Disentangling Memory and Reasoning Ability in Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated strong performance in handling complex tasks that require both extensive knowledge and reasoning abilities. However, the existing LLM inference pipeline operates as an opaque process without explicit separation between knowledge retrieval and reasoning steps, making the model's decision-making process unclear and disorganized. Recent research has shown that this ambiguity will lead to issues such as knowledge forgetting, which significantly impact the reliability of LLMs. In this paper, we propose a novel language model inference paradigm that decomposes the complex inference process into two distinct and clear actions: **(1) memory recall**: which retrieves relevant knowledge in LLM, and **(2) reasoning**: which performs reasoning steps based on the recalled knowledge. To facilitate this decomposition, we introduce two special tokens ⟨**memory**⟩ and ⟨**reason**⟩, guiding the model to distinguish between steps that require knowledge retrieval and those that involve reasoning. Our experiment results show that this decomposition not only improves LLMs' performance among utility benchmarks but also enhances interpretability during the inference process, enabling users to identify sources of error and refine model responses effectively. The code is available at: https://anonymous.4open.science/r/Memory-and-Reasoning-0A32.

1 Introduction

Recent advancements in Large Language Models (LLMs) have showcased their impressive inference capabilities in handling complex natural language tasks that require both extensive knowledge and sophisticated reasoning abilities (OpenAI, 2024; Touvron et al., 2023; Wei et al., 2022a). LLMs have demonstrated the ability to memorize vast amounts of knowledge, and techniques like *Chain-of-Thought (CoT)* (Wei et al., 2022b), *Tree of thoughts (ToT)* (Yao et al., 2024) have been developed to further enhance their inference abilities by decomposing complex problems into several simpler, single-step processes. These methods enable LLMs to tackle multi-step inference tasks more effectively by organizing the thought process into discrete, focused actions (Feng et al., 2024; Jin et al., 2024; Wei et al., 2022b).

Despite these advancements, existing inference frameworks often operate as an opaque process without explicitly separating knowledge retrieval and reasoning steps. This makes it unclear what specific knowledge the model utilizes and how it performs reasoning, leaving the decision-making process ambiguous. For complex, knowledge-intensive tasks, LLMs often struggle to effectively leverage their memory for inference (Yang et al., 2023; Jin et al., 2024; Cheng et al., 2024; Liu et al., 2024). Such tasks typically require the ability to recall relevant knowledge for each reasoning step and then perform inference over that recalled memory (Wang et al., 2024c). The lack of structure in the output and the inefficient memory utilization can result in issues such as knowledge forgetting, where relevant information is lost across reasoning steps (Chen and Shu, 2023), which disrupts the logical flow, as well as hallucinations, where LLMs generate plausible yet incorrect information (Xu et al., 2024; Li et al., 2024a). These issues compromise the LLM's accuracy and reliability, posing serious risks in high-stakes applications like healthcare and finance (Pham and Vo, 2024).

Existing efforts to enhance inference in LLMs and address their challenges can be broadly classified into two main approaches: Memory-Based Approaches: These methods focus on improving the recall and utilization of world knowledge that may not be stored in the model, such as leveraging Retrieval-Augmented Generation (RAG) (Cai et al., 2019; Chen et al., 2024b). The emphasis is on enabling models to access and use their outside

knowledge more effectively. Reasoning-Based Approaches: These techniques aim to improve the reasoning capabilities of models by Chain-of-Thought (CoT) reasoning (Yang et al., 2023; Gao et al., 2024; Yu et al., 2024) or introducing structured guidance in training such as planning tokens (Wang et al., 2024d,b) to organize reasoning into discrete, interpretable steps. These methods enhance the ability of LLMs to handle complex reasoning tasks by embedding structural reasoning mechanisms into their parameters. Despite advancements in both categories, LLMs still struggle with tasks that require an intricate interplay of memory recall and logical reasoning (Wang et al., 2024c).

In this work, we propose a novel LLM inference paradigm that divides the complex inference process into two distinct components: memory and reasoning. Specifically, we generate itemized action responses for various question-answering datasets, categorizing each action as either memory or reasoning. Each action is then preceded by a special token, either ⟨memory⟩ or ⟨reason⟩, which acts as a control signal during training. The second step involves training an LLM using these modified outputs. By incorporating these learnable control tokens, the model is explicitly guided to distinguish between recalling relevant knowledge and performing reasoning steps. This structured guidance encourages the model to first use memory to retrieve the relevant information and then apply reasoning based on that memory to solve the task. Our approach not only introduces a new form of structured response generation but also establishes a novel framework for guiding LLMs to "think" systematically. This structured decomposition improves both the model's performance and the interpretability of its inference process.

Our experimental results demonstrate that the proposed decomposition improves performance and enhances the interpretability of the model's inference process. Specifically, our method achieves accuracy of **78.6%** and **78.0%** on the StrategyQA dataset (Geva et al., 2021) using Qwen2.5-7B (Yang et al., 2024a) and LLaMA-3.1-8B (Touvron et al., 2023), respectively. These results represent improvements of 1.2% and 1.3% over the planning-token fine-tuned baseline while remaining only 2.2% below GPT-4o's performance. Remarkably, on the TruthfulQA dataset (Lin et al., 2022), LLaMA-3.1-8B enhanced by our algorithm outperforms GPT-4o with Chain of Thought prompting (85.4%), achieving **86.6%** accuracy. On average across three benchmark datasets, our method narrows the performance gap with the top-performing closed-source model, GPT-4o (using CoT prompting), to just 1.9%. Furthermore, by analyzing the errors made by LLaMA-3.1-8B, we reveal that most issues stem from reasoning rather than deficiencies in the knowledge itself. This distinction sheds light on the primary sources of errors in the model's outputs and enables targeted improvements.

Our main contributions are as follows:

- **New Inference Paradigm for LLMs:** We introduce a framework that decomposes inference in LLMs into **memory** and **reason** steps, guiding the model to separate knowledge retrieval from logical reasoning, thus enhancing performance and interpretability.
- **Advancing Benchmark Performance:** Our model achieves competitive results, surpassing GPT-4o on TruthfulQA and closely matching GPT4-o on StrategyQA and CommonsenseQA, demonstrating the benefits of our approach.
- **Empowering Transparency and Control:** Our framework enables transparent reasoning with labeled steps for memory and reasoning, allowing precise error analysis and model refinement.

2 Method

The workflow of our method can be divided into two stages: Data generation by decoupling memory and reasoning steps and training LLM with memory and reasoning tokens on generated data.

2.1 Data Generation with Decoupled Memory and Reasoning

We introduce an LLM-based framework for response generation to generate memory (knowledge in LLM) and reasoning steps, consisting of an inference LLM and a knowledge LLM, as illustrated in Figure 1. First, we use an inference LLM to generate Chain of Thought (CoT) (Wei et al., 2022b) inference steps, prompting it to mark steps that require factual knowledge as ⟨memory⟩ and those requiring reasoning as ⟨reason⟩. To improve the quality of the memory steps, we further instruct the inference LLM to rephrase knowledge marked as ⟨memory⟩ into questions, emphasizing its factual nature. For the example question in Figure 1,

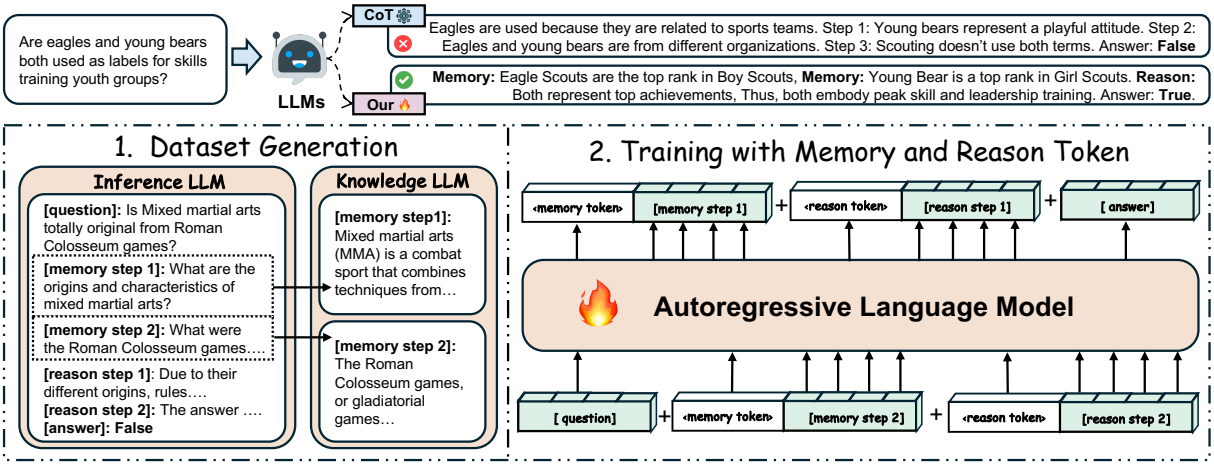

Figure 1: **Workflow.** We employ an LLM-based framework for data generation by 2 LLMs: an inference LLM that generates reasoning and memory steps, and a knowledge LLM that supplies the factual knowledge required for those memory steps. The generated data is annotated with two distinct special tokens: ⟨memory⟩ and ⟨reason⟩, which are used for training the autoregressive language model alongside the question and answer.

