Hierarchical Deconstruction of LLM Reasoning: A Graph-Based Framework for Analyzing Knowledge Utilization

Miyoung Ko[1*] Sue Hyun Park[1*] Joonsuk Park[2,3,4†] Minjoon Seo[1†]
[1]KAIST, [2]NAVER AI Lab, [3]NAVER Cloud, [4]University of Richmond
{miyoungko, suehyunpark, minjoon}@kaist.ac.kr
park@joonsuk.org

Abstract

Despite the advances in large language models (LLMs), how they use their knowledge for reasoning is not yet well understood. In this study, we propose a method that deconstructs complex real-world questions into a graph, representing each question as a node with predecessors of background knowledge needed to solve the question. We develop the DEPTHQA dataset, deconstructing questions into three depths: (i) recalling conceptual knowledge, (ii) applying procedural knowledge, and (iii) analyzing strategic knowledge. Based on a hierarchical graph, we quantify *forward discrepancy*, a discrepancy in LLM performance on simpler sub-problems versus complex questions. We also measure *backward discrepancy* where LLMs answer complex questions but struggle with simpler ones. Our analysis shows that smaller models exhibit more discrepancies than larger models. Distinct patterns of discrepancies are observed across model capacity and possibility of training data memorization. Additionally, guiding models from simpler to complex questions through multi-turn interactions improves performance across model sizes, highlighting the importance of structured intermediate steps in knowledge reasoning. This work enhances our understanding of LLM reasoning and suggests ways to improve their problem-solving abilities.

1 Introduction

With the rapid advancement of Large Language Models (LLMs), research interest has increasingly centered on their reasoning capabilities, particularly in solving complex questions. While many studies have assessed the general reasoning capabilities of LLMs (Wei et al., 2022a; Qin et al., 2023; Srivastava et al., 2023), the specific aspect of how these models recall and then utilize factual knowledge during reasoning has not been thoroughly explored. Some research (Dziri et al., 2023; Press et al., 2023; Wang et al., 2024) concentrate on straightforward reasoning tasks such as combining and comparing simple biographical facts to investigate the implicit reasoning skills of LLMs. However, real-world questions often demand more intricate reasoning processes that cannot be easily broken down into simple factual units. For instance, as presented in Figure 1, to answer "Why does ReLU training take less time than sigmoid or tanh training?", one must not only recall what an activation function is but also compare the characteristics of activation functions and understand the causal relationship between gradients and training speed. This type of reasoning requires drawing conclusions beyond simply aggregating facts.

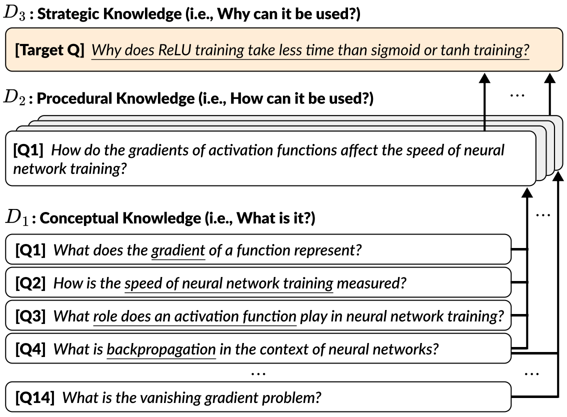

D_3 : Strategic Knowledge (i.e., Why can it be used?)
[Target Q] Why does ReLU training take less time than sigmoid or tanh training?
...
D_2 : Procedural Knowledge (i.e., How can it be used?)
[Q1] How do the gradients of activation functions affect the speed of neural network training?
...
D_1 : Conceptual Knowledge (i.e., What is it?)
[Q1] What does the gradient of a function represent?
[Q2] How is the speed of neural network training measured?
[Q3] What role does an activation function play in neural network training?
[Q4] What is backpropagation in the context of neural networks?
...
...
[Q14] What is the vanishing gradient problem?

Figure 1: Example of reasoning across depths, showing a sequence of questions from D_1 (conceptual knowledge) to D_3 (strategic knowledge). Questions that ask deeper levels of knowledge require reasoning from multiple areas of shallower knowledge, which are represented as sub-questions.

To analyze the reasoning ability of LLMs in solving real-world questions, we propose a deconstruction of complex questions into a graph structure. In this structure, each node is represented by a question that signifies a specific level of knowledge. We

[*]Equal contribution.
[†]Equal advising.

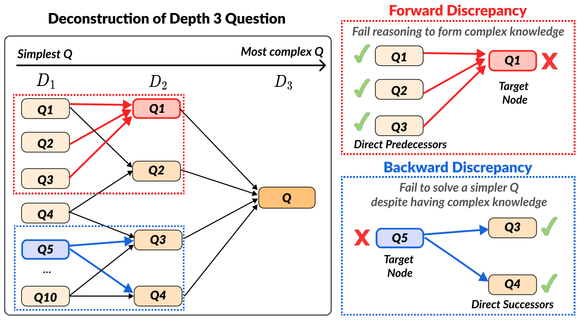


Figure 2: Hierarchical structure of a deconstructed D_3, illustrating forward and backward discrepancies. Transition to deeper nodes requires acquiring and reasoning with knowledge from the connected shallower nodes.

adopt Webb's Depth of Knowledge (Webb, 1997, 1999, 2002), which assesses both the content and the depth of understanding required. Webb's Depth of Knowledge categorizes questions into three levels: mere recall of information (D_1), application of knowledge (D_2), and strategic thinking (D_3). The transition from shallower to deeper nodes involves applying the knowledge gained from shallower nodes and performing reasoning to tackle harder problems. This approach emphasizes the gradual accumulation and integration of knowledge to address real-world problems effectively.

We introduce the resulting DEPTHQA, a collection of deconstructed questions and answers derived from human-written, scientific D_3 questions in the TutorEval dataset (Chevalier et al., 2024). The target complex questions are in D_3, and we examine the utilization of multiple layers of knowledge and reasoning in the sequence of D_1, D_2, and D_3. Figure 2 illustrates how the deconstruction process results in a hierarchical graph connecting D_1 to D_3 questions. Based on the hierarchical structure, we first measure forward reasoning gaps, denoted as *forward discrepancy*, which are differences in LLM performance on simpler subproblems compared to more complex questions requiring advanced reasoning. Additionally, we introduce *backward discrepancy*, which quantifies inconsistencies where LLMs can successfully answer complex inquiries but struggle with simpler ones. This dual assessment provides a comprehensive evaluation of the models' reasoning capabilities across different levels of complexity.

Using DEPTHQA, we investigate the knowledge reasoning ability of various instruction-tuned LLMs in the LLaMA 2 (Touvron et al., 2023), LLaMA 3 (AI@Meta, 2024), Mistral (Jiang et al., 2023), and Mixtral (Jiang et al., 2024) family, varying in size from 7B to 70B. We compare the relationship between model capacities and depthwise discrepancies, showing that smaller models exhibit larger discrepancies in both directions. We further analyze how reliance on memorization of training data affects discrepancy, revealing that forward and backward discrepancies in large models originate from distinct types of failures. Finally, to examine the importance of structured intermediate steps in reasoning, we gradually guide models from simpler to more advanced questions through multi-turn interactions, consistently improving performance across various model sizes.

The contributions of our work are threefold:

- We propose to connect complex questions with simpler sub-questions by deconstructing questions based on depth of knowledge.
- We design the DEPTHQA dataset to evaluate LLMs' capability to form complex knowledge through reasoning. We measure forward and backward reasoning discrepancies across different levels of question complexity. [1]
- We investigate the reasoning abilities of LLMs with various capacities, analyzing the impact of model size and training data memorization on discrepancies. We demonstrate the benefits of structured, multi-turn interactions to perform complex reasoning.

2 Related Work

Recent advancements have highlighted the impressive reasoning abilities of transformer language models across a wide range of tasks (Wei et al., 2022a; Zhao et al., 2023). Despite the success, numerous studies have found that these models often struggle with various types of reasoning, such as commonsense and logical reasoning (Qin et al., 2023; Srivastava et al., 2023). Even advanced models like GPT-4 (Achiam et al., 2023) have been noted to struggle with implicit reasoning over their internal knowledge, especially when it comes to effectively combining multiple steps to solve compositionally complex problems (Talmor et al., 2020; Rogers et al., 2020; Allen-Zhu and Li, 2023; Yang et al., 2024; Wang et al., 2024).

To address these challenges, several studies have focused on better Chain-of-Thought-style prompt-

[1] We release our dataset and code at github.com/kaistAI/knowledge-reasoning.

ing or fine-tuning LLMs to verbalize the intermediate steps of knowledge and reasoning during inference (Nye et al., 2021; Wei et al., 2022b; Kojima et al., 2022; Wang et al., 2022; Sun et al., 2023; Wang et al., 2023b; Liu et al., 2023). This approach has significantly improved performance, especially in larger models with strong generalization capabilities. Theoretical and empirical studies investigate the advantages of verbalizations, highlighting their role in enhancing the reasoning capabilities of language models (Feng et al., 2023; Wang et al., 2023a; Li et al., 2024). The analysis of step-by-step reasoning abilities has matured further based on ontological (Saparov and He, 2023) and mechanistic perspectives (Hou et al., 2023a; Dutta et al., 2024).

In our proposed dataset, the most complex questions often necessitate implicit intermediate steps to reach a conclusion, which can be benefited from explicit verbalized reasoning. However, unlike previous works, our setup does not induce detailed step-by-step explanation *contained* in an answer to a complex question. Instead, we represent intermediate steps for a complex question in the form of *sub-questions* and gather answers to every subquestion, testing a model's understanding of intermediate knowledge individually. Our approach is similar to strategic question answering with intermediate answers (Geva et al., 2021; Press et al., 2023), but we further ensure a hierarchy of decompositions based on knowledge complexity. This allows examining discrepancies between questions of varying complexities, providing a distinct assessment of multi-step reasoning abilities.

Another line of work focuses on understanding transformers' knowledge and reasoning through controlled experiments (Chan et al., 2022; Akyürek et al., 2023; Dai et al., 2023; von Oswald et al., 2022; Prystawski et al., 2023; Feng and Steinhardt, 2024). Numerous studies on implicit reasoning often aim to identify latent reasoning pathways, but most have focused on simple synthetic tasks or toy models (Nanda et al., 2023; Conmy et al., 2023; Hou et al., 2023b), or evaluating through binary accuracy of short-form model predictions without considering intermediate steps (Yang et al., 2024; Wang et al., 2024). Our DEPTHQA, in contrast, challenges a model to answer complex *real-world* questions that require *diverse* reasoning types in *long-form* text. DEPTHQA further requires diverse types of reasoning across different depths, such as inductive and procedural reasoning, in addition to the comparative and compositional reasoning explored in prior studies (Press et al., 2023; Allen-Zhu and Li, 2023; Wang et al., 2024). This approach provides a more practical and nuanced assessment of the model's reasoning capabilities.

3 Graph-based Reasoning Framework

We develop a novel graph-based representation that delineates the dependencies between different levels of knowledge. We represent nodes as questions (Section 3.1) and edges as reasoning processes (Section 3.2). Based on the graph definition, we construct a dataset that encompasses diverse concepts and reasoning types (Section 3.3).

3.1 Knowledge Depth in Nodes

We represent each node as a question tied to a specific layer of knowledge. As our approach to addressing real-world problems emphasizes the *gradual* accumulation of knowledge similar to educational goals, we adopt the Webb's Depth of Knowledge (DOK) (Webb, 1997, 1999, 2002) widely used in education settings to categorize the level of questions. The depth of knowledge levels $D_k (k \in \{1, 2, 3\})$[2] in questions are defined as follows:

D_1. **Factual and conceptual knowledge**: The question involves the acquisition and recall of information, or following a simple formula, focusing on *what* the knowledge entails.

D_2. **Procedural knowledge**: The question necessitates the application of concepts through the selection of appropriate procedures and step-by-step engagement, concentrating on *how* the knowledge can be utilized.

D_3. **Strategic knowledge**: The question demands analysis, decision-making, or justification to address non-routine problems, emphasizing *why* the knowledge is applicable.

The levels can be viewed as *ceilings* that establish the extent or depth of an assessee's understanding (Hess, 2006), a concept recognized as a valuable assessment tool in educational contexts (Hess et al., 2009). Accordingly, we associate simpler questions with shallower depths and more complex questions with deeper depths.

[2] We exclude the highest level in the original Webb's DOK, D_4, as this level often includes interactive or creative activities and is rare or even absent in most standardized assessment (Webb, 2002; Hess, 2006).

3.2 Criteria for Reasoning in Edges

To conceptualize how simpler knowledge contributes to the development of complex knowledge, we define edges in our framework as transitions from a node at D_k to at least one direct successor node at D_{k+1}[3]. We perceive that advancing to deeper knowledge often requires synthesizing multiple aspects of simpler knowledge. Thus, a D_k node should connect to multiple direct predecessor D_{k-1} nodes. This configuration establishes hierarchical dependencies among D_1, D_2, and D_3 questions, effectively modeling the progression needed to deepen understanding and engage with higher-order knowledge (See graph in Figure 2). Additionally, we establish three criteria to ensure that edges accurately represent the reasoning processes from shallower questions.

C1. Comprehensiveness: Questions at lower levels should aim to cover all foundational concepts necessary to answer a question at higher levels. This ensures that no critical knowledge gaps exist as the complexity increases.

C2. Implicitness: Questions at lower levels should avoid directly revealing answers or heavily hinting at solutions for higher-level questions. This encourages independent reasoning relying on the synthesis of implicit connections between nodes rather than straightforward clues.

C3. Non-binary questioning: Questions should elicit detailed, exploratory responses instead of simple yes/no answers. Given that LLMs may have an inherent positivity bias which leads them to prefer affirmative responses (Augustine et al., 2011; Dodds et al., 2015; Papadatos and Freedman, 2023), this helps in evaluating deep reasoning abilities beyond superficial or biased reasoning.

