, including Hatchet, Hatchet II, Hatchet III, and Victor Crowley. It mentions the director Adam Green and the main antagonist Victor Crowley...
Document 5 is about the film Hatchet II...
Answer: Unfortunately, the passages do not provide information about the birthplace of the director of film Hatchet II.

ThinkNote

Step 1: Identify the film
The film is "Hatchet II".
Step 2: Identify the director of the film
The director of "Hatchet II" is Adam Green, who is also the creator of the Hatchet film series.
Step 3: Research Adam Green's birthplace
After conducting research, I found that Adam Green, the director of "Hatchet II", was born in Holliston, Massachusetts, USA.
This information is relevant to understanding the context of the question...
Answer: Adam Green, the director of "Hatchet II", was born in Holliston, Massachusetts, USA.

Figure 5: Case#1 of THINKNOTE. We annotate the knowledge that needs to be acquired during answering the question, and highlight the correct responses and incorrect responses using different colors.

continuously answer the given question (Step2), and finally accurately generates the answer based on the reasoning process (Step3).

As shown in Figure 7, another representative case highlights the limitations of LLMs in incorporating and fully understanding external knowledge. This case is from the 2WikiMultiHopQA dataset and asks for the birthplace of "Şehzade Yusuf Izzeddin's father". Different from the first Case, the retrieved evidence in this instance contains the crucial information required to answer the question. However, the evidence is entangled with complex logic and noise, making it challenging for the model to extract relevant information and refine the reasoning process. The Vanilla RAG model struggles to identify useful information from the retrieval evidence. Similarly, the Chain-of-Note model exhibits confusion during the summarization process: while it successfully identifies key information, it incorrectly dismisses it as irrelevant, suggesting difficulty in integrating external knowledge and resulting in a suboptimal response. In contrast, THINKNOTE generates an accurate response by leveraging knowledge assimilation to accurately analyze the retrieved evidence and then refining the reasoning process through thought accommodation.

A.5 Instructions for Mimicking Different Learning Behaviors

We present all the instructions used to mimic different learning behaviors in this subsection. The anchoring behavior instructions used by THINKNOTE are illustrated in Figure 9, guiding the model to ground its understanding of unfamiliar concepts by identifying essential information from external knowledge. For association behavior, THINKNOTE adopts the configuration shown in Figure 10, enabling it to connect new inputs with relevant prior knowledge in its parametric memory. The reasoning behavior is supported by the instructions in Figure 11, which help the model extract and organize logical relationships for structured understanding. Finally, the reflection behavior is guided by the instructions in Figure 12, allowing the model to evaluate and revise its outputs by comparing them against reliable external sources.

Question: Who is pictured on the 5000 dollar bill?
Documents: The $10 bill is unique in that it is the only denomination in circulation in which the portrait faces to the left. It also features one of two non-presidents on currently issued U.S. bills, the other being Benjamin Franklin on the $100 bill ... Hamilton also appears on the $500 Series EE Savings Bond. Hamilton's portrait has been featured on the front of the U.S. $10 bill since 1928. The source of the engraving is...

Vanilla RAG

The provided information does not mention the $5000 bill. It only talks about the $1, $5, $10, $20, and $100 bills, and the individuals featured on them, such as Edwin Stanton, Andrew Jackson, Abraham Lincoln, Alexander Hamilton, and others. Therefore, it is not possible to determine who is pictured on the $5000 bill.

Chain-of-Note

Document 1: The passage describes the design elements of the $5 and $10 Demand Notes, featuring portraits of Abraham Lincoln and Alexander Hamilton, respectively.
Document 2: The passage provides information about the United States ten-dollar bill, featuring a portrait of Alexander Hamilton facing to the left, and its uniqueness in featuring a non-president on U.S. paper currency...
Document 5: The passage discusses Alexander Hamilton's appearances on various denominations of U.S. currency, including the $10 bill, and his portrait being featured on the front of the bill since 1928.
Since no passage is relevant to the given question, I will provide a direct answer:
There is no information provided about who is pictured on the $5000 bill.

ThinkNote

Step1: The $5000 bill is not a currently circulating denomination of US currency. In fact, the largest denomination of US currency that is currently in circulation is the $100 bill.
Step2: However, the $5000 bill has been issued in the past. During the 1930s, the US government issued a $5000 Gold Certificate featuring a portrait of President James Madison.
Step3: The portrait on the $5000 Gold Certificate is likely to be a representation of President James Madison, who was the 4th President of the United States and is known for his role in drafting the United States Constitution.
Answer: The person pictured on the $5000 bill is President James Madison.