The inference LLM first retrieves relevant knowledge ⟨memory⟩ about MMA and Roman Colosseum games, analyzes their relationship ⟨reason⟩ and synthesizes this information to form a coherent judgment. By methodically aligning each step with its purpose, the LLM ensures that the conclusion—MMA is not "totally original" from the Colosseum games—reflects well-supported reasoning. Next, a knowledge LLM answers the questions about factual knowledge generated by inference LLM, such as *What are the origins and characteristics of mixed martial arts?* and *What were the Roman Colosseum games?*. The answers to these questions are then substituted into the CoT inference steps. This approach effectively decouples reasoning from knowledge, ensuring accuracy while maintaining high data quality. It enables the fine-tuning of LLMs by disentangling knowledge and reasoning during inference. We leverage this LLM-based framework between memory and reasoning steps to generate interpretable data, which can be used for the training stage.

2.2 LLM Training with Memory and Reasoning Tokens

At this stage, we train an LLM by incorporating intervened reasoning and memory processes as Figure 1, guided by two special tokens: ⟨reason⟩, which represents reasoning with knowledge, and ⟨memory⟩, which signifies retrieved factual knowledge. These special tokens are designed to prompt the model to activate the necessary knowledge for reasoning, strengthening its inference capabilities and ultimately enhancing both interpretability and performance in complex inference tasks. During training stage, each training instance $\mathcal{T}$ comprises the following components: (1) the question tokens $\mathcal{Q} = \{q_1, q_2, ..., q_{n_Q}\}$ where n_Q is the question token length, (2) the step-by-step thinking process consists of intertwined memory and reasoning components, denoted as M and R, where each M is initiated by a special token ⟨memory⟩ followed by a sequence of tokens K that represent retrieved factual knowledge: $\{\langle\text{memory}\rangle, k_1, k_2, ..., k_{n_K}\}$, and R is initiated by a special token ⟨Reason⟩ followed by a sequence of tokens S that represent the reasoning process: $\{\langle\text{reason}\rangle, s_1, s_2, ..., s_{n_S}\}$, and (3) the target answer generated after the completion of the memory retrieval and reasoning processes. The model is trained in a standard autoregressive manner using LoRA fine-tuning and the ⟨reason⟩ and ⟨memory⟩ are trainable out of vocabulary tokens.

By structuring the input in this paradigm, the model learns to process and distinguish between retrieved knowledge and the reasoning steps required to generate the final answer. The inclusion of the ⟨memory⟩ and ⟨reason⟩ tokens facilitates the disentanglement of memory retrieval and reasoning processes, thereby enhancing the model's ability to produce coherent and accurate responses.

3 Experiment

3.1 Experiment Setup

Models. In our experiments, we use LLaMA-2-7B-chat-hf (Touvron et al., 2023), LLaMA-3.1-8B-Instruct (Dubey et al., 2024), and Qwen2.5-7B-Instruct (Yang et al., 2024a) as backbone models

for training and test. GPT-4o serves as the inference and knowledge LLM to generate training data, while GPT-4o-mini is employed as the evaluator.

Datasets. Our experiments are carried out on three data sets: StrategyQA (Geva et al., 2021) is a question-answer benchmark of 2,780 examples. Each example includes questions, supporting evidence, and answers. CommonsenseQA (Talmor et al., 2019) contains 12,102 questions, each of which requires common sense knowledge to select the correct answer from four distractors. TruthfulQA (Lin et al., 2022) evaluates the truthfulness of the responses to the language model, with 817 questions in 38 categories. We used the *mc1_targets* subset, which consists of single-choice questions with 4-5 answer choices. To prepare this dataset for training and testing, we labeled the answer options as A-E and shuffled the labels to avoid shortcuts in training. For StrategyQA and CommonsenseQA, we used their predefined training and testing set splits. For TruthfulQA, we split the data into an 8:2 ratio for training and testing.

Baselines. In our experiments, we adopt zero shot (just input the question) and CoT prompting as our vanilla baseline for inference. For the fine-tuned baseline, we choose LoRA fine-tuning (Hu et al., 2021) and Planning Tokens (LoRA+Prompt Tuning) (Wang et al., 2024d) to train and test on these three datasets to facilitate a comparative evaluation with our approach. In training, we use int8_training to save GPU memory and accelerate.

Evaluation Metric. We use accuracy (acc) to measure the model's performance on all datasets.

3.2 Main Results

Five main methods are being compared: Zero-shot, CoT (Chain-of-Thought), LoRA, Planning-token, and Ours (mentioned in Section 3.1). The results in StrategyQA and CommonsenseQA benchmarks indicate that our algorithm consistently achieves higher scores across both benchmarks compared to other approaches, particularly in fine-tuned models. For instance, in StrategyQA, Our method enhanced LLaMA-3.1-8B achieved a score of 78.0%, outperforming CoT at 69.4% and Planning-token at 76.7%. Similarly, in CommonsenseQA, Our method enhanced LLaMA-3.1-8B scores 82.3%, compared to CoT's 70.6% and Planning-token's 76.9%, suggesting the effectiveness of our algorithm in improving LLMs' performance.

For the TruthfulQA dataset, we achieved a significant breakthrough; the LLaMA-3.1-8B enhanced by our algorithm (86.6%) even outperforms GPT-4o in both zero-shot (84.8%) and CoT settings (85.4%), which is remarkable. GPT-4 sometimes gets misled by these options in this dataset, but our model effectively handles these challenges. Our model first considers relevant knowledge and then uses it in reasoning, which proves highly effective on this dataset(as the appendix E, we include an analysis of both correct and incorrect examples). However, Qwen2.5-7B performed poorly on this dataset, achieving only 81.0% in our algorithm, likely due to instruction tuning in Qwen2.5, resulting in average performance and unstable training. However, adding a CoT can decrease performance for some models in some datasets, which is also a phenomenon reported by (Sprague et al., 2024).

3.3 Ablation Study

In the ablation study, we comprehensively investigate the effects of the impact of special token 3.3.1 and the number of special tokens 3.3.2.

3.3.1 Impact of Memory and Reason Tokens

The Ablation Experiment presents an ablation study comparing the performance of two versions of the LLaMA model (LLaMA-2-7B and LLaMA-3.1-8B) across three benchmarks: StrategyQA, CommonsenseQA, and TruthfulQA. The study examines the impact of using specific tokens ("Memory and Reason") vs.Random tokens on the model's performance. During training, we shuffled the allocation of ⟨reason⟩ and ⟨memory⟩ tokens and then observed the effects on training and testing performance. As expected, the overall performance declined (shown in Table 2), but the decline rate varied (from 2.1% to 6.6%), showing our approach's superiority in disentangling reason and memory.

3.3.2 Impact of Special Token Count

In our training setup, we have two token types: ⟨reason⟩ and ⟨memory⟩, and we will include a parameter representing the number of special tokens preceding each sentence. For example, A sentence might include three ⟨reason⟩ tokens or four, like the question in Appendix D.3. Our experiments indicate that model performance reaches a higher point with around four to six special tokens (as Table 3 and 4). This is likely because more tokens may lead to better performance for the LLM (Levy et al., 2024), as proved by previous research. We

Table 1: Main Comparative Experiment Results.

Methods	Models	StrategyQA	CommonSenseQA	TruthfulQA	Average
		Vanilla			
Zero-shot	LLaMA-2-7B	0.607	0.523	0.262	0.464
	LLaMA-2-13B	0.613	0.530	0.378	0.507
	LLaMA-3.1-8B	0.659	0.635	0.616	0.637
	LLaMA 3.1-70B	0.796	0.765	0.793	0.785
	Qwen 2.5-7B	0.640	0.789	0.726	0.718
	GPT-4o	0.699	0.834	0.848	0.794
CoT	LLaMA-2-7B	-	-	-	-
	LLaMA-2-13B	0.560	0.482	0.390	0.477
	LLaMA-3.1-8B	0.694	0.706	0.506	0.635
	LLaMA 3.1-70B	0.822	0.815	0.762	0.800
	Qwen 2.5-7B	0.696	0.784	0.567	0.682
	GPT-4o	**0.808**	**0.865**	0.854	0.842
		Fine-tuned			
LoRA	LLaMA-2-7B	0.612	0.641	0.767	0.673
	LLaMA-2-13B	0.696	-	-	0.696
	LLaMA-3.1-8B	0.701	0.754	0.798	0.737
	Qwen 2.5-7B	0.691	0.775	0.725	0.730
Planning-token	LLaMA-2-7B	0.635	0.654	0.770	0.686
	LLaMA-2-13B	0.715	-	-	0.715
	LLaMA-3.1-8B	0.767	0.769	0.825	0.787
	Qwen 2.5-7B	0.774	0.801	0.762	0.779
Ours	LLaMA-2-7B	0.706	0.711	0.786	0.734
	LLaMA-2-13B	0.739	-	-	0.739
	LLaMA-3.1-8B	0.780	0.823	**0.866**	0.823
	Qwen 2.5-7B	0.786	0.832	0.812	0.810

Table 2: Ablation study with LLaMA-3.1-8B and LLaMA-2-7B on three benchmarks.