3.3 Dataset: DepthQA

We create DepthQA, a new question answering dataset for testing graph-based reasoning. The dataset is constructed in a top-down approach, deconstructing D_3 nodes into D_2 nodes, then into D_1, creating multiple edges at each step (Table 1). We design the construction process to meticulously backtrack the knowledge necessary for complex questions while meeting our three criteria for reasoning transition representation.

Domain	# Questions			# Edges between questions	
	D_1	D_2	D_3	$D_1 \rightarrow D_2$	$D_2 \rightarrow D_3$
Math	573	193	49	774	196
Computer Science	163	54	14	212	55
Environmental Science	147	44	11	175	44
Physics	140	40	10	154	40
Life Sciences	98	28	7	111	28
Math → {CS, Physics}	-	-	-	11	0
Total	1,121	359	91	1,437	363

Table 1: Statistics of DepthQA.

D_3 question curation We select real-world questions from the TutorEval (Chevalier et al., 2024) dataset, which contains human-crafted queries based on college-level mathematical and scientific content from textbooks[4] available on `libretexts.org`. Note that while these textbooks may be part of models' pre-training data due to online availability, TutorEval's human-written questions challenge models to generalize familiar concepts beyond direct training examples. We procure only complex D_3 questions from TutorEval, sorting them out using GPT-4 Turbo[5] (Achiam et al., 2023) with guidance on depth of knowledge levels. From an initial set of 834 questions, we manually refine our selection to 91 self-contained D_3 questions, ensuring clarity. We use GPT-4 Turbo to generate reference answers for each TutorEval question[6], based on the original context and the model's self-annotated depth of knowledge. These reference answers are guided by the ground-truth key points provided by the author of each question.

Question deconstruction For each D_k question, we use GPT-4 Turbo to generate up to four D_{k-1} questions. The prompt includes definitions for all three knowledge depths and decomposition examples to guide the deconstruction process. We provide D_k with its reference answer to ensure extracted knowledge remains relevant for more challenging questions, adhering to **C1 (Comprehensiveness)**. We decide the optimal number of decompositions to four based on qualitative analysis, balancing comprehensiveness and implicitness: outlining every implicit reasoning step enhances comprehensiveness but may reduce implicitness,

[3] A foundational concept may apply to multiple advanced questions.

[4] Textbooks are designed with a scaffolding approach to knowledge development.

[5] We use the `gpt-4-0125-preview` version for GPT-4 Turbo throughout this work, including data construction, verification, and experiments.

[6] Chevalier et al. (2024) reports that GPT-4 excels in solving TutorEval problems with 92% correctness.

Depth	Reasoning type	Example question	%
3	Comparative	*In the context of computer programming, what is the difference between for and while, are they always exchangeable? Show me some cases where one is strongly preferred to the other.*	21.1
	Causal	*How does deflection of hair cells along the basilar membrane result in different perceived sound frequences?*	10.5
	Inductive	*How could a process satisfying the first law of thermodynamics still be impossible?*	8.8
	Criteria Development	*Explain if a matrix always have a basis of eigenvectors.*	8.8
2	Relational	*What factors influence the time complexity of searching for an element in a data structure?*	22.6
	Procedural	*Describe the process involved in solving cubic equations using the cubic formula.*	13.4
	Application	*How can sustainable agricultural practices contribute to food security and economic development in developing countries?*	7.3

Table 2: Representative examples of required reasoning skills in D_3 and D_2. % of instances within each depth that include the reasoning type is reported. Note that multiple reasoning types can be included in a single question.

and vice versa. Our prompt instructions carefully address this tradeoff to satisfy **C2 (Implicitness)**.

Deduplication and question augmentation We identified redundancies in knowledge and reasoning processes, where similar content appeared across different D_1 nodes linked to the same D_2 node, or between unconnected D_1 and D_2 nodes (example in Table 4). To address this, we utilize a Sentence Transformers embedding model[7] (Reimers and Gurevych, 2019) to detect and remove near-duplicate questions based on cosine similarity of their embeddings. We then employ GPT-4 Turbo to generate new, targeted questions and answers, filling any gaps in knowledge coverage. This approach has reduced misclassification of D_1 questions as D_2 by 88%, markedly enhancing **C2 (Implicitness)**. It has also decreased the total number of near-duplicates by decreased by 88%, further improving **C1 (Comprehensiveness)**. We subsequently update our graph data structure with these modifications.

Question debiasing Lastly, we undertake the task of manually rewriting 53 questions that originally invoke binary "yes" or "no" answers, ensuring **C3 (Non-binary Questioning)**. For example, a question that begins with "If I understand correctly..." is transformed into "Clarify my understanding that...", prompting the model to directly engage in analytical thinking rather than relying on simple affirmations or negations of the correctness.

Verification of hierarchy We conduct human annotation to verify the three criteria that shapes the reasoning hierarchy, reporting positive results in Appendix B. On 27.5% of DepthQA, an average of 83.5% of relations are fully comprehensive and 89.5% of sub-questions are fully implicit, with 98.7% of all questions being non-binary. Further details and examples in the construction process are in Appendix A. Prompts are in Appendix J.1.

3.4 Diversity of Reasoning Processes

Using a sample of 20 D_3 questions along with their interconnected 80 D_2 and 320 D_1 questions, we examine the types of reasoning needed to progress from basic to complex knowledge levels. We discover that nearly all questions necessitate the identification and extraction of several pieces of relevant information to synthesize comprehensive answers. Table 2 displays examples of questions requiring advanced reasoning skills, such as interpreting relationships between concepts, applying specific conditions, and handling assumptions, demonstrating that basic knowledge manipulation is insufficient. This diversity in reasoning types within our dataset robustly challenges LLMs to demonstrate sophisticated cognitive abilities. Detailed statistics and additional examples of reasoning types are provided in Appendix D.

4 Experiments

In this section, we present experiments on the depthwise reasoning ability of LLMs using DepthQA. We first explain the evaluation metrics and models (Section 4.1). Experimental results that follow are overall depthwise and discrepancy evaluation results (Section 4.2), the impact of memorization in knowledge reasoning (Section 4.3), and the effect of enforcing knowledge-enhanced reasoning via multi-turn inputs or prompt inputs (Section 4.5).

4.1 Experiment Setup

Depthwise evaluation For each question q_k with depth k (D_k), we score the factual correctness of the predicted answer on a scale from 1 to 5. We

[7] sentence-transformers/all-mpnet-base-v2

Model	Average Accuracy ↑				Forward Discrepancy ↓			Backward Discrepancy ↓		
	D_1	D_2	D_3	Overall	$D_2 \rightarrow D_3$	$D_1 \rightarrow D_2$	Overall	$D_2 \rightarrow D_3$	$D_1 \rightarrow D_2$	Overall
LLaMA 2 7B Chat	3.828	3.320	3.165	3.673	0.130	0.181	0.176	0.219	0.110	0.134
LLaMA 2 13B Chat	4.289	3.872	3.615	4.155	0.152	0.158	0.157	0.126	0.078	0.088
LLaMA 2 70B Chat	4.495	4.153	4.022	4.390	0.126	0.136	0.134	0.136	0.063	0.079
Mistral 7B Instruct v0.2	4.280	3.897	4.000	4.176	0.092	0.157	0.147	0.144	0.070	0.088
Mixtral 8x7B Instruct v0.1	4.599	4.532	4.429	4.574	0.087	0.079	0.081	0.063	0.063	0.063
LLaMA 3 8B Instruct	4.482	4.351	4.286	4.440	0.083	0.096	0.093	0.088	0.072	0.075
LLaMA 3 70B Instruct	**4.764**	**4.749**	**4.648**	**4.754**	**0.065**	**0.050**	**0.053**	**0.043**	**0.044**	**0.044**
GPT-3.5 Turbo	4.269	4.251	4.011	4.250	0.100	0.072	0.078	0.046	0.067	0.063

Table 3: Depthwise reasoning performance of large language models. **Bold** indicates the best-performing model, and underline represents the second best performance. A darker color indicates a higher discrepancy.

employ the LLM-as-a-Judge approach, which correlates highly with human judgments in scoring long-form responses (Zheng et al., 2024; Kim et al., 2024a; Lee et al., 2024; Kim et al., 2024b). Specifically, we utilize GPT-4 Turbo (Achiam et al., 2023) for absolute scoring. Following Kim et al. (2024a) and Lee et al. (2024), the model generates a score and detailed feedback for each question, reference answer, and prediction based on a defined scoring rubric. Further details on the evaluation process are provided in Appendix E. The exact input prompt for the LLM judge including the accuracy score rubric is in Appendix J.3. The reliability of the LLM evaluation results in our setting is evidenced by high annotation agreement with human evaluations, as explained in Appendix F. We report **average accuracy** at D_k, the averaged factual correctness of questions at depth k.

Discrepancy evaluation As we deconstruct complex questions into a hierarchical graph, we can measure *forward discrepancy* and *backward discrepancy* between neighboring questions. **Forward discrepancy** measures the differences in performance on sub-problems compared to deeper questions requiring advanced reasoning. Given a question q_k at $D_k \in \{2, 3\}$, let $DP(q_k)$ represents a set of direct predecessor questions at D_{k-1}. Then forward discrepancy for q_k is defined as follows:

$$\text{Forward Discrepancy}(q_k) =$$
$$\max\left(0, \frac{1}{4}\left(\text{avg}_{q \in DP(q_k)}[f(q)] - f(q_k)\right)\right) \quad (1)$$

where f is a function of a question that outputs factual correctness, as measured by an LLM evaluator. **Backward discrepancy**, conversely, quantifies inconsistencies where LLMs can successfully answer deeper questions but struggle with shallower ones. Given a question q_k with $D_k \in \{1, 2\}$, let $DS(q_k)$ represent a set of direct successor questions at D_{k-1}. Then backward discrepancy is defined as follows:

$$\text{Backward Discrepancy}(q_k) =$$
$$\max\left(0, \frac{1}{4}\left(\text{avg}_{q \in DS(q_k)}[f(q)] - f(q_k)\right)\right). \quad (2)$$

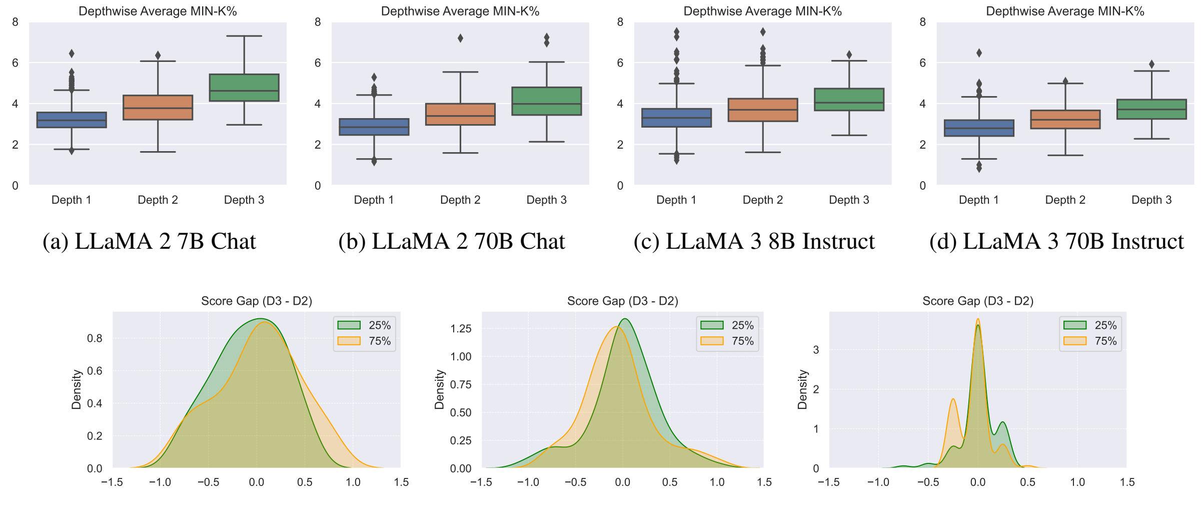
Both forward discrepancy and backward discrepancy are normalized to the range [0, 1] by dividing by the *maximum possible score gap*, which is 4 at our scoring range from 1 to 5. To highlight gaps across depths, we set a strict accuracy threshold of 4 and report the average discrepancies only for examples where the mean score for $DP(q^k)$ and $DS(q^k)$ exceeds this threshold. This excludes cases where models perform poorly at both depths.

Models We mainly probe into the depthwise knowledge reasoning ability of open-source LLMs. We test representative open-source models based on the LLaMA (Touvron et al., 2023) architecture, including LLaMA 2 {7B, 13B, 70B} Chat (Touvron et al., 2023), Mistral 7B Instruct v0.2 (Jiang et al., 2023), Mixtral 8x7B Instruct v0.1 (Jiang et al., 2024), and LLaMA 3 {8B, 70B} Instruct (AI@Meta, 2024). Additionally, we include the latest GPT-3.5 Turbo[8] (OpenAI, 2022) to compare the performance of these open-source models against a proprietary model.

4.2 Depthwise Knowledge Reasoning Results

Larger models exhibit smaller discrepancies. Table 3 presents the overall depthwise reasoning performance of LLMs. As anticipated, solving questions at D_3 is the most challenging, showing the lowest average accuracy for all models. LLaMA 3 70B Instruct demonstrates the best performance across all depths, with Mixtral 8x7B Instruct achieving the second-best results. LLaMA 3

[8] gpt-3.5-turbo-0125



(a) LLaMA 2 7B Chat (b) LLaMA 2 70B Chat (c) LLaMA 3 8B Instruct (d) LLaMA 3 70B Instruct

(e) LLaMA 2 7B Chat (f) LLaMA 2 70B Chat (g) LLaMA 3 70B Instruct

Figure 3: Memorization analysis with Min-K% probability. (a)-(d) show the distribution of average Min-K% probabilities at each depth. (e)-(g) present the distribution of score differences between neighboring questions, whose Min-K% probability is in the bottom 25% or top 75%. A positive gap indicates backward discrepancy, while a negative gap represents forward discrepancy.

