# Investigating How Large Language Models Leverage Internal Knowledge to Perform Complex Reasoning


**Anonymous ACL submission**



## Abstract

Despite significant advancements, there is a limited understanding of how large language models (LLMs) utilize knowledge for reasoning. To address this, we propose a method that deconstructs complex real-world questions into a graph, representing each question as a node with parent nodes of background knowledge needed to solve the question. We develop the DEPTHQA set, 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*, discrepancies in LLMs' 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 have more discrepancies than larger models. 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 understand the causal relationship between gradients and training speed, and compare the characteristics of activation functions. This requires drawing conclusions beyond simply aggregating facts about individual activation functions.

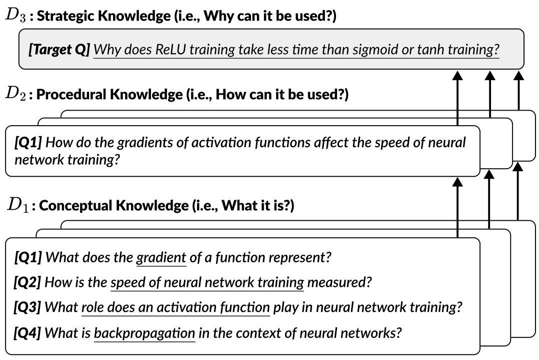



Figure 1: Example of reasoning across depths, showing the sequence of questions from $D_1$ (conceptual knowledge) to $D_3$ (strategic knowledge).

To investigate the reasoning ability of LLMs in solving real-world questions, we propose the 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 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 and reasoning gained from the shallower nodes. This approach emphasizes the grad-

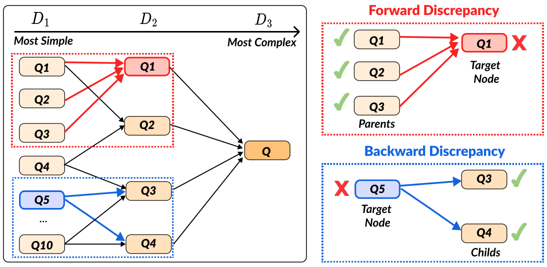



Figure 2: Hierarchical structure of a deconstructed $D_3$, illustrating forward and backward discrepancies.

ual 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). Our focus is on $D_3$ as the target complex questions, examining 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 the *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.
- We investigate the reasoning abilities of LLMs with various capacities, analyzing the impact of model size and training data memorization on discrepancies.

## 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). However, despite these rapid developments, 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 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 tackle these challenges, several studies have focused on prompting 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 method has significantly improved performance, especially in larger models with robust generation capabilities. Theoretical analyses further support the benefits of erbalizations, validating their role in improving the reasoning capabilities of language models (Feng et al., 2023; Wang et al., 2023a; Li et al., 2024). In our data, the complexity of $D_3$ questions often necessitates intermediate steps to derive a conclusion, similar to explicit verbalized reasoning. However, unlike previous works, our setup does not mandate detailed stepwise answers, posing a direct query to the models. We compare the discrepancy between questions with different complexities, enabling a more realistic 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., 2023), or evaluating through binary accuracy of the model's short-form predictions without considering intermediate steps (Yang et al., 2024; Wang et al., 2024). Our DEPTHQA, in contrast, challenges the model to answer complex questions that require diverse reasoning types in long-form text. DEPTHQA further requires diverse types of reasoning across different depths, such as relational and causal 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 by investigating the gap in forward and backward reasoning directions.

## 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 $D_k (k \in \{1, 2, 3\})$[1] 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 correlate simpler questions with shallower depths and more complex questions with deeper depths.

### 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 representing transitions from each child node at $D_k$ to at least one parent node at $D_{k+1}$[2]. We perceive that advancing to deeper knowledge often requires synthesizing multiple aspects of simpler knowledge; thus, a parent $D_{k+1}$ node should connect to multiple child $D_k$ 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. Additionally, we establish three criteria for the graph that must be met to ensure 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 con-

[1] 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).

[2] We acknowledge that a foundational concept may apply to multiple advanced questions.

nections 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 as a testbed for graph-based reasoning. The dataset is constructed in a top-down approach by deconstructing $D_2$ nodes from $D_3$ nodes, followed by $D_1$ nodes from $D_2$ nodes, creating numerous edges at each step (Table 1). We design the construction process to meticulously backtrack the knowledge necessary to solve complex questions while meeting the three criteria to ensure the representation of reasoning transition.