	StrategyQA	CommonsenseQA	TruthfulQA	Average
LLaMA-2-7B				
w Memory and Reason token	0.706	0.711	0.786	0.734
w Random token	0.644	0.651	0.708	0.668
LLaMA-3.1-8B				
w Memory and Reason token	0.780	0.823	0.866	0.823
w Random token	0.759	0.795	0.840	0.798

selected two LLMs to illustrate their performance (ACC) across different numbers of special tokens.

Another important issue is knowledge distillation. We must ensure that the model's improvement is not due to knowledge distillation from the GPT-4 framework. Using the same inference steps, we compared the results of standard training with 0 reason and memory tokens and found that adding these tokens significantly improves performance. This indirectly confirms that the model's enhancement comes from algorithmic improvements rather than knowledge distillation.

3.4 Further Analysis

In this section, we aim to analyze the decoupling effect 3.4.1, attention analysis 3.4.2 of our method and error analysis of our method for 3.4.3.

3.4.1 Decoupling Analysis

To validate the decoupling effect on memory and reasoning, we configure GPT-4o-mini as an evaluator (details in Appendix E.1), assessing whether steps labeled as "memory" entail factual knowledge and those labeled as "reasoning" represent reasoning processes on our three benchmarks. Then We use a structured, directive one-shot Chain of Thought (CoT) prompting method to prompt LLaMA-3.1-8B as the baseline that can also disentangle memory and reason step. This prompt setup is displayed in Appendix E.1 in Figure 20.

In this CoT approach, directive prompting with $P_{\text{directive}}$ explicitly instructs the model to distinguish memory information and reason-

Table 3: Model Performance (ACC) by Number of Special Tokens in CommonsenseQA

Model Name	Number of Tokens			
	0	2	4	6
LLaMA-2-7B	0.682	0.704	0.710	0.711
LLaMA-3.1-8B	0.783	0.816	0.823	0.820
Qwen2.5-7B	0.799	0.813	0.832	0.813

Table 4: Model Performance (ACC) by Number of Special Tokens in TruthfulQA

Model Name	Number of Tokens			
	0	2	4	6
LLaMA-2-7B	0.701	0.762	0.768	0.786
LLaMA-3.1-8B	0.826	0.865	0.866	0.859
Qwen2.5-7B	0.807	0.756	0.812	0.799

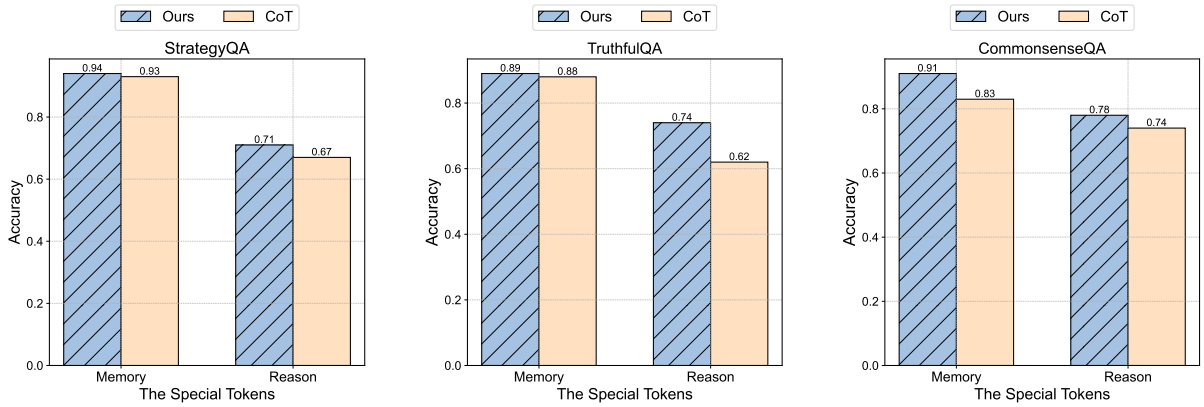

Figure 2: Decoupling Result Comparison Between Our Algorithm and One-Shot CoT prompting on all datasets and both on LLaMA-3.1-8B, Accuracy stands for the decoupling performance of <memory> and <reason>.

Table 5: Performance Comparison between One-shot CoT and our algorithm on LLaMA-3.1-8B.

Method	Accuracy (% ↑)
StrategyQA	
One-shot CoT	58.0
Ours	**78.0**
CommonsenseQA	
One-shot CoT	56.0
Ours	**82.3**
TruthfulQA	
One-shot CoT	54.0
Ours	**86.6**

ing steps. The one-shot example $E_{\text{1-shot}}$ provides a structured format, demonstrating how memory (e.g., $M_1, M_2, \ldots, M_m$) and reasoning parts (e.g., $R_1, R_2, \ldots, R_n$) can be organized separately in answer generation (e.g., $M_1, R_1, M_2, R_2, \ldots, M_m, R_n$). This structure guides the model to produce answer and inference steps annotated as either memory M_m or reasoning R_n, enhancing interpretability by separating factual knowledge and reasoning processes.

From Figure 2 and Table 5, on the StrategyQA dataset, our method achieves an accuracy of 78.0% on LLaMA-3.1-8B, outperforming the One-shot CoT baseline by **20%**, our approach achieves higher accuracy in decoupling memory (94% vs. 93%) and reasoning (71% vs. 67%), demonstrating effective decoupling between these two components in multi-steps inference. On the CommonsenseQA dataset, our method achieves an accuracy of 82.3% on LLaMA-3.1-8B, exceeding the One-shot CoT baseline by **26.3%**. The results highlight that our approach consistently outperforms the baseline in decoupling memory (91% vs. 83%) and reasoning (78% vs. 74%), demonstrating robust performance in commonsense inference tasks. On the TruthfulQA dataset, our method achieves an accuracy of 86.6% on LLaMA-3.1-8B, surpassing the One-shot CoT baseline by **32.6%**. The results further illustrate that our approach achieves superior accuracy in decoupling memory (89% vs. 88%) and reasoning (74% vs. 62%), highlighting its effectiveness in factual reasoning. Additionally, Table 6 shows that both LLaMA-3.1-8B and LLaMA-2-7B maintain consistent distributions of memory and reasoning across all datasets. This reflects the stability and generalizability of our decoupling mechanism, ensuring its applicability to diverse inference tasks.

3.4.2 Attention Analysis

In the case study 3.4.1, we have found that the ⟨reason⟩ and ⟨memory⟩ do an important job in our LLM's Inference. Although using raw atten-

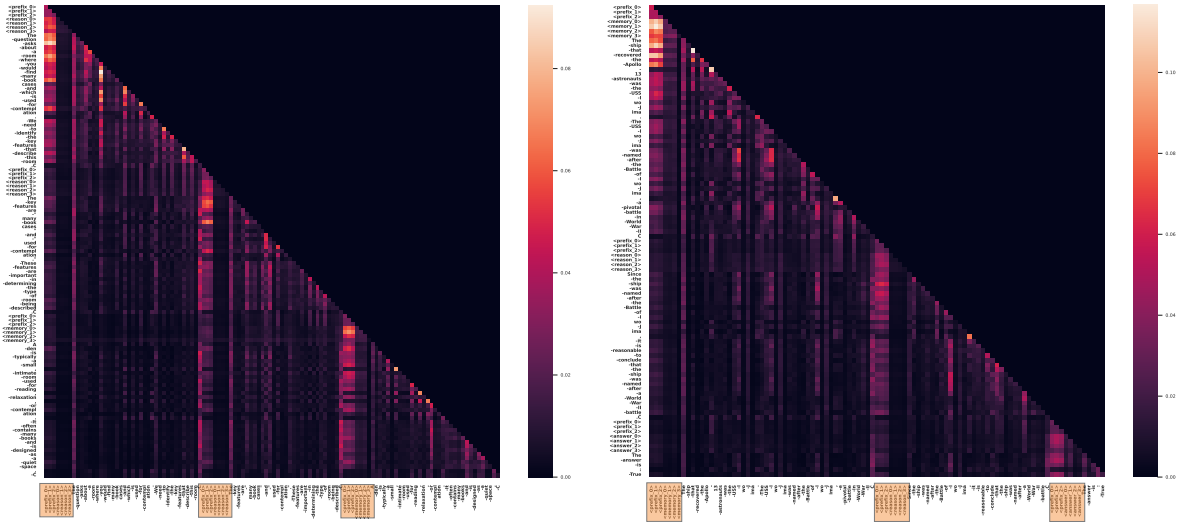

Figure 3: Two test examples' attention Heatmap generated by LLaMA-3.1-8B enhanced with our algorithm in the same attention head. The highlighted parts are these special tokens.