70B Instruct also exhibits the lowest discrepancies for both forward and backward discrepancy metrics, effectively answering questions at all depths with minimal discrepancies. Conversely, the least capable model, LLaMA 2 7B Chat, shows the lowest average accuracy along with the highest forward and backward discrepancies. Note that the relatively low forward discrepancy from $D_1 \rightarrow D_2$ for LLaMA 2 7B Chat is due to its low performance at D_2. This observation highlights the varying capabilities of different LLMs in handling questions at different depths and the inconsistencies in reasoning across depths.

Contrasting patterns of discrepancies We observe distinct patterns when analyzing forward and backward discrepancies separately. These discrepancies can be understood as a product of intensity (the magnitude of the discrepancies) and frequency (the proportion of questions showing a positive discrepancy). Frequency indicates how often forward discrepancy or backward discrepancy occurs, while intensity reflects the strength of the discrepancy when it happens. Our analysis shows that forward discrepancy tends to occur more frequently but with lower intensity. For example, LLaMA 3 8B Instruct exhibits an intensity of 0.225 with a frequency of 41.44%. In contrast, backward discrepancy is less common but has a higher intensity when they appear. Specifically, LLaMA 3 8B Instruct shows an intensity of 0.323 with a frequency of 23.32% for backward discrepancies. The intensity and frequency for all models are provided in Appendix G.

4.3 Memorization in Depthwise Knowledge Reasoning

4.3.1 Depthwise Memorization

To determine whether solving complex questions requires reasoning rather than memorization of training data, we use a pre-training data detection method to approximate potential aspects of memorization. Following Shi et al. (2023), we compare the **Min-K% probability** within models. Higher values suggest a smaller possibility of predictions directly existing in the training data. To elaborate, Min-K% probability is calculated by averaging the negative log-likelihood of the K% least probable tokens in the model's predictions. In the case where a given prediction was seen during training, outlier words with low probabilities would appear less frequently, resulting in high probabilities for the K% tokens. Since Min-K% probability is the average negative log-likelihood of such tokens, the resulting value would be lower in this case. [9]

Models rely less on memorization for complex questions. Figure 3 (a)-(d) presents the depthwise average of the Min-K% probability for four models. We observe that as the depth increases,

[9] For our calculations, we set k to 20 and use a sequence length of 128.

the Min-K% probability also increases for all models. This indicates that answering questions based on simple conceptual knowledge corresponding to D_1 is more likely to be solved by recalling training data. While shallow questions (D_1) can be addressed through memorization, solving deeper questions (D_3) requires more than just recalling a single piece of memorized knowledge, indicating a need for genuine reasoning capabilities.

4.3.2 Memorization Gap between Depths

Further analysis of questions in the bottom 25% and top 75% quantiles of the Min-K% probability distribution provides additional insights. Note that questions in the top 75% quantile are more likely to appear in the training data, while those in the bottom 25% are less likely. Figure 3 (e)-(g) shows the score difference between neighboring questions ($D_2 \rightarrow D_3$) whose Min-K% probability is in the bottom 25% or top 75%. We calculate the **memorization gap** as the difference between the factual correctness of D_3 and D_2, normalized by the maximum gap of 4. A positive value indicates higher factual accuracy for the deeper questions, signifying backward discrepancy, while a negative value indicates higher accuracy for the shallower question, representing forward discrepancy.

Variance of gaps We observe that the model with the smallest capacity, LLaMA 2 7B Chat, exhibits large variances in both positive and negative directions, showing significant forward and backward discrepancies. In contrast, models with larger capacities, such as LLaMA 2 70B Chat and LLaMA 3 70B Instruct, demonstrate smaller variances.

Potential causes of discrepancies Additionally, models with larger capacities tend to show relatively higher forward discrepancies—distribution concentrated on the negative side—for the top 75% examples, which rely less on memorization. On the other hand, the bottom 25% distribution is concentrated on positive values, indicating relatively more backward discrepancies. This suggests that as model capacity increases, failures in knowledge reasoning result in forward discrepancies, while failures due to reliance on memorization may lead to backward discrepancies. The depthwise Min-K% probability and score difference for other models are provided in Appendix H.

4.4 Qualitative Analysis of Backward Discrepancy

To better understand how the more abnormal inconsistency—backward discrepancy—can emerge, we qualitatively analyze backward discrepancy cases from the weakest model in our experiments, LLaMA 2 7B chat, and the strongest model in our experiments, LLaMA 3 70B Instruct. The examples we refer to in the following paragraphs are listed in Appendix I.

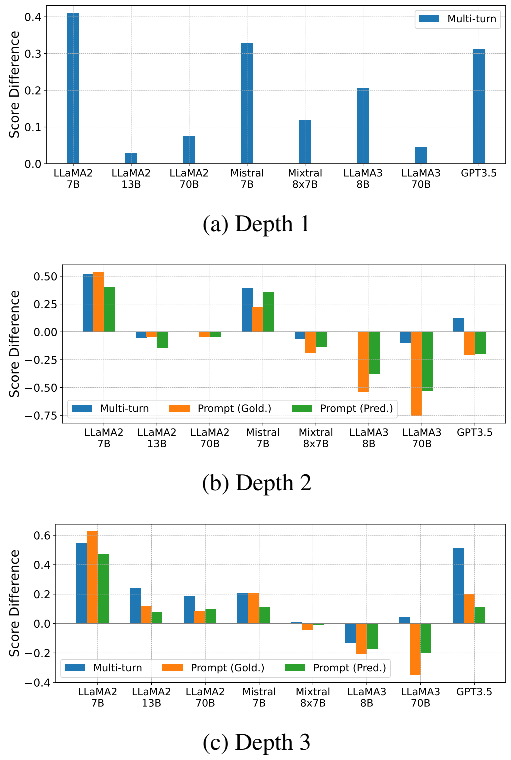
We observe that backward discrepancies often stem from the models' ability to articulate high-level concepts but struggle with translating this understanding into precise, step-by-step procedures, particularly when mathematical operations are involved. This is illustrated in Example 1, where both models explain the importance of continued fraction representation for tangle numbers well (D_3) but fail to accurately describe the process of constructing a tangle for a given number (D_2).

In backward discrepancy cases, answers to deeper questions are more likely to be text-based and conceptual, making them easier for models to memorize that data. In contrast, shallower questions require execution of mathematical or logical operations, where the variability in the elements makes answers harder to memorize verbatim. This elucidates memorization effects on backward discrepancy analyzed in Section 4.3.2.

Interestingly, we also observe how the degree of memorization contributing to backward discrepancy can vary with model capacity. Example 2 shows LLaMA 2 7B Chat accurately reasoning about time complexity (D_3) but introducing non-standard terminology for specific operations (D_2), suggesting the model's struggle with precise recall of basic concepts. Conversely, Example 3 demonstrates LLaMA 3 70B Instruct correctly recalling a complex formula (D_3) but failing to apply it practically (D_2). This indicates that the model can extensively memorize information but still struggle with its flexible application. This observation exemplifies why variance of memorization gaps can differ by model capacity, as described in Section 4.3.2.

4.5 Effect of Explicit Reasoning Process

In this study, as presented in Figure 1 (a), D_3 questions can be solved through sequential reasoning, utilizing answers from D_1 to D_3 questions. Previous studies on implicit reasoning (Wei et al., 2022b; Press et al., 2023; Zhou et al., 2023) have shown



(a) Depth 1

(b) Depth 2

(c) Depth 3

Figure 4: Performance change after providing shallower questions. Note that D_1 is not reported for prompt inputs, as D_1 does not have shallower questions.

that enforcing LLMs to reason through intermediate steps explicitly can improve their reasoning ability. We investigate whether explicitly providing these reasoning processes to the model can aid in solving complex questions.

We encourage the model to reason by providing shallower questions in three ways: (i) **Multi-turn**, where shallower questions are provided as user queries in a multi-turn conversation; (ii) **Prompt (Gold)**, where shallower questions and their gold answers are provided in prompts; (iii) **Prompt (Pred.)**, where shallower questions with the model's predictions are given in prompts. Note that prompt-based approaches require shallower QA pairs as inputs, which cannot be applied to D_1 questions. The prompt template for each approach is provided in Appendix J.2.

Explicitly providing shallower solutions is beneficial for small models and complex questions. Figure 4 illustrates the depthwise performance changes after incorporating deconstructed question information. Providing shallower questions benefits models with smaller capacities, such as LLaMA 2 7B Chat and Mistral 7B Instruct v0.2. For relatively simpler questions (D_2), the benefit is less pronounced or may even decrease the performance of more capable models (>7B). However, intermediate questions (D_2) are beneficial for complex questions (D_3), except for models with large capacities ($\geq$ 56B). These findings align with recent research on decomposing a complex question into simpler sub-tasks and solving sub-tasks prior to the final answer (Juneja et al., 2023; Khot et al., 2023), which have shown high performance improvements for solving complex problems across different model sizes.

Implicitly guiding reasoning via multi-turn interactions best improves performance. When comparing the two prompt-based inputs, smaller models tend to perform better with gold answers (Gold.), while more capable models favor self-prediction results (Pred.). This preference likely arises because more capable models align better with their own generated outputs, which reflect their advanced internal reasoning processes. The multi-turn approach provides the most stable results across all depths, enhancing the performance of smaller models while causing minimal performance drops for larger models. Additionally, the multi-turn approach improves D_1 performance by providing context or domain information as part of the interaction history.

5 Conclusion

In this study, we explore the reasoning capabilities of LLMs by deconstructing real-world questions into a graph. We introduce DEPTHQA, a set of deconstructed D_3 questions mapped into a hierarchical graph, requiring utilization of muliple layers of knowledge in the sequence of D_1, D_2 to D_3. This hierarchical approach provides a comprehensive assessment of LLM performance by measuring forward and backward discrepancies between simpler and complex questions. Our comparative analysis of LLMs with different capacities reveals an inverse relationship between model capacities and discrepancies. Memorization analysis suggests that the sources of forward and backward discrepancies in large models stem from different types of failures. Lastly, we demonstrate that guiding models from shallower to deeper questions through multi-turn interactions stabilizes performance across the majority of models. These findings emphasize the importance of intermediate knowledge extraction in understanding LLM reasoning capabilities.

Limitations

Small sample size Our dataset, DEPTHQA, consists of 91 complex (D_3) questions from the TutorEval dataset, along with 1,480 derived shallower (D_2, D_1) questions. Despite the diversity in reasoning types explored (Section 3.4) and the hierarchical structuring of subquestions, the limited number of complex questions and the narrow content scope restrict the generalizability of our findings. The selection of TutorEval as our primary source is based on the challenge of manually developing or even sourcing intricate questions that necessitate advanced reasoning skills; such questions require (1) maintaining real-world relevance, (2) eliciting long-form answers, and (3) having minimal risk of test set contamination. Within TutorEval, complex D_3 questions represent only 33.6% of its 834 questions, which further reduces to 10.9% when excluding questions that require external knowledge retrieval. We encourage future research to build larger, more diverse datasets to more robustly assess knowledge reasoning capabilities of LLMs.

GPT-4 data generation and evaluation All questions except for D_3 and reference answers in DEPTHQA are generated by GPT-4 Turbo. To ensure the quality of these questions, we have established strict decomposition criteria (Section 3.2) and implemented rigorous procedures including detailed instructions, question augmentation, manual rewriting and verification by human annotators (Section 3.3). The reliability of the answers is supported by findings from Chevalier et al. (2024), which demonstrate GPT-4's high accuracy of 92% on TutorEval problems as assessed by human evaluators. However, there may exist inaccuracies due to unseen errors in the decomposition process or unverified knowledge produced by the model.

Furthermore, we utilize GPT-4 Turbo to assess the correctness of model predictions. Following protocols from previous studies (Kim et al., 2024a,b) which highlight GPT-4's strong correlation with human judgments on long-form content, we provide detailed instructions and specific scoring rubrics to the evaluator to ensure that the evaluation process aligns closely with our objectives. In addition, we conduct human evaluations and compare with GPT-4 Turbo evaluations, and measure sufficiently high inter-annotator agreement (Appendix F). Still, the evaluation method is subject to bias inherent in LLM judges.

Acknowledgement

We thank Hyeonbin Hwang, Sohee Yang, and Sungdong Kim for constructive feedback and discussions. This work was partly supported by KAIST-NAVER Hypercreative AI Center and Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (RS-2024-00398115, Research on the reliability and coherence of outcomes produced by Generative AI, 30%).

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*.

AI@Meta. 2024. Llama 3 model card.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. What learning algorithm is in-context learning? investigations with linear models. In *ICLR*.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.2, knowledge manipulation. *arXiv*.

Adam A Augustine, Matthias R Mehl, and Randy J Larsen. 2011. A positivity bias in written and spoken english and its moderation by personality and gender. *Social Psychological and Personality Science*, 2(5):508–515.

Santiago Castro. 2017. Fast Krippendorff: Fast computation of Krippendorff's alpha agreement measure. https://github.com/pln-fing-udelar/fast-krippendorff.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. In *NeurIPS*, pages 18878–18891.

Alexis Chevalier, Jiayi Geng, Alexander Wettig, Howard Chen, Sebastian Mizera, Toni Annala, Max Jameson Aragon, Arturo Rodríguez Fanlo, Simon Frieder, Simon Machado, et al. 2024. Language models as science tutors. *arXiv*.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *NeurIPS*. Curran Associates, Inc.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of ACL*.

Peter Sheridan Dodds, Eric M Clark, Suma Desu, Morgan R Frank, Andrew J Reagan, Jake Ryland Williams, Lewis Mitchell, Kameron Decker Harris, Isabel M Kloumann, James P Bagrow, et al. 2015. Human language reveals a universal positivity bias. *Proceedings of the national academy of sciences*, 112(8):2389–2394.

Subhabrata Dutta, Joykirat Singh, Soumen Chakrabarti, and Tanmoy Chakraborty. 2024. How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning. *TMLR*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang (Lorraine) Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. In *NeurIPS*.

Guhao Feng, Yuntian Gu, Bohang Zhang, Haotian Ye, Di He, and Liwei Wang. 2023. Towards revealing the mystery behind chain of thought: a theoretical perspective. *NeurIPS*.