Figure 6: Case#2 of THINKNOTE. We annotate the knowledge that needs to be acquired during answering the question, the correct responses and incorrect responses using different colors.

Question: : Where was the father of Şehzade Yusuf Izzeddin born?
Documents: War, the Baghdad expedition of 1535, and the Siege of Szigetvár in 1566...Dürrünev gave birth to the couple's first child, a son, Şehzade Yusuf Izzeddin on 29 September 1857. After Abdülaziz's accession to the throne on 25 June 1861, Dürrünev was given the title of ""Baş Kadın""... Her elder sister was married to Şehzade Yusuf Izzeddin... The brother of Mehmed V, he became heir to the throne after the 1916 suicide of Abdülaziz's son Şehzade Yusuf Izzeddin as the eldest male member of the House of Osman... He acceded to the throne after the death of Mehmed V. He was girded with the Sword of Osman on 4 July 1918, as the thirty-sixth

Vanilla RAG

The provided information does not mention the father of Şehzade Yusuf Izzeddin. However, it is mentioned that Şehzade Yusuf Izzeddin was born on 29 September 1857.

Chain-of-Note

Reading Notes:
Document 1 does not mention the father of Şehzade Yusuf Izzeddin.
Document 2 mentions that Dürrünev gave birth to a son, Şehzade Yusuf Izzeddin, on 29 September 1857. However, it does not provide information about the father of Şehzade Yusuf Izzeddin.
Document 3 does not mention the father of Şehzade Yusuf Izzeddin.
Document 4 (mentions that Şehzade Yusuf Izzeddin was the brother of Mehmed VI and the son of Abdülaziz. However, it does not provide information about the father of Şehzade Yusuf Izzeddin.
Answer: Unfortunately, the provided passages do not provide information about the father of Şehzade Yusuf Izzeddin.

ThinkNote

Step1: Şehzade Yusuf Izzeddin was a Ottoman prince.
Step2: Ottoman princes were typically the sons of sultans or high-ranking officials.
Step3:. To find the father of Şehzade Yusuf Izzeddin, we need to find his parentage.
Step4: According to Document 4, Şehzade Yusuf Izzeddin was the son of Sultan Abdülaziz.
Step5: Sultan Abdülaziz was born in Constantinople, which was the capital of the Ottoman Empire at that time.
Answer: The birthplace of the father of Şehzade Yusuf Izzeddin is Constantinople (modern-day Istanbul).

Figure 7: Case#3 of THINKNOTE. We annotate the knowledge that needs to be acquired during answering the question, the correct responses and incorrect responses using different colors.

Prompt for CoT

```
Answer the question: {Question} Let us think step by step.
```

Prompt for Vanilla RAG

```
Passage: {Passages}
Based on your knowledge and the provided information, answer the question: {Question}
```

Prompt for Chain-of-Note

```
Task Description:
1. Read the given question and five Wikipedia passages to gather relevant information.
2. Write reading notes summarizing the key points from these passages.
3. Discuss the relevance of the given question and Wikipedia passages.
4. If some passages are relevant to the given question, provide a brief answer based on the
passages.
5. If no passage is relevant, directly provide answer without considering the passages.
Question: {Question}
Passages: {Passages}
```

Figure 8: Prompts for Baseline Models.

The Anchoring Instruction of ThinkNote

Knowledge Assimilation.

[Instruction]: You are a cognitive scientist, to answer the following question: {Question}
I will provide you with several retrieved passages:
Passages: {Passages}

Task Description:
Please extract content that may be unfamiliar to the model from these passages, which can provide the model with relevant background and unknown knowledge and concepts, helping it better understand the question. and analyze the role of these contents.

Thought Accommodation.

[Instruction]: To solve the problem, please think and reason step by step, then answer.
Question: {Question}

[Instruction]: Here is an answer generated by a language model with the reasoning process.
Question: {Question}
Answer: {Chain-of-Thought}
To provide the language model with relevant background and unknown knowledge and concepts, helping it better understand the question.
I retrieved some knowledge that is may unfamiliar with the model:
Knowledge: {Assimilated Knowledge}

Please verify the above reasoning process for errors, then enhance this reasoning process using retrieved knowledge to help it better understand the question.
Afterward, give the answer based on the enhanced reasoning process.

Figure 9: The Anchoring Instruction of THINKNOTE.

The Association Instruction of ThinkNote

Knowledge Assimilation.