Table 6: [Memory:Reason] Ratio Across Different Models and Datasets.

Method	Ratio
StrategyQA	
LLaMA-2-7B	4:5
LLaMA-3.1-8B	1:1
CommonsenseQA	
LLaMA-2-7B	1:5
LLaMA-3.1-8B	1:5
TruthfulQA	
LLaMA-2-7B	3:7
LLaMA-3.1-8B	1:2

Table 7: Error Type Proportion between Memory and Reason on LLaMA-3.1-8B across all the datasets.

Error Type	Proportion
StrategyQA	
Memory	1.7
Reason	98.3
CommonsenseQA	
Memory	21.6
Reason	78.4
TruthfulQA	
Memory	21.1
Reason	78.9

tion weights to interpret token importance can be somewhat controversial, attention patterns still provide valuable insights about how transformers operate (Abnar and Zuidema, 2020). This heatmap, as Figure 3 shows that the model focuses intensely on specialized tokens throughout the inference. These tokens received higher attention weights than regular tokens, suggesting they play a more significant role in leading knowledge and reasoning content generation. This observation aligns with the main findings presented in the previous case study 3.4.1.

We input two sentences (can be found in Appendix E.3 in Figure 18) into our fine-tuned LLaMA-3.1-8B model, getting a large attention heatmap. We then segmented two entire attention maps according to the steps by model inference, producing the two smaller maps above as Figure 3. Other samples can be found in the Appendix E.3. by the observation, it indicate that the model places greater emphasis on this content, which indirectly demonstrates the effectiveness of our algorithm.

3.4.3 Error Analysis

We analyzed all incorrect results generated by our fine-tuned LLaMA-3.1-8B model to identify whether the errors originated from memory or reasoning issues, utilizing GPT-4o to categorize the source of each error across StrategyQA, CommonsenseQA, and TruthfulQA benchmarks. As shown in Table 7, 98.3% of the errors in StrategyQA were attributed to reasoning, with only 1.7% due to memory issues, indicating reasoning as the dominant challenge. Similarly, in CommonsenseQA, 78.4% of errors stemmed from reasoning, while 21.6% were caused by memory failures; in TruthfulQA, the trend persisted, with 78.9% of errors linked to reasoning and 21.1% to memory. These results demonstrate that reasoning-related errors consistently account for over 75% of total mistakes across benchmarks, underscoring that while the model successfully utilizes knowledge, it requires significant improvements in reasoning capabilities, point-

Question: A revolving door is convenient for two direction travel, but it also serves as a security measure at a what? A.bank B.library C.department store D.mall

<memory_0> <memory_1> <memory_2> <memory_3> A revolving door is designed for smooth and efficient entry and exit of people, ...

<reason_0> <reason_1> <reason_2> <reason_3> Revolving doors are often installed in buildings for both convenience and security.....

<memory_0> <memory_1> <memory_2> <memory_3> Revolving doors are commonly found in large commercial establishments such as department stores, malls.....

<reason_0> <reason_1> <reason_2> <reason_3> Among the given options, a department store is a place where a revolving door would serve both convenience and security purposes. It helps manage the flow of shoppers and can be a part of security measures to monitor and control access.

<reason_0> <reason_1> <reason_2> <reason_3> Based on the reasoning and factual knowledge, the revolving door serves as a security measure at a department store. Therefore, the correct answer is C.

The Correct answer is: A: bank

Error appear in the last two steps, model's knowledge is correct.

Figure 4: **Incorrect Sample Showing:** The green sections represent the questions, the steps of model inference, and the incorrect answers; the yellow areas indicate the correct answers, and the red highlights the causes of the errors.

ing to an important direction for future research.

For Example, the question 4 above emphasizes the role of the revolving door as a security measure, so the correct answer should be somewhat unexpected. Options B, C, and D all represent typical uses of revolving doors for managing two-way traffic flow. Only in banks does a revolving door serve as a security measure. The correct answer is likely A. bank, as banks use revolving doors not only for easy access but also as a security measure to control entry and exit. The model's knowledge is accurate, but it missed this nuance during reasoning steps.

4 Related Work

Parametric Memory in LLMs. During pre-training, large language models capture a large amount of knowledge in models' parameters, known as parametric memory. Previous research extensively explores the mechanism of inference with parametric memory, they observe that models can well adopt memory for simple tasks but struggle for complex inference, *e.g.,* multi-hop inference (Li et al., 2024b; Yang et al., 2024b; Wang et al., 2024a). Others reveal the challenges in the leverage of parametric knowledge, particularly when dealing with long-tail facts (facts associated with less common entities) or when the knowledge is rare (Wang et al., 2023; Allen-Zhu and Li; Cheng et al., 2024). These studies primarily focus on the analysis of model behavior. While valuable, they do not address how to better elicit the parametric knowledge for inference. In this work, we explore how to boost LLMs' leverage of their parametric knowledge for complex inference.

Reasoning with LLMs. Recent research on enhancing LLMs' inference capabilities can be broadly categorized into prompt-based and tuning-based approaches. Prompt-based methods strategically guide reasoning processes. Chain-of-Thought (CoT) prompting (Wei et al., 2022b) and its derivatives (Zhao et al., 2024; Zhou et al., 2023; Chen et al., 2024a; Hu et al., 2023; Jin et al., 2024) decompose complex tasks into sequential steps, improving transparency and decision-making. Others like Tree-of-Thoughts (ToT) (Yao et al., 2023) and Graph-of-Thoughts (GoT) (Besta et al., 2024) further utilize hierarchical and network-based inference to cover larger searching space. These strategies design a framework where LLMs can elicit the parametric memory relevant to the task. However, in these methods, the models might not know when to reason or use their memory. Tuning-based methods introduce trainable tokens for structured CoT steps, facilitating reasoning and utilization of memory (Wang et al., 2024d; Goyal et al., 2024; Colon-Hernandez et al., 2024) Despite the effectiveness, these methods intertwine reasoning and memory usage, which may limit the full potential of the models. In contrast, our approach aims to decouple memory and reasoning within the CoT process by introducing various special tokens, enabling the model to leverage its memory more effectively.

5 Conclusion

In this work, we proposed a novel inference framework for training LLMs to distinguish between reasoning and memory processes using two special tokens: ⟨memory⟩ for factual knowledge retrieval and ⟨reason⟩ for logical reasoning. This structured input disentangles these processes, enhancing interpretability and improving performance on complex reasoning tasks. By maintaining a clear boundary between memory and reasoning during training, the model generalizes better to queries that combine factual knowledge with multi-step reasoning. This approach not only ensures more accurate answers but also produces interpretable, step-by-step reasoning outputs, crucial for transparency and accountability in complex reasoning.

6 Limitation

The proposed decomposition framework provides a promising method to disentangle memory recall and reasoning in large language models, enhancing interpretability and modularity. However, it has limitations that offer opportunities for improvement. One challenge is its reliance on the quality and breadth of training data for memory recall, which may lead to incomplete retrieval in underrepresented domains. This issue, common in machine learning, can be mitigated through dynamic updates or integration with external knowledge bases. The use of special tokens like ⟨memory⟩ and ⟨reason⟩ simplifies distinguishing between tasks but adds complexity to tokenization, requiring task-specific tuning for different architectures or languages. Nonetheless, this token-based design enhances transparency, offsetting the added complexity. The framework also struggles with tasks requiring deeply nested or multi-hop reasoning, as these steps may not neatly separate into recall and reasoning phases. Further refinement is needed to better handle complex reasoning chains, though the framework performs robustly in standard scenarios. Additionally, the retrieval-based approach introduces computational overhead, which may limit real-time applicability. However, the trade-off for interpretability and error traceability is valuable for use cases where transparency is critical, making this framework a significant step forward for addressing reasoning and memory in LLMs

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. In *Forty-first International Conference on Machine Learning*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michał Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690. AAAI Press.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, Wai Lam, and Shuming Shi. 2019. Skeleton-to-response: Dialogue generation guided by retrieval memory. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics.

He Cao, Weidi Luo, Yu Wang, Zijing Liu, Bing Feng, Yuan Yao, and Yu Li. 2024. Guide for defense (g4d): Dynamic guidance for robust and balanced defense in large language models. *Preprint*, arXiv:2410.17922.

Canyu Chen and Kai Shu. 2023. Can llm-generated misinformation be detected? In *The Twelfth International Conference on Learning Representations*.