Jiahai Feng and Jacob Steinhardt. 2024. How do language models bind entities in context? In *ICLR*.

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. *TACL*.

K Hess. 2006. Applying webb's depth-of-knowledge (dok) levels in science. *Accessed November*, 10.

Karin Hess, Ben Jones, Dennis Carlock, and John R Walkup. 2009. Cognitive rigor: Blending the strengths of bloom's taxonomy and webb's depth of knowledge to enhance classroom-level processes. *ERIC Document (Online Database)*.

Yifan Hou, Jiaoda Li, Yu Fei, Alessandro Stolfo, Wangchunshu Zhou, Guangtao Zeng, Antoine Bosselut, and Mrinmaya Sachan. 2023a. Towards a mechanistic interpretation of multi-step reasoning capabilities of language models. In *EMNLP*.

Yifan Hou, Jiaoda Li, Yu Fei, Alessandro Stolfo, Wangchunshu Zhou, Guangtao Zeng, Antoine Bosselut, and Mrinmaya Sachan. 2023b. Towards a mechanistic interpretation of multi-step reasoning capabilities of language models. In *EMNLP*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv*.

Gurusha Juneja, Subhabrata Dutta, Soumen Chakrabarti, Sunny Manchanda, and Tanmoy Chakraborty. 2023. Small language models fine-tuned to coordinate larger language models improve complex reasoning. In *EMNLP*.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks. In *ICLR*.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024a. Prometheus: Inducing evaluation capability in language models. In *ICLR*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024b. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *NeurIPS*.

Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

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 *SOSP*.

Seongyun Lee, Seungone Kim, Sue Hyun Park, Geewook Kim, and Minjoon Seo. 2024. Prometheus-vision: Vision-language model as a judge for fine-grained evaluation. *arXiv*.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. 2024. Chain of thought empowers transformers to solve inherently serial problems. In *ICLR*.

Jiacheng Liu, Ramakanth Pasunuru, Hannaneh Hajishirzi, Yejin Choi, and Asli Celikyilmaz. 2023. Crystal: Introspective reasoners reinforced with self-feedback. In *EMNLP*.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. In *ICLR*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv*.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

Henry Papadatos and Rachel Freedman. 2023. Your llm judge may be biased. https://www.lesswrong.com/posts/S4aGGF2cWi5dHtJab/your-llm-judge-may-be-biased. Accessed: 2023-06-14.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of EMNLP*.

Ben Prystawski, Michael Y. Li, and Noah Goodman. 2023. Why think step by step? reasoning emerges from the locality of experience. In *NeurIPS*.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a general-purpose natural language processing task solver? In *EMNLP*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP-IJCNLP*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *TACL*.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *ICLR*.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *ArXiv*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *TMLR*.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2023. Recitation-augmented language models. In *ICLR*.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-on what language model pre-training captures. *TACL*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv*.

Johannes von Oswald, Eyvind Niklasson, E. Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2022. Transformers learn in-context by gradient descent. In *ICML*.

Boshi Wang, Xiang Deng, and Huan Sun. 2022. Iteratively prompt pre-trained language models for chain of thought. In *EMNLP*.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023a. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *ACL*.

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *ICLR*.

Norman L Webb. 1997. Criteria for alignment of expectations and assessments in mathematics and science education. research monograph no. 6.

Norman L Webb. 1999. Alignment of science and mathematics standards and assessments in four states. research monograph no. 18.

Norman L Webb. 2002. Depth-of-knowledge levels for four content areas. *Language Arts*, 28(March):1–9.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *TMLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *NeurIPS*.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024. Do large language models latently perform multi-hop reasoning? *arXiv*.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2024. FLASK: Fine-grained language model evaluation based on alignment skill sets. In *ICLR*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Ji rong Wen. 2023. A survey of large language models. *ArXiv*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *NeurIPS*, 36.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complexz reasoning in large language models. In *ICLR*.

A Details in Dataset Construction

Classifying questions based on depth of knowledge To categorize questions from the TutorEval dataset (Chevalier et al., 2024), we use GPT-4 Turbo set at a temperature of 0.7, following the specific prompt detailed in Table 19. We evaluate the model's classification accuracy using a validation set of 50 questions, which we have previously annotated with their respective depth of knowledge levels. Our optimal prompting strategy involves incorporating key points from each question provided in the original dataset and instructing the model to provide a step-by-step explanation of its classification reasoning. This approach achieves a precision of 0.67 and a recall of 0.77, with a low rate of false positives. Analysis of the entire set of 834 questions reveals the distribution of depth levels: 43% at D_2, 33.6% at D_3, 23.3% at D_1, and only one question at D_4.

D_3 question filtering and disambiguation From the 280 D_3 questions initially identified, we manually exclude questions that are not self-contained, meaning they refer to specific contexts or excerpts in textbook passages that cannot be seamlessly integrated into our input. Examples include questions like, "I don't understand the point of *Theorems 4.3.2 and 4.3.3*. Why do we care about these statements?" and "Please tell me the common conceptual points between *the Weinrich and Wise 1928 study* and *the Roland et al. 1980 paper*." Additionally, we disambiguate questions to ensure clarity and context accuracy. For example, the question "Why is branching unstructured? And is it a bad design choice?" was initially vague about its reference to 'branching.' Upon review, we identify the context as computer programming rather than database systems and revise the question to: "In the context of computer programming, why is branching considered unstructured, and is it considered a poor design choice?".

Question deduplication and augmentation As explained in Section 3.3, we leverage cosine similarity of question embeddings produced by a Sentence Transformers embedding model[10] (Reimers and Gurevych, 2019) to identify near-duplicate questions. Specifically, within the same depth 1 or 2, we apply a similarity threshold of 0.9 to identify duplicates and eliminate them. For questions across D_1 and D_2, we remove D_2 questions with a similarity score ≥ 0.9 to D_1 questions and remove D_1 questions with a similarity score between 0.8 and 0.9 to D_1 questions, under the rationale that the D_2 question is effectively at D_1 level and vice versa. The similarity threshold and the candidate to remove was determined through qualitative analysis. We analyze that the false positive rate of D_1 questions misclassified as D_2 decreases from 60% to 7% (reduction rate of 88%). Table 4 compares top-1 similar question pairs between D_2 and D_1 before and after deduplication. We further use GPT-4 Turbo with temperature 0.7 (prompt in Table 25 and 26) to augment questions filling the gap of knowledge. Table 5 illustrates how an augmented question successfully replaces a previously redundant question, enhancing comprehensiveness.

Top-1 before deduplication (similarity = 0.97)
D_2: How do you calculate the determinant of a matrix?
D_1: How do you find the determinant of a matrix?

Top-1 after deduplication (similarity = 0.93)
D_2: What does it mean for two vectors to be orthogonal, and how can you verify this property?
D_1: What does it mean for two vectors to be orthogonal?

Table 4: Top-1 similar question pairs between D_2 and D_1 before and after the deduplication and augmentation process. While the pair above shares essentially the same depth of knowledge, the pair below substantially differ in knowledge depth due to the D_2 question asking additional procedures.

Describe how division and remainders work when considering congruence modulo a number.

1. What is the result of a division called?
2. How is a remainder defined in division?
3. What does it mean for two numbers to be congruent modulo a number?
4. What does the term 'congruence modulo a number' mean?
   ⇒ What is the modulo operation in mathematics?

Table 5: The original 4th shallower question (red) is asking redundant knowledge addressed in the 3rd question. We remove the duplicate question and replace it with a question asking a different concept (blue).

Motivation of question debiasing In our preliminary study, we found that models tend to favor "yes" over "no" at the beginning of the response to a question that can be answered in binary format, as exemplified in Table 6. We recognize that the inherent positivity bias in models (Augustine et al., 2011; Dodds et al., 2015; Papadatos and Freedman, 2023) has the potential to skew the model's rea-

[10] sentence-transformers/all-mpnet-base-v2

Original question: Being congruent modulo an ideal sounds a bit like two numbers being congruent modulo another number. **Is there a connection** between these two notions? ⇒ Yes	*Flipped question:* Are the concepts of being congruent modulo an ideal and two numbers being congruent modulo another number **entirely distinct with no connection**? ⇒ No
Predicted answer: Yes, there is a connection between the two notions. In mathematics, when we say that two integers are congruent modulo n, it means that they have the same remainder when divided by n. For example, 3 and 6 are congruent modulo 3 because when we divide 3 by 3, the remainder is 0, and when we divide 6 by 3, the remainder is also 0. [...]	*Predicted answer:* Yes, the concepts of being congruent modulo an ideal and two numbers being congruent modulo another number are distinct. When we say that two numbers a and b are congruent modulo n, we mean that their difference a - b is a multiple of n. This is written as a ≡ b (mod n). [...]

Table 6: Example of Mistral 7B Instruct v0.2 favoring affirmative responses over negative responses when the knowledge required is consistent but only the question format is flipped.

Are there problems that one can use standard induction to prove but cannot use strong induction to prove? ⇒ What kind of problems can be proven using standard induction but not strong induction?
If I understand correctly, adding sine functions always results in a new sine function? ⇒ Clarify my understanding that adding sine functions always results in a new sine function.
Can a linear transformation map all points of a vector space to a single point, and under what conditions does this occur? ⇒ Describe the possibility of a linear transformation mapping all points of a vector space to a single point. Under what conditions does this occur?

Table 7: Example conversions of a binary question into a non-binary question.

soning processes and consequently obscure a true evaluation of its capability to reason and articulate nuanced thoughts. To mitigate this, we debias problematic questions by reframing them into more exploratory inquiries. Example transformations are in Table 7.

B Human Verification on Data Quality

2 of the authors and one graduate student who volunteered annotate 27.5% of DepthQA, verifying the three criteria we hold in Section 3.2: Comprehensiveness (C1), Implicitness (C2), and Non-binary questioning (C3). Comprehensiveness and Implicitness are especially crucial criteria for sub-questions to ensure the hierarchy in the reasoning process, as Comprehensiveness ensures no critical knowledge gaps with increasing depth, while Implicitness ensures no straightforward clues, encouraging implicit reasoning between sub-questions.

To set up the procedure, we randomly sample 5 D_3 questions from each of the 5 domains in our dataset and use all questions derived from the selected D_3 questions, totaling 25 D_3, 100 D_2, and 396 D_1 questions. Given 25 $D_3 \rightarrow D_2$ and 100 $D_2 \rightarrow D_1$ relations, the relations are divided into 40, 40, 45 and are assigned to the three workers. For each relation, the main question and the sub-questions (predecessors) are provided along with their gold answers. Then the labeler is asked to check whether the relation is conceptually comprehensive and whether each question is implicit or non-binary. The labeler can choose from three varying degrees of comprehensiveness and implicitness due to the subjective nature of the criteria. The annotation interface is shown in Figure 7.

Table 8, 9, and 10 reports the annotation statistics. Table 10 shows that the decompositions into shallower questions are fully comprehensive (C1) in 88% of $D_3 \rightarrow D_2$ relations and 79% of $D_2 \rightarrow D_1$ relations, reaching 100% and 97% when taking partially comprehensive relations as well, respectively. Also, Table 9 shows that 87% of D_2 and 91.9% of D_1 questions do not hint at solutions for more complex questions (C2), with similarly low failure rates. We also find in Table 10 that

C1. Comprehensiveness	$D_3 \rightarrow D_2$		$D_2 \rightarrow D_1$	
	Count	%	Count	%
Comprehensive	22	88.0	79	79.0
Partially comprehensive	3	12.0	18	18.0
Insufficient	0	0.0	3	3.0

Table 8: Human annotation on Comprehensiveness of a subset of DepthQA question relations.

C2. Implicitness	D_2		D_1	
	Count	%	Count	%
Fully implicit	87	87.0	364	91.9
Partially comprehensive	13	13.0	31	7.8
Insufficient	0	0.0	1	0.3

Table 9: Human annotation on Implicitness of a subset of DepthQA sub-questions.

C3. Non-binary Questioning	D_3		D_2		D_1	
	Count	%	Count	%	Count	%
Open-ended	24	96.0	100	100.0	396	100.0
Binary	1	4.0	0	0.0	0	0.0

Table 10: Human annotation on Non-binary questioning of a subset of DEPTHQA sub-questions.

Reasoning Type	Depth 3		Depth 2	
	Count	%	Count	%
Comparative	12	21.1	19	11.6
Relational	10	17.5	37	22.6
Causal	6	10.5	19	11.6
Inductive	5	8.8	6	3.7
Criteria Development	5	8.8	13	7.9
Procedural	4	7.0	22	13.4
Evaluative	4	7.0	12	7.3
Example	2	3.5	8	4.9
Quantitative	2	3.5	6	3.7
Application	2	3.5	19	11.6
Other	5	8.8	3	1.8
Total	57	100	164	100

Table 11: Distribution of reasoning types for D_3 and D_2 in a subset of DEPTHQA. Multiple reasoning types can be included in one instance.

nearly all questions require open-ended answers (C3). Human verification data provides evidence that our synthetically generated edges in the adequately represent the reasoning process.

C Dataset License

The TutorEval (Chevalier et al., 2024) dataset from which we source complex questions has not disclosed the license yet. Our DEPTHQA is subject to OpenAI's Terms of Use for the generated data. We will notify the intended use of our dataset for research when releasing our dataset to the public.

D Reasoning Type Analysis

In Table 11, we report the distribution of reasoning types annotated by the authors on a sample of 20 D_3 questions and D_2 and D_2 related to them. Table 15 outlines the definition of each reasoning type and a representative example set of questions that best elicits such reasoning. We provide question deconstructions examples in Table 16 and Table 17 where each showcases distinct reasoning types and knowledge.

E Details in Main Experiments

E.1 Model Inference

To inference LLMs used in our experimental setup (Section 4.1), we use a standardized API from OpenRouter[11] to access LLMs and use the complementary LiteLLM[12] interface to call model generations. An exception is LLaMA 7B Chat, which is not hosted in OpenRouter; we use the HuggingFace model and the vLLM (Kwon et al., 2023) inference engine for this particular model, performing local inference with mixed precision on 1 NVIDIA A6000 40GB GPU. We use the default sampling parameters suited for each model. The specific prompt templates used to induce reasoning paths are organized in Appendix J.2. The inference on the whole pass of DEPTHQA finishes within 10 minutes. We report single-run results.