[Instruction]: You are a cognitive scientist, to answer the following
Question: {Question}
I will provide you with several retrieved passages:
Passages: {Passages}

Task Description:
Please extract foundational knowledge that may be familiar to the model or advanced information beyond the model's already familiar foundational knowledge from these passages, and analyze the role of these contents. Summarize and consolidate these contents, which should deepen the model's understanding of the question through familiarity with these basic and advanced pieces of information.
This process aims to encourage the model to comprehend the question more thoroughly and expand its knowledge boundaries.

Thought Accommodation.

[Instruction]: To solve the problem, please think and reason step by step, then answer.
Question: {Question}

[Instruction]: Here is an answer generated by a language model with the reasoning process.
Question: {Question}
Answer: {Chain-of-Thought}
To deepen the language model's understanding of the question through familiarity with basic and advanced pieces of information. Encourage the language model to comprehend the question more thoroughly and expand its knowledge boundaries.
I retrieved some foundational knowledge that is familiar to the model or advanced information beyond the language model's already familiar foundational knowledge from these passages.
Knowledge: {Assimilated Knowledge}

Please verify the above reasoning process for errors,
then enhance this reasoning process using retrieved knowledge to deepen the understanding of the question through familiarity with basic and advanced pieces of information, comprehend the question more thoroughly, and expand the knowledge boundaries.
Afterward, give the answer based on the enhanced reasoning process.

Figure 10: The Association Instruction of THINKNOTE.

The Reasoning Instruction of ThinkNote

Knowledge Assimilation.

[Instruction]: You are a logician, to answer the following question:
{Question}
I will provide you with several retrieved passages:
Passages: {Passages}

Task Description:
Please extract content from these passages that can help enhance the model's causal reasoning and logical inference abilities.
Consolidate these contents, and analyze how the selected information may impact the improvement of the model's causal reasoning and logical inference capabilities.

Thought Accommodation.

[Instruction]: To solve the problem, please think and reason step by step, then answer.
Question: {Question}

[Instruction]: Here is an answer generated by a language model with the reasoning process.
Question: {Question}
Answer: {Chain-of-Thought}
To improve the language model's causal reasoning and logical inference capabilities.
I retrieved some knowledge that can help enhance the language model's causal reasoning and logical inference abilities.
Knowledge: {Assimilated Knowledge}

Please verify the above reasoning process for errors, then enhance this reasoning process using retrieved knowledge to enhance the causal reasoning and logical inference abilities.
Afterward, give the answer based on the enhanced reasoning process.

Figure 11: The Reasoning Instruction of THINKNOTE.

The Reflection Instruction of ThinkNote

Knowledge Assimilation.

[Instruction]: Fact-checking refers to the process of confirming the accuracy of a statement or claim through various sources or methods.
This process aims to ensure that statements or claims are based on reliable and verifiable information while eliminating inaccurate or misleading content.
Fact-checking may involve the examination of data, literature, expert opinions, or other trustworthy sources. In the context of artificial intelligence, model illusion refers to the overconfidence response of the AI. When a model exhibits an 'illusion' (a tendency to output deceptive data), it indicates that the training data used by the model does not necessarily support the rationality of its outputs.
You are a scientist researching fact-checking and addressing model illusions in artificial intelligence. To answer the following question:
{Question}
I will provide you with several retrieved passages:
Passages: {Passages}

Task Description:
Please extract content from these passages that may be contradictory to the model's existing knowledge. Identify information that, when added, could update the model's knowledge and prevent factual errors, alleviating model illusions. Note that these passages are retrieved from the most authoritative knowledge repositories, so they are assumed to be correct.

Thought Accommodation.

[Instruction]: To solve the problem, please think and reason step by step, then answer.
Question: {Question}

[Instruction]: Here is an answer generated by a language model with the reasoning process.
Question: {Question}
Answer: {Chain-of-Thought}
To update the language model's knowledge and prevent factual errors, alleviating model illusions.
I retrieved some knowledge that may update the language model's knowledge and prevent factual errors, alleviating model illusions
Note that these passages are retrieved from the most authoritative knowledge repositories, so they are assumed to be correct.
Knowledge: {Assimilated Knowledge}

Please verify the above reasoning process for errors, then enhance this reasoning process using retrieved knowledge to update the language model's knowledge, prevent factual errors and alleviate model illusions.
Afterward, give the answer based on the enhanced reasoning process.

Figure 12: The Reflection Instruction of THINKNOTE