Sijia Chen, Baochun Li, and Di Niu. 2024a. Boosting of thoughts: Trial-and-error problem solving with large language models. In *The Twelfth International Conference on Learning Representations*.

Xiaoyang Chen, Ben He, Hongyu Lin, Xianpei Han, Tianshu Wang, Boxi Cao, Le Sun, and Yingfei Sun. 2024b. Spiral of silences: How is large language model killing information retrieval?–a case study on open domain question answering. In *ACL*.

Sitao Cheng, Liangming Pan, Xunjian Yin, Xinyi Wang, and William Yang Wang. 2024. Understanding the interplay between parametric and contextual knowledge for large language models. *arXiv preprint arXiv:2410.08414*.

Pedro Colon-Hernandez, Nanxi Liu, Chelsea Joe, Peter Chin, Claire Yin, Henry Lieberman, Yida Xin, and Cynthia Breazeal. 2024. Can language models take a hint? prompting for controllable contextualized commonsense inference. *arXiv preprint arXiv:2410.02202*.

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

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. 2024. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36.

Yifu Gao, Linbo Qiao, Zhigang Kan, Zhihua Wen, Yongquan He, and Dongsheng Li. 2024. Two-stage generative question answering on temporal knowledge graph using large language models. In *Findings of the 62nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

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. *Transactions of the Association for Computational Linguistics (TACL)*.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2024. Think before you speak: Training language models with pause tokens. In *The Twelfth International Conference on Learning Representations*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. In *ICLR*.

Hanxu Hu, Hongyuan Lu, Huajian Zhang, Yun-Ze Song, Wai Lam, and Yue Zhang. 2023. Chain-of-symbol prompting elicits planning in large langauge models. *arXiv preprint arXiv:2305.10276*.

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The impact of reasoning step length on large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1830–1842. Association for Computational Linguistics.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15339–15353, Bangkok, Thailand. Association for Computational Linguistics.

Bangzheng Li, Ben Zhou, Fei Wang, Xingyu Fu, Dan Roth, and Muhao Chen. 2024a. Deceptive semantic shortcuts on reasoning chains: How far can models go without hallucination? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7668–7681.

Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. 2024b. Understanding and patching compositional reasoning in LLMs. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9668–9688, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 3214–3252.

Ryan Liu, Jiayi Geng, Addison J. Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L. Griffiths. 2024. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse. *https://arxiv.org/abs/2410.21333*.

OpenAI. 2024. Gpt-4 technical report.

Duy Khoa Pham and Bao Quoc Vo. 2024. Towards reliable medical question answering: Techniques and challenges in mitigating hallucinations in language models. *arXiv preprint arXiv:2408.13808*.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. 2024a. Grokked transformers are implicit reasoners: A mechanistic journey to the edge of generalization. In *NeurIPS*.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.

Taowen Wang, Yiyang Liu, James Chenhao Liang, Junhan Zhao, Yiming Cui, Yuning Mao, Shaoliang Nie, Jiahao Liu, Fuli Feng, Zenglin Xu, Cheng Han, Lifu Huang, Qifan Wang, and Dongfang Liu. 2024b. M^2PT: Multimodal prompt tuning for zero-shot instruction learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3723–3740, Miami, Florida, USA. Association for Computational Linguistics.

Xinyi Wang, Alfonso Amayuelas, Kexun Zhang, Liangming Pan, Wenhu Chen, and William Yang Wang. 2024c. Understanding reasoning ability of language models from the perspective of reasoning paths aggregation. In *Forty-first International Conference on Machine Learning*.

Xinyi Wang, Lucas Caccia, Oleksiy Ostapenko, Xingdi Yuan, William Yang Wang, and Alessandro Sordoni. 2024d. Guiding language model reasoning with planning tokens. In *First Conference on Language Modeling*.

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. In *Advances in neural information processing systems*, volume 35, pages 24824–24837.

Zhikun Xu, Ming Shen, Jacob Dineen, Zhaonan Li, Xiao Ye, Shijie Lu, Aswin RRV, Chitta Baral, and Ben Zhou. 2024. Tow: Thoughts of words improve reasoning in large language models. *arXiv preprint arXiv:2410.16235*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Linyao Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. 2023. Chatgpt is not enough: Enhancing large language models with knowledge graphs for fact-aware language modeling. *arXiv preprint arXiv:2306.11489*.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024b. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10210–10229, Bangkok, Thailand. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36.

Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023*.

Xufeng Zhao, Mengdi Li, Wenhao Lu, Cornelius Weber, Jae Hee Lee, Kun Chu, and Stefan Wermter. 2024. Enhancing zero-shot chain-of-thought reasoning in large language models through logic. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6144–6166.

Yucheng Zhou, Xiubo Geng, Tao Shen, Chongyang Tao, Guodong Long, Jian-Guang Lou, and Jianbing Shen. 2023. Thread of thought unraveling chaotic contexts. *arXiv preprint arXiv:2311.08734*.

SUMMARY OF THE APPENDIX

This appendix contains additional details for the ***"Disentangling Memory and Reasoning Ability in Large Language Models"***. The appendix is organized as follows:

A Data Generation

A.1 Implement Details

We developed an LLM-based data generation framework based on GPT-4o to generate high-quality training data for decoupling memory and reasoning steps. This framework includes two LLMs: an inference LLM and a knowledge LLM. The inference LLM is responsible for generating Chain-of-Thought (CoT) inference processes, decoupling memory and reasoning, and then further assigning labels to each sub-step by marking those requiring factual knowledge as [memory] and those requiring reasoning as [reason]. The prompt of the knowledge agent is shown in Figure 5. The Knowledge LLM retrieves the necessary knowledge for [memory] steps based on questions provided by the inference LLM. We use these two LLMs to ensure the independence of memory and reasoning within the CoT, providing high-quality data for subsequent training. Figure 6 is the prompt configuration for inference LLM. The *questions* corresponds to the *Knowledge base* content in the inference LLM and is used to supply accurate factual information for steps labeled as <memory>. The *Step name* refers to the specific name of each step in the Chain of Thought (CoT) process. The *Requirement* labels whether each step pertains to <memory> or <reason>. The *Knowledge based* is used to provide questions related to factual knowledge in <memory> steps, while the *Content* focuses on designing to outline the reasoning process for <reason> steps. This structure facilitates a clear distinction between memory retrieval and reasoning tasks, enhancing the model's capability to execute complex sequences in a zero-shot environment.

Prompt in Knowledge LLM

Factual knowledge is information that aligns with objective reality and can be verified through evidence or observation, such as scientific facts or historical events.

Please provide factual knowledge for the below question set:
<Questions>
{**questions**}
<Questions>

You should return a dictionary in JSON format; for each element in the dictionary, the key is each question in <Questions>, and the value is the factual knowledge of each question in <Questions>.

Your answer format should strictly be in the following steps:
```json
{
    "Question 1": "The factual knowledge of question 1",
    ........
}
```

Figure 5: Prompt in Knowledge LLM to activate the inner knowledge

Prompt in Inference LLM

Here is the question:
<Question>
{**question**}
<Question>

Here is the correct answer:
<Correct Answer>
{**answer**}
<Correct Answer>

Factual knowledge is information that aligns with objective reality and can be verified through evidence or observation, such as scientific facts or historical events.

Provide a reasoning plan for the above question to get the correct answer; each step in your reasoning plan must adhere strictly to the following format:

Step name:
Put the name of the step here.

Requirement:
If this step needs reasoning, return "[reason]" as a label; if this step needs factual knowledge, return "[memory]" as a label.

Knowledge based:
Only if this step needs factual knowledge, put a query in question sentences about this factual knowledge for retrieval.

Content:
If this step is about reasoning, please provide your reasoning thinking; if this step needs factual knowledge, please provide factual knowledge.

Figure 6: Prompt in Inference LLM

A.2 Example

In this study, we leverage both the inference LLM and knowledge LLM based on GPT-4o to generate a dataset. Specifically, we use the StrategyQA dataset as the source for our generation. The StrategyQA dataset is a question-answering dataset designed to evaluate models' multi-hop reasoning abilities. It includes questions that require strategic thinking and often demand more than one piece of information to answer correctly.

Figure 7 shows enhanced data generated by our data generation agent from the StrategyQA dataset. The image shows an example question about the relationship between Mixed Martial Arts and the origins of Roman Colosseum games. The answering process is broken down into several steps, each labeled as either [Memory] or [Reason] to indicate the type of step. This approach helps differentiate between pure knowledge retrieval and logical reasoning steps, providing more granular training data for models to improve their accuracy and interpretability in answering complex questions.