E.2 LLM-as-a-Judge Evaluation

When prompting GPT-4 Turbo to evaluate model responses, we use a temperature of 1.0, nucleus sampling with `top_p` of 0.9, and maximum number of generation tokens of 1,024, following previous works (Ye et al., 2024; Kim et al., 2024a,b; Lee et al., 2024). The prompt template including the score rubric is in Table 30. We report single-run results. See Table 18 for example output format. Unlike prior works that emphasize the use of instance-specific scoring rubrics (Kim et al., 2024a,b; Lee et al., 2024), our initial experiments comparing evaluations given a common rubric and instance-specific rubric showed that instance-specific rubrics increase noise in evaluation and decrease the quality of evaluation. We speculate that it is because the focus of our evaluation is on a *common* factor of factual correctness, *i.e.*, whether the model accurately uses knowledge in the reasoning process, different from conventional benchmark evaluations.

F Reliability of LLM-as-a-Judge

To assess the reliability of LLM evaluations in our analysis, we conduct human evaluation of LLM responses and calculate the agreement between annotations. We randomly sample 20 model responses for each score level (1 to 5) as evaluated by GPT-4 Turbo, with the question and response model being random as well. 2 of the authors and one graduate student who volunteered evaluate 46, 46, and 48

[11] `openrouter.ai`

[12] `litellm.vercel.app/docs/providers/openrouter`

Model	Average Forward Discrepancy			Value			Frequency (%)		
	$D_2 \rightarrow D_3$	$D_1 \rightarrow D_2$	Overall	$D_2 \rightarrow D_3$	$D_1 \rightarrow D_2$	Overall	$D_2 \rightarrow D_3$	$D_1 \rightarrow D_2$	Overall
LLaMA 2 7B Chat	0.1304	0.1814	0.1756	0.2708	0.2683	0.2685	48.15	67.62	65.40
LLaMA 2 13B Chat	0.1524	0.1582	0.1573	0.2572	0.2720	0.2697	59.26	58.14	58.31
LLAMA 2 70B Chat	0.1259	0.1361	0.1344	0.2633	0.2490	0.2512	47.83	54.68	53.50
Mistral 7B Instruct v0.2	0.0920	0.1569	0.1474	0.2031	0.2294	0.2267	45.28	68.39	65.01
Mixtral 8x7B Instruct v0.1	0.0868	0.0791	0.0806	0.1844	0.2058	0.2009	47.06	38.46	40.14
Llama 3 8B Instruct	0.0831	0.0957	0.0934	0.2225	0.2258	0.2253	37.33	42.38	41.44
Llama3 70B Instruct	0.0653	0.0497	0.0528	0.2176	0.2211	0.2202	30.00	22.47	23.99
GPT-3.5 Turbo	0.1002	0.0722	0.0779	0.1608	0.1369	0.1424	62.35	52.73	54.70

Table 12: Average intensity and frequency of forward discrepancy.

Model	Average Backward Discrepancy			Value			Frequency (%)		
	$D_2 \rightarrow D_3$	$D_1 \rightarrow D_2$	Overall	$D_2 \rightarrow D_3$	$D_1 \rightarrow D_2$	Overall	$D_2 \rightarrow D_3$	$D_1 \rightarrow D_2$	Overall
LLaMA 2 7B Chat	0.2193	0.1104	0.1342	0.3827	0.3589	0.3671	57.31	30.77	36.57
LLaMA 2 13B Chat	0.1255	0.0782	0.0879	0.3846	0.3339	0.3473	32.64	23.43	25.32
LLAMA 2 70B Chat	0.1363	0.0632	0.0787	0.3811	0.3258	0.3442	35.76	19.40	22.88
Mistral 7B Instruct v0.2	0.1442	0.0700	0.0881	0.3488	0.3071	0.3225	41.33	22.81	27.31
Mixtral 8x7B Instruct v0.1	0.0627	0.0635	0.0633	0.2979	0.2728	0.2781	21.04	23.27	22.76
Llama 3 8B Instruct	0.0878	0.0717	0.0752	0.3500	0.3141	0.3227	25.08	22.82	23.32
Llama3 70B Instruct	0.0427	0.0442	0.0438	0.2778	0.2692	0.2710	15.38	16.41	16.18
GPT-3.5 Turbo	0.0457	0.0672	0.0626	0.2892	0.2602	0.2644	15.79	25.81	23.68

Table 13: Average intensity and frequency of backward discrepancy.

Question depth	Human-Human	Human-GPT-4
D_3	0.4848 (n = 3)	0.7064 (n = 13)
D_2	0.6464 (n = 6)	0.7730 (n = 32)
D_1	0.5671 (n = 11)	0.7969 (n = 55)
Overall	0.5730 (n = 20)	0.7797 (n = 100)

Table 14: Krippendorf's Alpha between human-human and human-GPT-4 ratings on model responses to DEPTHQA questions. For human-GPT agreement, the scores of predictions rated by the three human raters are averaged. The number of responses in each measurement is reported below the Krippendorf's Alpha value.

unique responses, respectively, and all 3 workers label the remaining 20 responses set aside for inter-annotator agreement. The human raters are given only 1 instance at a time and individually score it on a scale of 1 to 5, under the exact setting of our LLM-as-a-Judge experiments. The evaluation interface is shown in Figure 8. Following Ye et al. (2024), we measure Krippendorf's Alpha (Krippendorff, 2018; Castro, 2017) with an ordinal metric to reliability between three human raters and between humans and GPT-4 Turbo.

Table 14 reports the agreement results. The results show that the human-GPT agreement is substantially high, approaching 0.80, the commonly accepted reliability threshold (Krippendorff, 2018). While the sample size is smaller, there is also moderate human-human agreement. This implies that the individual absolute rating scheme is effective and that GPT-4 Turbo evaluations are aligned with humans in our setting.

G Discrepancy Results

To separately observe how frequently each discrepancy occurs and its intensity when it happens, Table 12 and Table 13 show the average intensity and frequency of each forward and backward discrepancy. Note that the average discrepancy is calculated as the product of the value and frequency. Overall, forward discrepancies appeared more frequently, although their intensity was relatively low (between 0.14 and 0.26). In contrast, backward discrepancies appeared less than 25%, except for LLaMA 2 7B, which exhibited high intensity (between 0.26 and 0.37).

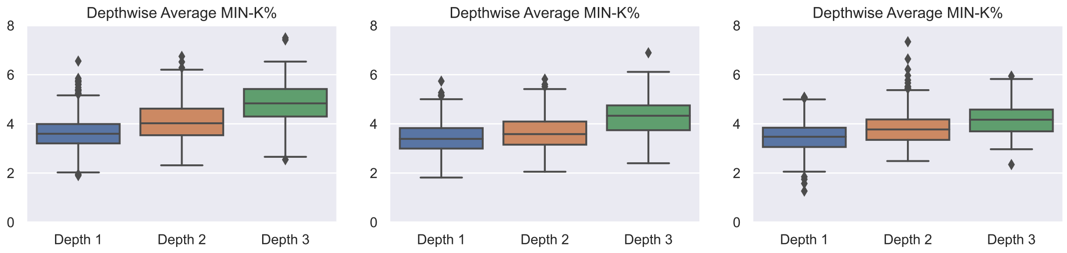
H Overall Results with Min-K% Probability

H.1 Depthwise Min-K% Prob.

In Figure 5, we plot the Min-k% probability of LLaMA 2 13B Chat, Mistral 8B Instruct and Mixtral 8x7B Instruct. Similar to Figure 3, D_3 shows the highest average Min-K% probability, indicating the least memorization over all three models.

H.2 Score Gap within Neighboring Questions

Figure 6 presents the KDE plot of the factual accuracy gap between q_3 and q_2 for q_3 instances whose

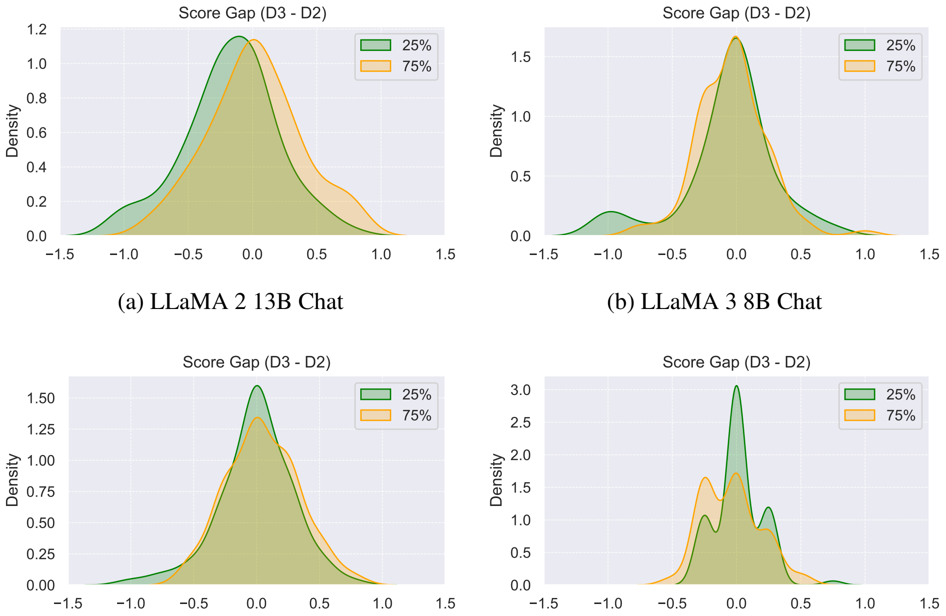


(a) LLaMA 2 13B Chat

(b) Mistral 8B Instruct

(c) Mixtral 8x7B Instruct

Figure 5: Average Min-K% probability at each depth. Lower values indicate more memorization while higher values indicate less memorization.



(a) LLaMA 2 13B Chat

(b) LLaMA 3 8B Chat

(c) Mixtral 8x7B Instruct

(d) Mixtral 8x7B Instruct

Figure 6: Factual accuracy difference between neighboring q_3 and q_2 in bottom 25% and top 75% quantiles. Positive gap indicates backward discrepancy and negative gap represents forward discrepancy.

Min-%K probability is in the bottom 25% and top 75%. A positive gap represents higher factual accuracy for q_3, indicating backward discrepancy. In contrast, a negative difference represents forward discrepancy.

I Backward Discrepancy Examples

Different backward discrepancy examples are shown in Example 1, 2, and 3.

J Prompts

J.1 Data construction

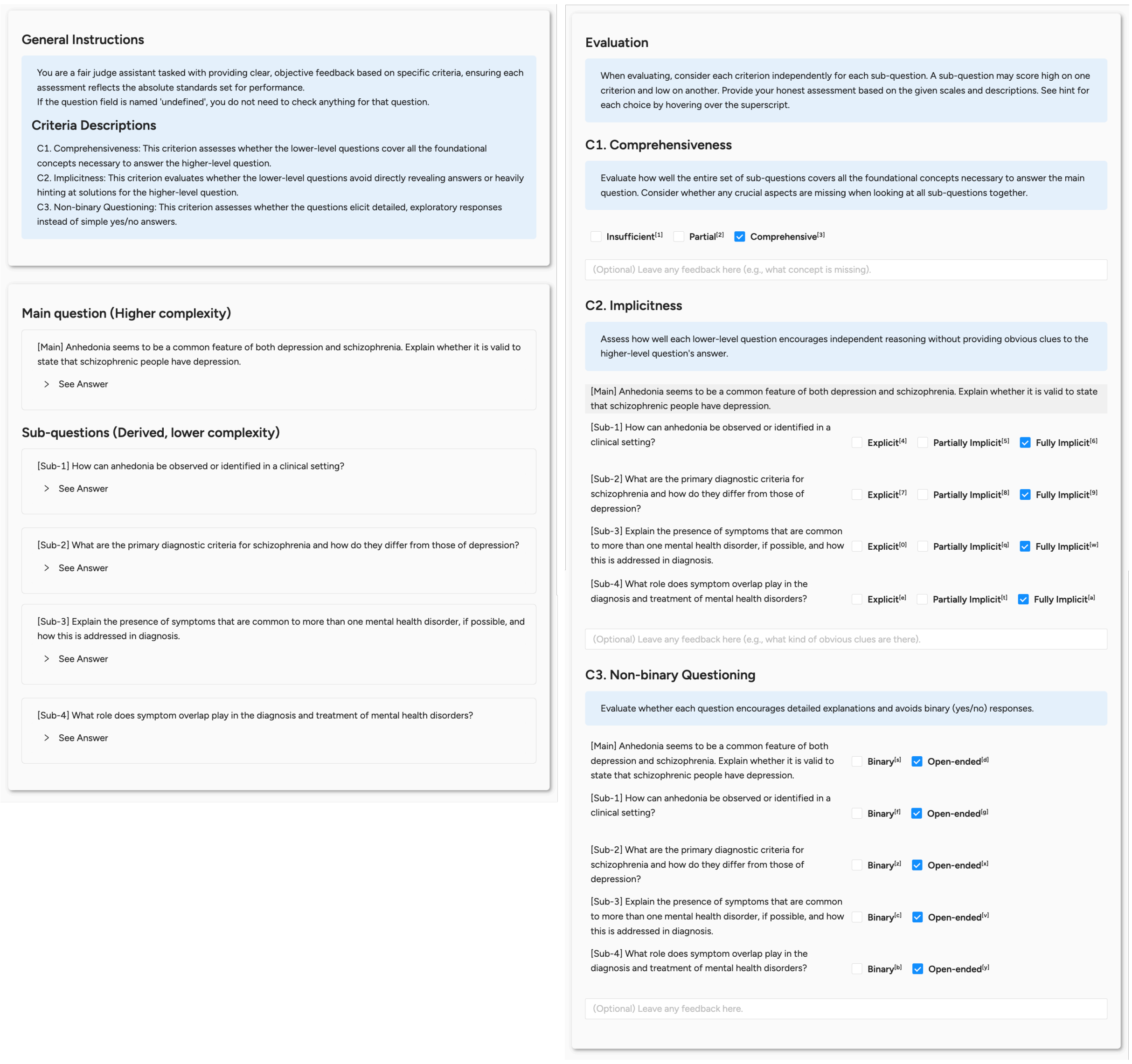
We provide the prompts used to classify TutorEval questions (Table 19), generate D_3 answers (Table 20), generate D_2 or D_2 answers (Table 21), generate questions at D_2 (Table 23) and D_1 (Table 24), and augment questions at D_2 (Table 25) and D_1 (Table 26). For generating or augmenting any question at D_2 or D_1, we use the same system prompt (Table 22) that describes the definitions of depths of knowledge.