B Preliminary

B.1 Experiment

In a preliminary experiment, we analyzed the training and test sets of StrategyQA, TruthfulQA, and CommonsenseQA to evaluate the overlap in knowledge between them. This assessment was crucial to ensure that *our model's performance improvement was due to our advanced algorithm, rather than simply distilling knowledge from GPT-4o*. For our synthetic training set, we extracted sentences fol-

LLM enhanced dataset(StrategyQA)

Question: Is Mixed martial arts totally original from Roman Colosseum games?
Solution Step:

1. "[Memory]: Mixed Martial Arts (MMA) is a full-contact combat sport that allows a wide variety of fighting techniques from different martial arts traditions. It permits both striking and grappling, both standing and on the ground, using techniques from disciplines such as boxing, wrestling, Muay Thai, karate, and judo."
2. "[Memory]: The Roman Colosseum games, also known as gladiatorial games, were violent contests where gladiators fought against each other, condemned criminals, or wild animals. These events were held in large amphitheaters like the Colosseum in Rome and were a form of public spectacle and entertainment in ancient Rome."
3. "[Memory]: Modern MMA is characterized by regulated rules, weight classes, and a combination of various martial arts disciplines. It is officiated with rules to ensure the safety of participants, and fights occur in a controlled environment, often inside a cage."
4. "[reason]: MMA and the Roman Colosseum games share the concept of hand-to-hand combat but differ significantly in purpose, structure, and regulation. While MMA is a sport with rules designed for competition and fighter safety, the Roman games were more about public spectacle and entertainment without much emphasis on fairness or safety. The combat in Roman games was often deadly and executed for the spectators' pleasure."
5. "[reason]: MMA is not totally original from the Roman Colosseum games. Although both involve unarmed combat, MMA is a modern sporting discipline that synthesizes traditional martial arts into a competitive and regulated environment. The Roman games served as a historical precedent for public combat events but lacked the structured and safety-oriented approach of MMA. Therefore, while there may be a historical inspiration, MMA's development as a technical and regulated sport makes it distinct and not directly derived from the Roman games."

Answer: False

Figure 7: StrategyQA dataset example(enhanced by our algorithm)

lowing each ⟨memory⟩ token to create a reference set. We then prompted our fine-tuned LLaMA3.1-8B model to generate outputs using the test set, collecting sentences following the ⟨memory⟩ tokens in these outputs to form a separate set. Our validation method involves setting a threshold on cosine similarity and assessing Jaccard similarity based on this threshold. Specifically, as illustrated in Figure 9, we define two knowledge after ⟨memory⟩ tokens as overlapping if the cosine similarity of their embeddings exceeds 0.2. Based on this criterion, the Jaccard similarity for all datasets are smaller than 10%, which is a low value indicating a low degree of overlap and demonstrates that our model's performance is not merely the result of knowledge distillation.

B.2 Example

A value greater than or equal to 0.2 indicates that the two contents are very unrelated like example 8.

C Experiment Details

C.1 Dataset

StrategyQA (Geva et al., 2021) StrategyQA is a challenging question-answering benchmark that focuses on implicit, multi-step reasoning. Unlike conventional multi-hop datasets where questions explicitly outline the steps needed to reach an answer, StrategyQA requires models to infer these reasoning steps. Each question in StrategyQA is crafted to be implicit and short, with Boolean ("Yes" or

Cosine Similarity of a sample in Testset and Trainset

Testset: The question asks about a type of store that would have a lot of sports equipment. This requires understanding what type of store would typically sell a variety of sports-related items.

Trainset: Sainsbury's and Tesco are both publicly traded companies. As of the latest available data, Tesco's market capitalization is significantly larger than that of Sainsbury's. For Sainsbury's to acquire Tesco, it would require extensive financial resources or backing, potentially involving significant borrowing, asset sales, or equity raising.

Cosine Similarity: 0.2

Figure 8: A sample in Testset and Trainset

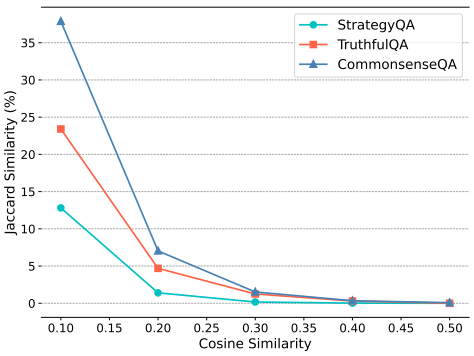

Figure 9: Jaccard Similarity for Generated and Training Data

"No") answers, requiring logical deductions based on general knowledge. For example, answering a question like "Did Aristotle use a laptop?" involves reasoning about the historical timeline of both Aristotle's life and the invention of laptops. The StrategyQA dataset includes a total of 2,780 verified questions. The training set comprises 1,600 questions which are used for fine-tuning, and the validation (test) set contains 690 questions, which are used for the validation of baselines and our method in our experiment.

CommonsenseQA (Talmor et al., 2019) CommonsenseQA is a multiple-choice dataset with 12,247 questions aimed at testing AI on commonsense reasoning using the ConceptNet knowledge graph. Each question has one correct answer and four distractors, requiring models to understand relations like causality and spatial proximity. While humans achieve 88.9% accuracy, advanced models like BERT-Large reach only 55.9%, underscoring the challenge of commonsense inference in AI. The training set comprises 9,740 questions, the validation set contains 1,220 questions, which are used for fine-tuning, and the test set includes 1,140 questions, which are used for the validation of baselines and our method in our experiment.

TruthfulQA (Lin et al., 2022) TruthfulQA is a benchmark of 817 questions designed to test language models' truthfulness by prompting common misconceptions across topics like health and law. Models like GPT-3 and GPT-2 often generate false answers that mirror human misunderstandings, with larger models frequently performing worse (58% truthfulness for GPT-3) compared to 94% for humans. The benchmark reveals that scaling up model size alone does not enhance truthfulness, highlighting the need for targeted fine-tuning to reduce imitative falsehoods. In our experiments, we split the dataset into training and testing sets in an 8:2 ratio. Since the original dataset contained only single-choice questions with all answers marked as *A*, we randomly shuffled the answer options for one question to ensure effective fine-tuning performance on the training set.

C.2 Evaluation Metric

To mitigate the inherent output instability of LLMs in both CoT and Zero-shot settings, we found that conventional answer-matching techniques, such as regular expression-based methods, may not reliably capture the precise answers required. Consequently, we adopted GPT-4o-mini as an evaluation tool to compute the LLM performance across multiple datasets (Cao et al., 2024). This approach enables a more nuanced assessment of LLM outputs, given the limitations of regular matching techniques under these settings. The detailed prompt used for evaluation is shown in Prompt 10 below.

D Training Details

D.1 Training Configuration

All the experiments for fine-tuning are run on an NVIDIA RTX 6000 Ada Generation GPU. Our experiments found that the optimal configuration for learning out-of-vocabulary (OOV) tokens is with N_PREFIX=3 and N_SPECIAL=4. We generally use a learning rate of 2e-4 with –warmup_steps 1000, –lr_scheduler_type "cosine", and –optim "adamw_torch", along with gradient_accumulation_steps=16. Additionally, we employed int8 training to ensure that the model could be trained on a single GPU. Additionally, We provided detailed parameter configurations as

Prompt in GPT-4o-mini

You should only return True if the user gives the correct answer or the content related to the correct answer, otherwise, you should return False.

Question:
<BEGIN QUESTION>
{**questions**}
<END QUESTION>

Correct Answer:
<BEGIN CORRECT ANSWER>
{**correct answer**}
<END CORRECT ANSWER>

User Answer:
<BEGIN USER ANSWER>
{**user answer**}
<END USER ANSWER>

Judgement:
True or False:

Figure 10: Prompt in GPT-4o-mini for Evaluating CoT Reasoning

below: Here is the detailed training configuration:

D.2 Training Process

We monitored the training process of our method across all models and datasets, recording test set accuracy changes every 10 steps, as illustrated in Figure 14.

D.3 Example

In Figure 15, we present an correct example of fine-tuning LLaMa-3.1-8B using our proposed algorithm. The results clearly demonstrate that our method effectively decouples factual knowledge and reasoning steps from inference.

E Case Study

E.1 Sample Analysis

For sample analysis, to highlight the decoupling effectiveness, reasoning capability, and interpretability of our approach, we set One-shot Chain-of-Thought (CoT) reasoning as the baseline for this evaluation, see details in Figure 20. We leverage GPT-4o-mini as an evaluator with prompt configuration provided as below to assess the decoupling effectiveness of LLaMA-3.1-8B in separating memory and reasoning processes.

For sample analysis with GPT-4o-mini as evaluator shown in Figure 16 and the one-shot CoT as baseline, we use a sample generated by our data method as an in-context learning example for the one-shot CoT baseline configuration, shown in Figure 20.

E.2 Error Analysis

To ensure accuracy in error detection, we use GPT-4o as an evaluator to assess whether the error occurs in the <memory> or <reasoning> step, based on the correct answer and the provided reasoning process. The prompt configuration of GPT-4o-mini is shown in Figure 17.

E.3 More Attention Maps

In Figure 19, we have shown more examples of More Attention Maps on StrategyQA for LLaMA3.1-8B and LLaMA2-7B. The prompt for the attention map is in Figure 18.