J.2 Inference

We provide the prompts used for zero-shot (Table 27), Prompt (Gold) and Prompt (Pred.) (Table 28), and multi-turn (Table 29) inference.

J.3 Evaluation

The prompt used for LLM-as-a-Judge evaluation is in Table 30.



General Instructions

You are a fair judge assistant tasked with providing clear, objective feedback based on specific criteria, ensuring each assessment reflects the absolute standards set for performance.
If the question field is named 'undefined', you do not need to check anything for that question.

Criteria Descriptions

C1. Comprehensiveness: This criterion assesses whether the lower-level questions cover all the foundational concepts necessary to answer the higher-level question.
C2. Implicitness: This criterion evaluates whether the lower-level questions avoid directly revealing answers or heavily hinting at solutions for the higher-level question.
C3. Non-binary Questioning: This criterion assesses whether the questions elicit detailed, exploratory responses instead of simple yes/no answers.

Main question (Higher complexity)

[Main] Anhedonia seems to be a common feature of both depression and schizophrenia. Explain whether it is valid to state that schizophrenic people have depression.

> See Answer

Sub-questions (Derived, lower complexity)

[Sub-1] How can anhedonia be observed or identified in a clinical setting?

> See Answer

[Sub-2] What are the primary diagnostic criteria for schizophrenia and how do they differ from those of depression?

> See Answer

[Sub-3] Explain the presence of symptoms that are common to more than one mental health disorder, if possible, and how this is addressed in diagnosis.

> See Answer

[Sub-4] What role does symptom overlap play in the diagnosis and treatment of mental health disorders?

> See Answer

Evaluation

When evaluating, consider each criterion independently for each sub-question. A sub-question may score high on one criterion and low on another. Provide your honest assessment based on the given scales and descriptions. See hint for each choice by hovering over the superscript.

C1. Comprehensiveness

Evaluate how well the entire set of sub-questions covers all the foundational concepts necessary to answer the main question. Consider whether any crucial aspects are missing when looking at all sub-questions together.

☐ Insufficient[1] ☐ Partial[2] ☑ Comprehensive[3]

(Optional) Leave any feedback here (e.g., what concept is missing).

C2. Implicitness

Assess how well each lower-level question encourages independent reasoning without providing obvious clues to the higher-level question's answer.

[Main] Anhedonia seems to be a common feature of both depression and schizophrenia. Explain whether it is valid to state that schizophrenic people have depression.

[Sub-1] How can anhedonia be observed or identified in a clinical setting?	☐ Explicit[4]	☐ Partially Implicit[5]	☑ Fully Implicit[6]
[Sub-2] What are the primary diagnostic criteria for schizophrenia and how do they differ from those of depression?	☐ Explicit[7]	☐ Partially Implicit[8]	☑ Fully Implicit[9]
[Sub-3] Explain the presence of symptoms that are common to more than one mental health disorder, if possible, and how this is addressed in diagnosis.	☐ Explicit[0]	☐ Partially Implicit[q]	☑ Fully Implicit[w]
[Sub-4] What role does symptom overlap play in the diagnosis and treatment of mental health disorders?	☐ Explicit[e]	☐ Partially Implicit[t]	☑ Fully Implicit[a]

(Optional) Leave any feedback here (e.g., what kind of obvious clues are there).

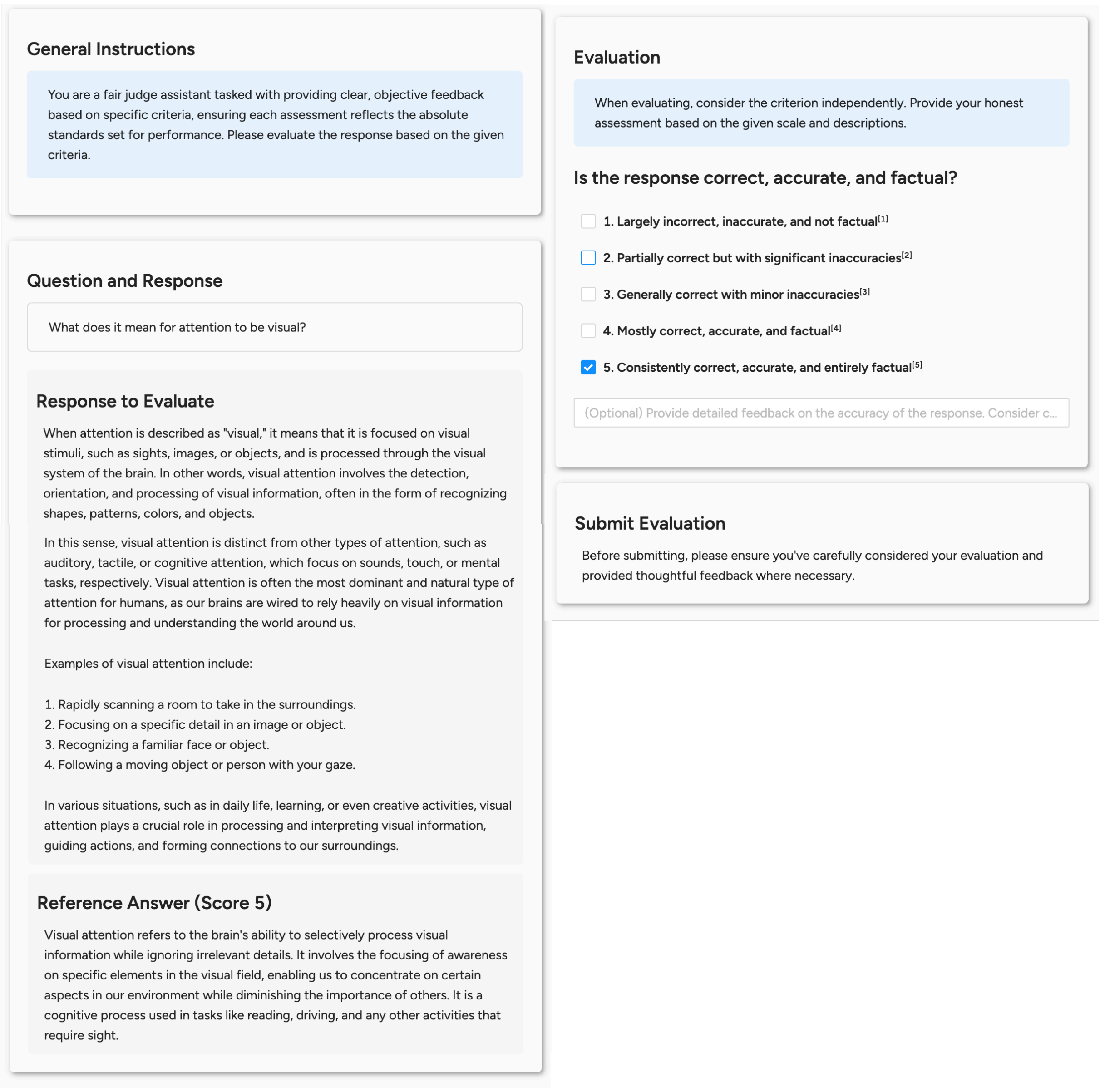
C3. Non-binary Questioning

Evaluate whether each question encourages detailed explanations and avoids binary (yes/no) responses.

[Main] Anhedonia seems to be a common feature of both depression and schizophrenia. Explain whether it is valid to state that schizophrenic people have depression.	☐ Binary[s]	☑ Open-ended[d]
[Sub-1] How can anhedonia be observed or identified in a clinical setting?	☐ Binary[f]	☑ Open-ended[g]
[Sub-2] What are the primary diagnostic criteria for schizophrenia and how do they differ from those of depression?	☐ Binary[z]	☑ Open-ended[x]
[Sub-3] Explain the presence of symptoms that are common to more than one mental health disorder, if possible, and how this is addressed in diagnosis.	☐ Binary[c]	☑ Open-ended[v]
[Sub-4] What role does symptom overlap play in the diagnosis and treatment of mental health disorders?	☐ Binary[b]	☑ Open-ended[y]

(Optional) Leave any feedback here.

Figure 7: Interface for human annotators to check if Comprehensiveness (C1), Implicitness (C2), Non-binary questioning (C3) hold between a question and its sub-questions in DEPTHQA.



General Instructions

You are a fair judge assistant tasked with providing clear, objective feedback based on specific criteria, ensuring each assessment reflects the absolute standards set for performance. Please evaluate the response based on the given criteria.

Question and Response

What does it mean for attention to be visual?

Response to Evaluate

When attention is described as "visual," it means that it is focused on visual stimuli, such as sights, images, or objects, and is processed through the visual system of the brain. In other words, visual attention involves the detection, orientation, and processing of visual information, often in the form of recognizing shapes, patterns, colors, and objects.

In this sense, visual attention is distinct from other types of attention, such as auditory, tactile, or cognitive attention, which focus on sounds, touch, or mental tasks, respectively. Visual attention is often the most dominant and natural type of attention for humans, as our brains are wired to rely heavily on visual information for processing and understanding the world around us.

Examples of visual attention include:

1. Rapidly scanning a room to take in the surroundings.
2. Focusing on a specific detail in an image or object.
3. Recognizing a familiar face or object.
4. Following a moving object or person with your gaze.

In various situations, such as in daily life, learning, or even creative activities, visual attention plays a crucial role in processing and interpreting visual information, guiding actions, and forming connections to our surroundings.

Reference Answer (Score 5)

Visual attention refers to the brain's ability to selectively process visual information while ignoring irrelevant details. It involves the focusing of awareness on specific elements in the visual field, enabling us to concentrate on certain aspects in our environment while diminishing the importance of others. It is a cognitive process used in tasks like reading, driving, and any other activities that require sight.

Evaluation

When evaluating, consider the criterion independently. Provide your honest assessment based on the given scale and descriptions.

Is the response correct, accurate, and factual?

☐ 1. Largely incorrect, inaccurate, and not factual[1]

☐ 2. Partially correct but with significant inaccuracies[2]

☐ 3. Generally correct with minor inaccuracies[3]

☐ 4. Mostly correct, accurate, and factual[4]

☑ 5. Consistently correct, accurate, and entirely factual[5]

(Optional) Provide detailed feedback on the accuracy of the response. Consider c...

Submit Evaluation

Before submitting, please ensure you've carefully considered your evaluation and provided thoughtful feedback where necessary.

Figure 8: Interface for human evaluators to evaluate an LLM's response on a question from DEPTHQA. The rubric shown is a simplified form of the actual factual accuracy rubric used in LLM evaluations.

Reasoning Type	Explanation	Example
Comparative	Compare two or more concepts, identifying similarities and differences.	D2 How do neutrinos differ from other subatomic particles, and why are they considered potential candidates for dark matter? D1 What are neutrinos? D1 What are subatomic particles? D1 What is dark matter? D1 What characteristics do particles need to be considered candidates for dark matter?
Relational	Specify and explain relationships and understand how different concepts are connected organically.	D2 Describe how eco-efficient urban planning can address the challenges of rapid urbanization in developing countries. D1 What is urbanization? D1 What does eco-efficient mean? D1 What are common challenges faced by rapidly urbanizing cities in developing countries? D1 What is urban planning?
Causal	Identify cause-and-effect relationships.	D2 Explain how the bending of stereocilia on hair cells leads to the depolarization of these cells. D1 What are stereocilia? D1 What is depolarization? D1 Where are hair cells located? D1 What is the function of hair cells in the ear?
Inductive	Make broad generalizations from specific observations and/or formulate a hypothesis about a particular concept.	D3 Can you sum up the point of connecting finite sums to integrals? This concept is still a bit obscure to me. D2 How do you approximate the area under a curve using rectangles or trapezoids? D2 Explain the process of taking the limit of a sum as the number of rectangles increases to infinity. D2 What is a Riemann sum, and how is it related to the concept of an integral? D2 How can finite sums be used to estimate real-world quantities that change continuously over an interval?
Criteria Development	Understand when and why specific criteria apply, and know the conditions or assumptions required for different cases.	D2 Under what conditions does the ideal gas law provide accurate predictions, and when does it not? D1 What is the ideal gas law equation? D1 What are the standard conditions for temperature and pressure in experiments? D1 What is meant by 'ideal gas'? D1 How do real gases differ from ideal gases?
Procedural	Select a procedure according to task need and perform it.	D2 Describe the process by which hair cells transduce mechanical energy from sound waves into electrical signals. D1 What are hair cells? D1 What is mechanical energy? D1 What are sound waves? D1 What are electrical signals?
Evaluative	Verify reasonableness of results.	D3 How can I evaluate the suitability of the ideal gas equation for a given gas? D2 How do you calculate the properties such as pressure, temperature, and volume using the ideal gas law? D2 What methods can be used to obtain experimental data for gas properties under specific conditions? D2 How can deviations from ideal gas behavior be identified and measured? D2 Under what conditions does the ideal gas law provide accurate predictions, and when does it not?
Example	Provide example for the given concept.	D2 Describe a scenario where energy is conserved but the process is thermodynamically impossible. D1 What does the law of conservation of energy state? D1 What is thermodynamic impossibility? D1 What is meant by energy conversion? D1 Can energy be created or destroyed?
Quantitative	Manipulate numerical data to make informed decisions.	D2 Explain the process and time complexity of deleting an element from a data structure like a linked list. D1 What is a linked list? D1 What is the definition of time complexity? D1 How is data stored in a linked list? D1 What does 'deleting an element' mean in the context of data structures?
Application	Apply concepts to practical situations.	D2 What policies can governments implement to encourage the transition towards a circular economy and sustainable business practices? D1 What is a circular economy? D1 What are sustainable business practices? D1 What is the role of government in regulating the economy? D1 What does the term 'policy' mean in the context of government regulation?

Table 15: Reasoning type explanation and examples. D_3, D_2, and D_1 questions are denoted as D3, D2, D1, respectively.

- D3 Does a matrix always have a basis of eigenvectors?
  - D2 How can you determine if a square matrix is diagonalizable?
    - D1 What is the definition of a square matrix?
    - D1 What are the characteristics of a diagonal matrix?
    - D1 What is meant by the eigenvalues of a matrix?
    - D1 How is the characteristic equation of a matrix defined?
  - D2 What is the process for finding the eigenvalues of a matrix?
  - D2 Explain how to compute eigenvectors from a given set of eigenvalues.
  - D2 Describe the method to perform a similarity transformation on a matrix.

Table 16: Snippet of hierarchical question deconstruction for analyzing matrix diagonalizability. The topmost complex question, D_3, requires developing criteria of whether the statement holds or not. The first D_2 question identifies the key property to help determine the case. The D_1 child questions addresses relevant definitions, characteristics, and formula in order to synthesize the foundational concepts.

- D3 I thought that animals from different species could not produce viable offspring. However, a horse and a donkey can produce viable offspring. Isn't this contradictory?
  - D2 How do horses and donkeys differ genetically and what makes them separate species?
  - D2 What is the definition of a viable offspring in biological terms?
  - D2 Can you explain the concept of hybrid vigor and how it applies to cross-species offspring such as mules?
    - D1 What is the definition of hybrid vigor?
    - D1 What is a mule?
    - D1 How is a mule produced?
    - D1 What are the parent species of a mule?
  - D2 What are the implications of sterility in hybrids like mules for their classification as members of a species?

Table 17: Snippet of hierarchical question deconstruction for understanding species differentiation and hybrid viability. Key reasoning steps include identifying exceptions, elucidating causal relationships, and hypothesizing potential outcomes in species classification criteria, culminating in a robust evaluation of biological definitions and exceptions.

Question:
I'm studying ideal and real gasses in engineering thermodynamics. Why can't we measure temperature in Fahrenheit?

Predicted answer:
In engineering thermodynamics, it's common to work with ideal and real gases, and temperature plays a crucial role in characterizing their behavior.

Actually, you can measure temperature in Fahrenheit, but not in the context of the ideal gas law or ideal gas equations. The issue lies in the definition and scaling of the Fahrenheit and Celsius temperature scales.

Celsius is the preferred choice for scientific and engineering applications because it's based on the concept of absolute zero, which is the theoretically lowest possible temperature. It's defined as 0 K (kelvin). This allows for a coherent and consistent framework for discussing temperature and thermodynamic quantities.

Fahrenheit, on the other hand, has a different zero point (32°F) and relies on arbitrary reference points to define its scale. Although it's still used for everyday applications, it doesn't provide the same theoretical foundation as Celsius.

The ideal gas law and the equations of state for gases are typically expressed in terms of absolute temperature (K or °R), and Celsius is the most compatible scale for these applications.

So, while you could measure temperature in Fahrenheit if needed, it's essential to understand the underlying science and choose the most relevant and consistent temperature scale when working with ideal and real gases in engineering thermodynamics.

Feedback:
Feedback: The response provides an accurate explanation regarding the use of Fahrenheit versus Celsius (and indirectly Kelvin) in the context of thermodynamics and ideal gases. It correctly identifies the reasons why Celsius and Kelvin are preferred for scientific studies, focusing on the relevance of absolute zero and the uniformity in scientific communication. However, it slightly misleads by stating that Celsius is based on the concept of absolute zero, when in fact, it is Kelvin that is based on this concept. Celsius is indeed more aligned with Kelvin compared to Fahrenheit but it is not based on absolute zero; rather, it is defined by the melting point of ice and the boiling point of water at one atmosphere of pressure, which are then used to extrapolate its relation to Kelvin. The response demonstrates a strong understanding but includes a minor inaccuracy regarding the basis of the Celsius scale. This inaccuracy does not significantly detract from the overall quality and factual correctness of the explanation but is worth noting for precision.

Score:
4

Table 18: Example LLaMA 8B Instruct zero-shot inference on a D_3 question followed by GPT-4 Turbo evaluation.

Example 1: Common Backward Discrepancy

D3 **question:** I understand the point of the tangle number. But why go to the extra lengths of considering the continued fraction representation of the tangle number.

LLaMA 2 7B Chat Predicted Answer (Score: 4)

(...) *The continued fraction representation of a tangle number allows us to represent the tangle as a rational number*, which can be easily manipulated and analyzed. This can be particularly useful when studying the properties of tangles, such as their convergence properties, their behavior under certain operations, and their relationship to other mathematical structures. (...)

LLaMA 3 70B Instruct Predicted Answer (Score: 4)

(...) By looking at the continued fraction representation, we can gain insight into the tangle's underlying topology and geometry. In particular, the continued fraction representation can reveal information about the tangle's symmetry, its decomposition into simpler components, and even its relationship to other tangles. (...)

D2 **question:** What is the process of constructing a tangle for a given rational number?

LLaMA 2 7B Chat Predicted Answer (Score: 1)

(...) Choose a base: There are many different bases that you can use to construct a tangle for a rational number. Some common bases include *the golden ratio, the Fibonacci sequence, and the square root of 2*. Choose a base that you feel comfortable working with and that will help you visualize the tangle. (...)

Feedback: Introduces unrelated concepts that are not part of the accepted method for constructing tangles based on rational numbers.

LLaMA 3 70B Instruct Predicted Answer (Score: 2)

(...) *Draw p arcs*: Draw p arcs from the top of the circle to the bottom, each crossing the circle at a single point. These arcs will form the strands of the tangle.
Twist the arcs: Twist the arcs around each other according to the following rule: for each of the q crossings, take the upper strand and pass it over or under the lower strand (it doesn't matter which) to create a twist. (...)

Feedback: Describes a process that diverges from the correct method of using continued fractions to construct a tangle. It also omits the critical step of expressing the rational number as a continued fraction and constructing the tangle diagram based on the continued fraction's terms.

Example 2: Backward Discrepancy by Small Model

D3 **question:** Given that node operations are O(1), explain whether this implies that all operations in data structures using nodes will also have O(1) complexity, and why or why not.

LLaMA 2 7B Chat Predicted Answer (Score: 5)

(...) *just because node operations have a constant time complexity doesn't necessarily mean that all operations in data structures using nodes will also have O(1) complexity*. The reason for this is that the time complexity of an operation can depend on many factors beyond just the size of the input, such as the data structure's internal organization, the algorithms used, and the hardware on which the data structure is running. (...)

D2 **question:** How does the insertion operation work in a linked list, and what factors affect its time complexity?

LLaMA 2 7B Chat Predicted Answer (Score: 2)

(...) Here are the common ways to perform insertion in a linked list:
1. *Linear Insertion*: (...) This approach has a time complexity of *O(1)* because we only need to traverse the list once to find the correct position. 2. *Shift-Insertion*: (...) This approach has a time complexity of O(n) because we need to traverse the entire list to shift all the nodes. 3. *Rotate-Insertion*: (...) This approach has a time complexity of O(n) because we need to traverse the entire list to rotate all the nodes. (...)

Feedback: Introduces non-standard terms ("Linear Insertion," "Shift-Insertion," "Rotate-Insertion") that are not typically used in describing linked list operations. Incorrectly states that insertion at a specific position has O(1) time complexity, when it should be O(n) in the worst case.

Example 3: Backward Discrepancy by Large Model

D3 **question:** The quadratic formula is pretty cool. What similar formulas, if any, exist for higher degree equations?

LLaMA 3 70B Instruct Predicted Answer (Score: 5)

(...) For cubic equations of the form $ax^3 + bx^2 + cx + d = 0$, there is a formula known as *Cardano's Formula* (or *Cardano's Method*). (...) For quartic equations of the form $ax^4 + bx^3 + cx^2 + dx + e = 0$, there is a formula known as *Ferrari's Method* (or *Ferrari's Solution*). (...) For equations of degree 5 or higher, there is no general formula that can be written using only addition, subtraction, multiplication, division, and nth roots (like the quadratic formula). This was proven by Niels Henrik Abel in 1824, and later independently by Évariste Galois in 1832. (...)

D2 **question:** Explain the steps to solve quartic equations using the quartic formula.

LLaMA 3 70B Instruct Predicted Answer (Score: 1)

The quartic formula! While it's not as well-known as the quadratic formula, it's a powerful tool for solving quartic equations of the form $ax^4 + bx^3 + cx^2 + dx + e = 0$. Here are the steps to solve quartic equations using the quartic formula:
Steps to Solve Quartic Equations:
1. *Write down the quartic equation in standard form*: $ax^4 + bx^3 + cx^2 + dx + e = 0$, where a, b, c, d, and e are constants.
2. *Rearrange the equation to have a = 1*: Divide both sides of the equation by a, so that the coefficient of x^4 is 1. This simplifies the formula.
3. *Calculate the values of p, q, and r*:
- $p = b^2 - 3c$
- $q = bc - 4d - a$
- $r = c^2 - 3bd + 12ae$
4. *Calculate the value of u*: $u = (r/2 \pm \sqrt{(r/2)^2 + (q/3)^3})^(1/3)$
5. *Calculate the value of v*: $v = (q + u - b^2/3)/(3u)$
6. *Calculate the value of w*: $w = (-b \pm \sqrt{b^2 - 4v})/2$
7. *Solve for x*: $x = (-b \pm 2w \pm \sqrt{w^2 - 4u})/2$

Feedback: Introduces an incorrect formula for solving quartic equations, which does not align with established methods like Ferrari's solution. The provided steps and calculations do not correspond to the correct procedure for solving quartic equations using the quartic formula or any other recognized method.

System prompt:
You are an excellent question classifier. You will be given (1) a question and (2) key points that a good response would address when answering the question. You have to classify at which Depth of Knowledge (DOK) level the question is located. DOK is a framework that focuses on the context which knowledge will be demonstrated. Here is the definition of each DOK level:

1. DOK-1 (Basic Knowledge and Recall): This level addresses “What is the knowledge?”. It evaluates the ability to remember, explain, or pinpoint fundamental facts, terms, principles, and procedures. It’s about recognizing or recollecting basic information and performing simple, direct tasks.
2. DOK-2 (Application of Knowledge and Skills): This level explores “How can the knowledge be used?”. It tests the ability to employ knowledge and concepts in practical situations, which involves choosing appropriate methods, solving straightforward problems, or interpreting data. This level acts as an intermediary step between fundamental understanding and more advanced reasoning.
3. DOK-3 (Analytical and Strategic Thinking): This level questions “Why can the knowledge be used?”. It challenges one to use strategic thought, logic, and problem-solving in intricate, abstract situations that might have more than one solution. This stage demands critical thinking, rationale, and conceptualization of theoretical scenarios.
4. DOK-4 (Extended and Integrative Knowledge): This level examines “How else can the knowledge be applied?”. It assesses the ability to conduct thorough research, apply concepts and skills in real-world scenarios, and integrate knowledge across different disciplines or sources. It involves developing original ideas, conducting experiments, and synthesizing information from various fields. Note that in the science domain, this level may be constrained to designing studies, experiments, and projects and is thus rare or even absent in most standardized assessment.

User prompt:
Please classify the following question into DOK-1, 2, 3, or 4. Refer to the key points to help your judgment. Think step-by-step and provide an explanation of your judgment. After providing your explanation, output the DOK level that is an integer of 1, 2, 3, or 4. The output format should looks as follows: {explanation for reaching the DOK decision} [RESULT]{DOK level that is an integer in the range 1 to 4}.

Question
{question}
Key points
{key_points}
Answer

Table 19: Prompt for classifying TutorEval questions.

System prompt:
You are an excellent assistant that effectively answers complex questions. You are given a passage, question, and key points to answer the question. Read the instruction and give an appropriate answer.

User prompt:
Chapter
{chapter}

Instruction
Answer the question below.
- You may refer to the contents in the chapter text above if necessary, but do NOT expose in your answer that you are referring to the provided source.
- Ensure that the answer is complete, fully satisfying the key points to answer the question.
- The answer must also match the level of cognitive complexity required, incorporating the context which the depth of knowledge will be demonstrated.

Question
{question}

Key points to answer the question
{key_points}

Complexity of the question
{explanation}

Answer

Table 20: Prompt for generating reference answer for a D_3 question.

System prompt:
You are a helpful assistant that accurately answers complex questions. Ensure that your answer is focused and compact.

User prompt:
`{question}`

Table 21: Prompt for generating reference answer for a D_1 or D_2 question.

System prompt:
You are an excellent question generator. You will be given a question and a gold answer to the question. You have to generate shallower questions from the given question. Here is the definition of the depth of knowledge a question requires:

1. Depth-1 (Basic Knowledge and Recall): This level addresses “What is the knowledge?”. It evaluates the ability to remember, explain, or pinpoint fundamental facts, terms, principles, and procedures. It’s about recognizing or recollecting basic information and performing simple, direct tasks.
2. Depth-2 (Application of Knowledge and Skills): This level explores “How can the knowledge be used?”. It tests the ability to employ knowledge and concepts in practical situations, which involves choosing appropriate methods, solving straightforward problems, or interpreting data. This level acts as an intermediary step between fundamental understanding and more advanced reasoning.
3. Depth-3 (Analytical and Strategic Thinking): This level questions “Why can the knowledge be used?”. It challenges one to use strategic thought, logic, and problem-solving in intricate, abstract situations that might have more than one solution. This stage demands critical thinking, rationale, and conceptualization of theoretical scenarios.

Table 22: System prompt for generating or augmenting D_1 or D_2 questions.

User prompt:

Instruction

Create maximum of 4 Depth-2 questions that are necessary to answer the provided Depth-3 question correctly.