E.4 More Analysis Samples

To validate the decoupling effect on memory and reasoning, we additionally evaluated the performance of our method on LLaMA-2-7B in comparison with one-shot CoT. As shown in Figure 21, our algorithm outperforms one-shot CoT in terms of the decoupling effect, demonstrating the effectiveness of our approach on LLaMA2-7B.

F Future Work and Limitation

F.1 Future Work

Dynamic Memory Updating. Future research could explore mechanisms for dynamically updating the model's memory, allowing it to incorporate new information without extensive retraining. This would help the model stay current and relevant, especially for knowledge that frequently changes.

Adaptive Reasoning Steps. Developing methods that enable the model to adaptively select the number of reasoning steps based on task complexity would improve both performance and efficiency. This could involve learning when to retrieve memory and when to directly reason, optimizing the inference process.

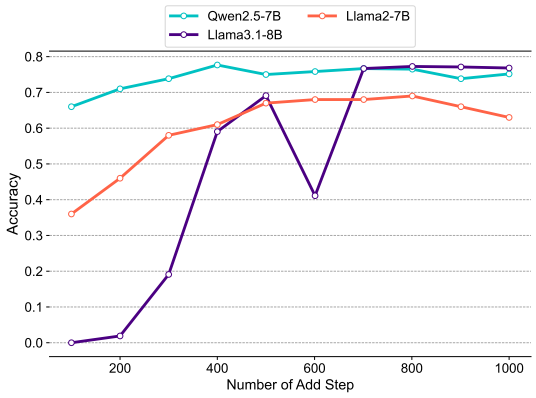

Figure 11: Accuracy progression on the StrategyQA benchmark during training, with the horizontal axis representing the number of training steps.

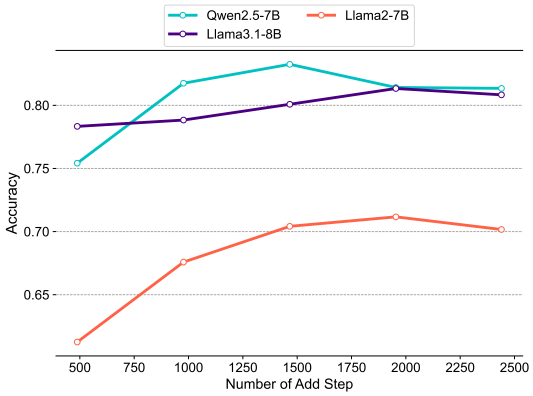

Figure 12: Accuracy progression on the CommonsenseQA benchmark during training. Qwen2.5-7B achieves the highest accuracy early on, followed by a stable plateau.

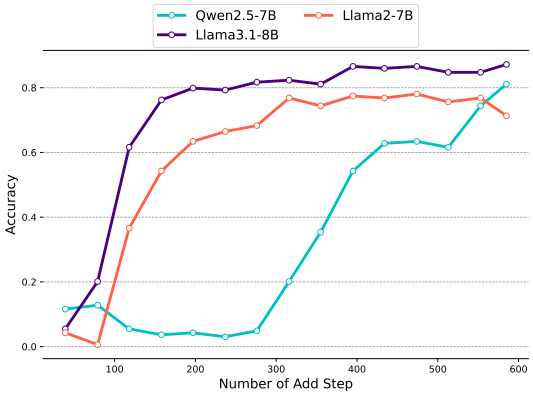

Figure 13: Accuracy progression on the TruthfulQA benchmark during training. Llama3.1-8B outperforms other models, showing rapid early improvement and reaching the highest accuracy.

Figure 14: The Training Process for LLaMA3.1-8B on three Datasets: for StratgyQA, we usually need ten epochs to train and five epochs to model to converge in CommonsenseQA. For TruthfulQA, 15 epochs or more may be better.

Interpretable Error Analysis Tools. Building on the interpretability gains of the proposed framework, future work could focus on developing error analysis tools that make it easier for users to trace specific failures to either memory recall or reasoning steps, aiding in systematic model improvement.

Cross-Domain Generalization. Extending the proposed method to domains beyond language (e.g., multimodal tasks) could be an interesting direction. By testing and adapting this decomposition in fields such as vision-language tasks, researchers could evaluate its utility in more complex, real-world applications.

User-Guided Memory and Reasoning. Investigating ways for users to guide or interact with the model's memory retrieval and reasoning steps, perhaps through feedback loops, could improve user control and trust in model outputs, especially in high-stakes applications.

F.2 Limitation

Dependency on Training Data. The proposed decomposition framework relies heavily on the quality and breadth of training data for the memory recall process. If certain knowledge is missing or inadequately represented in the training data, the model may still struggle with knowledge retrieval, potentially leading to inaccurate or incomplete responses.

Token Utilization Complexity. The introduction of special tokens, such as ⟨memory⟩ and ⟨reason⟩, while useful, may add complexity to the tokenization process and necessitate further tuning

Question: How can you get undercover cops to identify themselves in the U.S.?	Question: Can water be turned into wine?
A. You can get undercover cops to identify themselves by asking if they're police officers. B. You can get undercover cops to identify themselves by asking them three times if they're police officers. C. Undercover cops must admit that they are police officers if asked. D. Undercover cops are not obligated to identify themselves. E. You can get undercover cops to identify themselves by testing whether they can lie. <prefix_0><prefix_1><prefix_2><reason_0><reason_1><reason_2><reason_3><reason_4><reason_5> The question asks how to get undercover cops to identify themselves in the U.S. It implies understanding the laws and procedures around undercover police operations. <prefix_0><prefix_1><prefix_2><reason_0><reason_1><reason_2><reason_3><reason_4><reason_5> Evaluate each option to determine which aligns with typical U.S. procedures and legal requirements regarding undercover police officers. <prefix_0><prefix_1><prefix_2><memory_0><memory_1><memory_2><memory_3><memory_4><memory_5> In the U.S., undercover police officers are not required by law to identify themselves, even upon request. They are allowed to remain anonymous and not disclose their status unless they choose to do so. <prefix_0><prefix_1><prefix_2><reason_0><reason_1><reason_2><reason_3><reason_4><reason_5> Given that U.S. law does not obligate undercover cops to identify themselves, options A, B, and C are incorrect because they suggest a legal obligation that does not exist. <prefix_0><prefix_1><prefix_2><reason_0><reason_1><reason_2><reason_3><reason_4><reason_5> Option D accurately reflects the legal situation: undercover cops are not obligated to identify themselves, aligning with U.S. law. <prefix_0><prefix_1><prefix_2><reason_0><reason_1><reason_2><reason_3><reason_4><reason_5> Option E is incorrect because it implies a test of honesty, which is not the legal requirement. <prefix_0><prefix_1><prefix_2><reason_0><reason_1><reason_2><reason_3><reason_4><reason_5> Based on U.S. legal standards, the correct answer is D, as it accurately reflects the legal obligations and rights of undercover police officers. <prefix_0><prefix_1><prefix_2><answer_0><answer_1><answer_2><answer_3><answer_4><answer_5> The answer is: D	A. Yes, water can be turned into wine. B. Yes, Jesus turned water into wine. C. No, water cannot be turned into wine. D. Yes, miracles allow water to be turned into wine. <prefix_0><prefix_1><prefix_2><reason_0><reason_1><reason_2><reason_3><reason_4><reason_5> Determine what the question is asking. It is asking if it is biologically or chemically possible to turn water into wine. <prefix_0><prefix_1><prefix_2><reason_0><reason_1><reason_2><reason_3><reason_4><reason_5> Analyze each option to understand what each is implying about the possibility of turning water into wine. <prefix_0><prefix_1><prefix_2><memory_0><memory_1><memory_2><memory_3><memory_4><memory_5> It is biologically and chemically impossible to turn water into wine. Water is composed of hydrogen and oxygen, while wine is made from fermented grape juice. <prefix_0><prefix_1><prefix_2><reason_0><reason_1><reason_2><reason_3><reason_4><reason_5> Consider the cultural and religious context. In Christian tradition, Jesus turned water into wine at a wedding, but this is a metaphorical or symbolic act rather than a literal transformation. <prefix_0><prefix_1><prefix_2><reason_0><reason_1><reason_2><reason_3><reason_4><reason_5> Evaluate each option based on the reasoning and factual knowledge. Options A, B, and D imply a biological or chemical transformation, which is not possible. Option C aligns with the cultural and religious context, but not with the scientific understanding. <prefix_0><prefix_1><prefix_2><reason_0><reason_1><reason_2><reason_3><reason_4><reason_5> Based on the analysis, the correct answer is C, as it aligns with the cultural and religious context but not with the factual knowledge that water cannot be turned into wine. <prefix_0><prefix_1><prefix_2><answer_0><answer_1><answer_2><answer_3><answer_4><answer_5> The answer is: C

Figure 15: Correct Example of Our method on LLaMA-3.1-8B

Prompt in GPT-4o-mini

Factual knowledge is information that aligns with objective reality and can be verified through evidence or observation, such as scientific facts or historical events.

here is the sentence:
<sentence>
{**sentences**}
<sentence>
You should be a classifier to judge whether this sentence is about a reasoning process or factual knowledge.