- Remember that Depth-2 questions are centered on application of procedural knowledge and skills and Depth-3 questions are centered on analysis and strategic knowledge.
- Take into consideration the level of cognitive complexity required to solve the Depth-3 question, so that your generated questions fall under the description of Depth-2 appropriately.
- Ensure that your collection of generated Depth-2 questions adequately and comprehensively covers ALL the necessary factual or conceptual knowledge required to answer the given Depth-3 question.
- Ensure that all of your generated Depth-2 questions do not directly answer to the given Depth-3 question.
- The number of generated Depth-2 questions should not exceed 4.
- The generated Depth-2 questions should be in JSON format: {“Depth-2_questions”: [list of Depth-2 question strings]}

Example 1

Depth-3 question

What is the intuition behind the Gram - Schmidt procedure?

Generated Depth-2 questions

{“Depth-2_questions”: [’How do you project one vector onto another vector?’, ’What does it mean for two vectors to be orthogonal, and how can you verify this property?’, ’Describe the process of normalizing a vector.’, ’Explain how subtracting the projection of one vector from another results in orthogonality.’, ’Given a set of vectors, how can you determine if they are linearly independent?’, ’How can the concept of linear independence be used to form a basis for a vector space?’]}

Example 2

Depth-3 question

Why couldn’t we test general relativity effects using the Eotvos experiment?

Generated Depth-2 questions

{“Depth-2_questions”: [“How does the Eötvös experiment determine the equivalence between inertial mass and gravitational mass?”, “Describe the Equivalence Principle and its significance in the theory of General Relativity.”, “Identify experiments or observations that could directly test the predictions of General Relativity, such as time dilation or the bending of light.”, “How do experiments measuring time dilation differ in design and scope from those measuring mass equivalence?”]}

Example 3

Depth-3 question

Why are aldehydes more readily oxidized to carboxylic acids compared to ketones, and how does this difference in reactivity influence their identification in the laboratory?

Generated Depth-2 questions

{“Depth-2_questions”: [“How can you identify an aldehyde using Tollens’ reagent?”, “Why does the carbonyl carbon in aldehydes have a significant partial positive charge?”, “How does the structure of ketones differ from that of aldehydes, and how does this affect their reactivity towards oxidation?”]}

Example 4

Depth-3 question

In the context of computer programming, why is branching unstructured? And is it a bad design choice?

Generated Depth-2 questions

{“Depth-2_questions”: [“What are the key differences between structured and unstructured branching in programming?”, “How does the ’goto’ statement work in computer programming?”, “What are the potential risks involved with using unstructured branching in large software projects?”, “How does the structure of a program affect its maintainability?”, “How can the flow of execution in a program influence its debuggability?”]}

Depth-3 question

`{question}`

Answer to the Depth-3 question

`{answer}`

Generated Depth-2 questions

Table 23: User prompt for generating D_2 questions.

User prompt:

Instruction

Create maximum of 4 Depth-1 questions that are necessary to answer the provided Depth-2 question correctly.

- Remember that Depth-1 questions are centered on basic recall of factual and conceptual knowledge. Depth-2 questions are centered on application of procedural knowledge and skills.
- Take into consideration the level of cognitive complexity required to solve the Depth-2 question, so that your generated questions fall under the description of Depth-1 appropriately.
- Ensure that your collection of generated Depth-1 questions adequately and comprehensively covers ALL the necessary factual or conceptual knowledge required to answer the given Depth-2 question.
- Ensure that all of your generated Depth-1 questions do not directly answer to the given Depth-2 question.
- Try to exclude Depth-1 questions that ask too generic or commonsense knowledge.
- The number of generated DOK-2 questions should not exceed 4.
- The generated Depth-1 questions should be in JSON format: {“Depth-1_questions”: [list of Depth-1 question strings]}

Example 1

Depth-2 question

How can the concept of algebraic closure be demonstrated using polynomial equations with complex roots?

Generated Depth-1 questions

{“Depth-1_questions”: [’What is the definition of algebraic closure?’, ’What is a polynomial equation?’, ’What are complex roots in the context of polynomial equations?’, ’How can complex roots be represented?’]}

Example 2

Depth-2 question

How do you perform a convolution operation between two random variables?

Generated Depth-1 questions

{“Depth-1_questions”: [’What is a convolution operation?’, ’What is a random variable?’, ’How is the product of two functions calculated?’, ’What does it mean to integrate a function?’]}

Example 3

Depth-2 question

In what ways can a decision tree’s structure be represented programmatically?

Generated Depth-1 questions

{“Depth-1_questions”: [’What is a decision tree in the context of programming?’, ’What are the basic components of a decision tree?’, ’What is a data structure in programming?’, “What does ’represented programmatically’ mean?”]}

Example 4

Depth-2 question

How do neutrinos differ from other subatomic particles, and why are they considered potential candidates for dark matter?

Generated Depth-1 questions

{“Depth-1_questions”: [’What are neutrinos?’, ’What are subatomic particles?’, ’What is dark matter?’, ’What characteristics do particles need to be considered candidates for dark matter?’]}

Depth-2 question

{question}

Answer to the Depth-2 question

{answer}

Generated Depth-1 questions

Table 24: User prompt for generating D_1 questions.

User prompt:
Instruction
Create {count} Depth-2 question(s) that complement current Depth-2 questions, which are necessary to correctly answer the provided Depth-3 question.
- Remember that Depth-2 questions are centered on application of procedural knowledge and skills and Depth-3 questions are centered on analysis and strategic knowledge.
- Take into consideration the level of cognitive complexity required to solve the Depth-3 question, so that your generated questions fall under the description of Depth-2 appropriately.
- Complement the existing Depth-2 questions with additional questions to ensure they collectively cover all necessary procedural knowledge and skills required to answer the Depth-3 question effectively.
- Ensure that all of your generated Depth-2 questions do not directly answer to the given Depth-3 question.
- The number of all Depth-2 questions should not exceed 4.
- The generated Depth-2 questions should be in JSON format: {“Depth-2_questions”: [list of Depth-2 question strings]}

Example 1
Depth-3 question and current Depth-2 questions
What is the intuition behind the Gram - Schmidt procedure?
{“current_Depth-2_questions”: [’How do you project one vector onto another vector?’, ’What does it mean for two vectors to be orthogonal, and how can you verify this property?’, ’Describe the process of normalizing a vector.’, ’Explain how subtracting the projection of one vector from another results in orthogonality.’, ’Given a set of vectors, how can you determine if they are linearly independent?’]}
Generated complementary Depth-2 questions
{“complementary_Depth-2_questions”: [’How can the concept of linear independence be used to form a basis for a vector space?’]}

Example 2
Depth-3 question and current Depth-2 questions
Why couldn’t we test general relativity effects using the Eotvos experiment?
{“current_Depth-2_questions”: [“How does the Eötvös experiment determine the equivalence between inertial mass and gravitational mass?”, “Describe the Equivalence Principle and its significance in the theory of General Relativity.”, “Identify experiments or observations that could directly test the predictions of General Relativity, such as time dilation or the bending of light.”]}
Generated complementary Depth-2 questions
{“complementary_Depth-2_questions”: [“How do experiments measuring time dilation differ in design and scope from those measuring mass equivalence?”]}

Example 3
Depth-3 question and current Depth-2 questions
Why are aldehydes more readily oxidized to carboxylic acids compared to ketones, and how does this difference in reactivity influence their identification in the laboratory?
{“current_Depth-2_questions”: [“How can you identify an aldehyde using Tollens’ reagent?”, “Why does the carbonyl carbon in aldehydes have a significant partial positive charge?”]}
Generated complementary Depth-2 questions
{“complementary_Depth-2_questions”: [“How does the structure of ketones differ from that of aldehydes, and how does this affect their reactivity towards oxidation?”]}

Example 4
Depth-3 question and current Depth-2 questions
In the context of computer programming, why is branching unstructured? And is it a bad design choice?
{“current_Depth-2_questions”: [“What are the key differences between structured and unstructured branching in programming?”, “How does the ’goto’ statement work in computer programming?”]}
Generated complementary Depth-2 questions
{“complementary_Depth-2_questions”: [“What are the potential risks involved with using unstructured branching in large software projects?”, “How does the structure of a program affect its maintainability?”, “How can the flow of execution in a program influence its debuggability?”]}

Depth-3 question
{question}

Answer to the Depth-3 question
{answer}

Current Depth-2 questions
{“current_Depth-2_questions”: {current_questions}}

Generated {count} complementary Depth-2 questions

Table 25: User prompt for augmenting D_2 questions.

User prompt:

```
## Instruction
Create {count} Depth-1 question(s) that complement current Depth-1 questions, which are necessary to correctly answer the provided Depth-2 question.
- Remember that Depth-1 questions are centered on basic recall of factual and conceptual knowledge. Depth-2 questions are centered on application of procedural knowledge and skills.
- Take into consideration the level of cognitive complexity required to solve the Depth-2 question, so that your generated questions fall under the description of Depth-1 appropriately.
- Complement the existing Depth-1 questions with additional questions to ensure they collectively cover all necessary procedural knowledge and skills required to answer the Depth-2 question effectively.
- Ensure that all of your generated Depth-1 questions do not directly answer to the given Depth-2 question.
- Try to exclude Depth-1 questions that ask too generic or commonsense knowledge.
- The number of all Depth-1 questions should not exceed 4.
- The generated Depth-1 questions should be in JSON format: {“complementary_Depth-1_questions”: [list of Depth-1 question strings]}

## Example 1
### Depth-2 question and current Depth-1 questions
How can the concept of algebraic closure be demonstrated using polynomial equations with complex roots?
{“current_Depth-1_questions”: [’What is the definition of algebraic closure?’, ’What is a polynomial equation?’, ’What are complex roots in the context of polynomial equations?’]}
### Generated complementary Depth-1 questions
{“complementary_Depth-1_questions”: [’How can complex roots be represented?’]}

## Example 2
### Depth-2 question and current Depth-1 questions
How do you perform a convolution operation between two random variables?
{“current_Depth-1_questions”: [’What is a convolution operation?’, ’What is a random variable?’, ’How is the product of two functions calculated?’]}
### Generated complementary Depth-1 questions
{“complementary_Depth-1_questions”: [’What does it mean to integrate a function?’]}

## Example 3
### Depth-2 question and current Depth-1 questions
In what ways can a decision tree’s structure be represented programmatically?
{“current_Depth-1_questions”: [’What is a decision tree in the context of programming?’, ’What are the basic components of a decision tree?’]}
### Generated complementary Depth-1 questions
{“complementary_Depth-1_questions”: [’What is a data structure in programming?’, “What does ’represented programmatically’ mean?”]}

## Example 4
### Depth-2 question and current Depth-1 questions
How do neutrinos differ from other subatomic particles, and why are they considered potential candidates for dark matter?
{“current_Depth-1_questions”: [’What are neutrinos?’, ’What are subatomic particles?’]}
### Generated complementary Depth-1 questions
{“complementary_Depth-1_questions”: [’What is dark matter?’, ’What characteristics do particles need to be considered candidates for dark matter?’]}

## Depth-2 question
{question}

## Answer to the Depth-2 question
{answer}


## Current Depth-1 questions
{“current_Depth-1_questions”: {current_questions}}

## Generated {count} complementary Depth-1 questions
```

Table 26: User prompt for augmenting D_1 questions.

System prompt:
You are a helpful, respectful and honest assistant. Answer the question.

User prompt:
Question:
{D_k question}

Answer:

Table 27: Prompt for zero-shot inference.

System prompt:
You are a helpful, respectful and honest assistant. Answer the question.

User prompt:
QA pairs:
Q: {D_{k-1} question 1}
A: {D_{k-1} answer 1}
Q: {D_{k-1} question 2}
A: {D_{k-1} answer 2}
...
Question:
{D_k question}

Answer:

Table 28: Prompt for inference given reference answers or self-predictions on shallower questions.

System prompt:
You are a helpful, respectful and honest assistant. Answer the question.

User prompt:
Question:
{D_{k-1} question}

Answer:

[Last turn] System prompt:
You are a helpful, respectful and honest assistant. Answer the question.

User prompt:
Based on previous questions, answer the question. ## Question:
{D_k question}

Answer:

Table 29: Prompt for multi-turn inference.

System prompt:
You are a fair judge assistant tasked with providing clear, objective feedback based on specific criteria, ensuring each assessment reflects the absolute standards set for performance.

User prompt:
###Task Description:
An instruction (might include an Input inside it), a response to evaluate, and a score rubric representing a evaluation criteria are given.
1. Write a detailed feedback that assess the quality of the response strictly based on the given score rubric, not evaluating in general.
2. After writing a feedback, write a score that is an integer between 1 and 5. You should refer to the score rubric.
3. The output format should look as follows: "Feedback: (write a feedback for criteria) [RESULT](an integer number between 1 and 5)"
4. Please do not generate any other opening, closing, and explanations.

###The instruction to evaluate:
`{instruction}`

###Response to evaluate:
`{response}`

###Reference Answer (Score 5):
`{reference_answer}`

###Score Rubrics:
[Is the response correct, accurate, and factual?]
Score 1: The response is largely incorrect, inaccurate, and not factual. It demonstrates a fundamental misunderstanding of the query or topic, leading to irrelevant or completely erroneous information.
Score 2: The response is partially correct but contains significant inaccuracies or factual errors. It shows some understanding of the query or topic but fails to provide a fully accurate or reliable answer.
Score 3: The response is generally correct and factual but may include minor inaccuracies or lack of detail. It shows a good understanding of the query or topic but may miss some nuances or specific information.
Score 4: The response is mostly correct, accurate, and factual. It demonstrates a strong understanding of the query or topic, with only minimal inaccuracies or omissions that do not significantly detract from the overall quality of the response.
Score 5: The response is consistently correct, accurate, and entirely factual. It reflects a comprehensive understanding of the query or topic, providing detailed, precise, and fully reliable information without any inaccuracies or omissions.

###Feedback:

Table 30: Prompt for LLM-as-a-Judge evaluation with an accuracy score rubric.