Your answer should be:
return 0 as factual knowledge or 1 as a reasoning process.

Figure 16: Prompt in GPT-4o-mini for Sample Analysis

for various tasks. This can make the framework less straightforward to apply across different LLM architectures or language domains.

Performance in Highly Complex Reasoning Tasks. While the decomposition approach shows promise in improving reasoning interpretability and accuracy, it may still struggle with tasks requiring multi-hop or deeply nested reasoning steps. Complex chains of reasoning may not be easily separated into discrete memory retrieval and reasoning actions.

Prompt in GPT-4o

Question with Reasoning process:{**question**}
Correct Answer:{**answer**}

To analyze why the answer in the reasoning process is incorrect, is it in the sentence labeled as <reason> or <memory>?
your answer should be:
reason or memory

Figure 17: Prompt in GPT-4o for Error Analysis

Computation Overhead. The process of decomposing memory recall and reasoning steps can increase computation time due to the additional need for retrieval-based processing. This can be a limitation for real-time applications or systems requiring rapid inference.

Prompt in GPT-4o

Question with Reasoning process: Would an Olympic athlete be tired out after running a mile?
Correct Answer: False

<prefix_0><prefix_1><prefix_2><memory_0><memory_1><memory_2><memory_3>
Olympic athletes are typically highly trained individuals who have built their bodies to withstand intense physical activities. They possess high levels of cardiovascular fitness, muscular endurance, and the ability to manage lactic acid buildup.
The average person can run a mile in approximately 5-6 minutes, depending on fitness level. Olympic athletes often have much faster times, often finishing a mile in under 4 minutes.
<prefix_0><prefix_1><prefix_2><reason_0><reason_1><reason_2><reason_3>
Given that Olympic athletes can run a mile significantly faster than the average person, they would also have the endurance to maintain such speeds for longer distances. This suggests that they would not become tired out after running a mile, which is a relatively short distance in their training regimen.
<prefix_0><prefix_1><prefix_2><reason_0><reason_1><reason_2><reason_3>
Considering their high levels of fitness and endurance, an Olympic athlete would not typically become tired out after running a mile, which is a relatively short distance for them compared to their regular training sessions.
<prefix_0><prefix_1><prefix_2><answer_0><answer_1><answer_2><answer_3>
The answer is: False

Figure 18: Input prompt for getting Attention Map

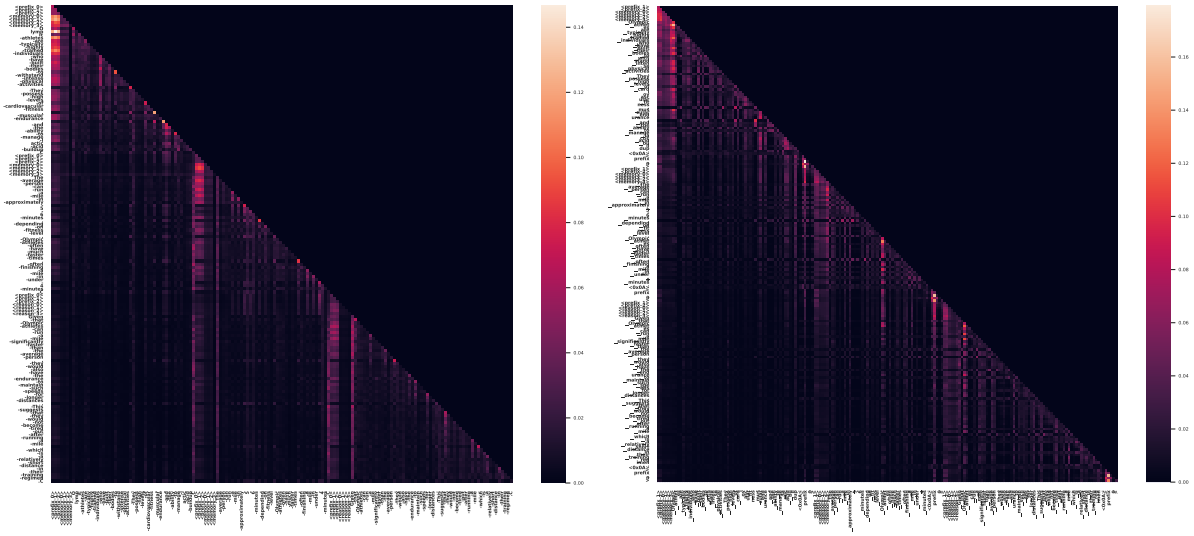

Figure 19: **Left**: Attention Map of LLaMA-3.1-8B. **Right**: Attention Map of LLaMA-2-7B.

One-shot CoT Example for Evaluating Factual Knowledge and Reasoning

Here is an example:

<Question>
'Is Mixed martial arts totally original from Roman Colosseum games?'
<Question>

<Steps>
'[memory]: Mixed Martial Arts (MMA) is a full-contact combat sport that allows a wide variety of fighting techniques from different martial arts traditions. It permits both striking and grappling, both standing and on the ground, using techniques from disciplines such as boxing, wrestling, Brazilian jiu-jitsu, Muay Thai, karate, and judo.',
'[memory]: The Roman Colosseum games, also known as gladiatorial games, were violent contests where gladiators fought against each other, condemned criminals, or wild animals. These events were held in large amphitheaters like the Colosseum in Rome and were a form of public spectacle and entertainment in ancient Rome.',
'[memory]: Modern MMA is characterized by regulated rules, weight classes, and a combination of various martial arts disciplines. It is officiated with rules to ensure the safety of participants, and fights occur in a controlled environment, often inside a cage.',
"[reason]: MMA and the Roman Colosseum games share the concept of hand-to-hand combat but differ significantly in purpose, structure, and regulation. While MMA is a sport with rules designed for competition and fighter safety, the Roman games were more about public spectacle and entertainment without much emphasis on fairness or safety. The combat in Roman games was often deadly and executed for the spectators' pleasure.",
"[reason]: MMA is not totally original from the Roman Colosseum games. Although both involve unarmed combat, MMA is a modern sporting discipline that synthesizes traditional martial arts into a competitive and regulated environment. The Roman games served as a historical precedent for public combat events but lacked the structured and safety-oriented approach of MMA. Therefore, while there may be a historical inspiration, MMA's development as a technical and regulated sport makes it distinct and not directly derived from the Roman games."
"[Answer]: The answer is incorrect."
<Steps>

Factual knowledge is information that aligns with objective reality and can be verified through evidence or observation, such as scientific facts or historical events.
If this step needs reasoning, return [reason] as the label, if this step needs factual knowledge return [rag] as the label.

Now, here is the question:
<Question>
{**question**}
<Question>

Your answer should be:
<Steps>
Put your generated [rag] and [reason] steps here
<Steps>

Figure 20: One-shot CoT Example for Evaluating Factual Knowledge and Reasoning

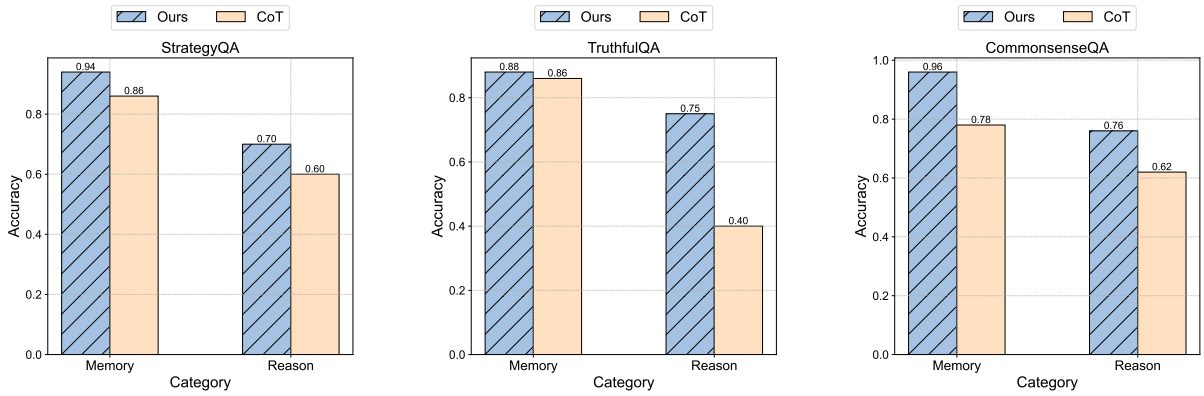

Figure 21: Decoupling Result Comparison Between Our Algorithm and One-Shot CoT prompting on all datasets and both on LLaMA-2-7B