**$D_3$ 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[3] available on `libretexts.org`. Note that while these textbooks are likely included in models' pre-training data due to its online availability, human-written questions in TutorEval challenges generalization of familiar concepts, which is not directly presented during training. We procure only complex $D_3$ questions from TutorEval, sorting them out using GPT-4 Turbo[4] (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. As the ground-truth solution to each TutorEval question is provided as key points, we then use GPT-4 Turbo to generate reference answers[5] based on the original context of each question and the required depth of knowledge annotated by the model itself in the previous step.

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

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

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

| Domain | # Questions | | | # Edges between questions | |
|---|---|---|---|---|---|
| | $D_1$ | $D_2$ | $D_3$ | $D_1 \leftrightarrow D_2$ | $D_2 \leftrightarrow D_3$ |
| 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 $\leftrightarrow$ {CS, Physics} | - | - | - | 11 | 0 |
| Total | 1,121 | 359 | 91 | 1,437 | 363 |

Table 1: Statistics of DEPTHQA.

**Question deconstruction** For each $D_k$ question, we generate up to four $D_{k-1}$ questions using GPT-4 Turbo. Within the prompt, we include definitions for all three depths of knowledge and decomposition examples to signify the purpose of deconstruction. We provide $D_k$ paired with its reference answer as well to facilitate the extraction of essential knowledge needed to tackle more challenging questions, thereby maintaining the relevance of questions and adhering to **C1 (Comprehensiveness)**. We decide the optimal number of decompositions to four through qualitative analysis as there is a tradeoff between comprehensiveness and implicitness: outlining every implicit reasoning step enhances comprehensiveness but can reduce implicitness, and the reverse is also true. Therefore, we clearly instruct these properties in the prompt to treat **C2 (Implicitness)**.

**Deduplication and question augmentation** We have identified instances where similar knowledge is posed by different $D_1$ nodes linked to the same $D_2$ node or even between $D_1$ and $D_2$ nodes without direct dependencies, leading to redundancy in knowledge and reasoning processes. To address this, we utilize a Sentence Transformers embedding model[6] (Reimers and Gurevych, 2019) to identify and eliminate near-duplicate questions through cosine similarity of their embeddings. We then engage GPT-4 Turbo to generate new, targeted questions and answers that effectively fill any gaps in knowledge coverage. This approach has led to a 88% reduction in misclassification of $D_1$ questions as $D_2$, markedly enhancing **C2 (Implicitness)**. Additionally, the total number of near-duplicates has decreased by 88%, further improving **C1 (Comprehensiveness)**. We continually update our graph data structure to incorporate these modifications, enabling more interconnectedness as some $D_2$ nodes now link to multiple $D_3$ nodes or more than four $D_1$ nodes.

[6]`sentence-transformers/all-mpnet-base-v2`

| 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.

**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.

We provide details and examples in the data construction process in Appendix A and prompts in Appendix G.1.

### 3.4 Diversity of Reasoning Processes

We examine the types of reasoning needed to progress from basic to complex knowledge levels using a sample of 20 $D_3$ questions along with their corresponding 80 $D_2$ and 320 $D_1$ questions. 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 C.

## 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 reasoning as multi-turn inputs or prompt inputs (Section 4.4).

### 4.1 Experiment Setup

**Depthwise evaluation** For each question $q_k$ with depth $k$, we score the factual correctness of the predicted answer on a scale from 1 to 5. We 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 D. 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 $q_k$ with depth $k \in \{2, 3\}$, $Parent(q_k)$ represents a set of parent questions at depth $k-1$. Forward discrepancy for $q_k$ is defined as follows:

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

where $f$ is function for factual correctness. Conversely, **backward discrepancy** quantifies incon-

| Model | Average Accuracy ↑ | | | | Forward Discrepancy ↓ | | | Backward Discrepancy ↓ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $D_1$ | $D_2$ | $D_3$ | Overall | $D_2 \leftrightarrow D_3$ | $D_1 \leftrightarrow D_2$ | Overall | $D_2 \leftrightarrow D_3$ | $D_1 \leftrightarrow D_2$ | 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. Darker color indicates higher discrepancy.

sistencies where LLMs can successfully answer deeper questions but struggle with shallower ones. Given a question $q_k$ with depth $k \in \{1, 2\}$, backward discrepancy is defined as follows:

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

where $Child(q_k)$ represents a set of child questions at depth $k+1$. Both forward discrepancy and backward discrepancy are normalized to the range [0, 1] by dividing by the maximum possible gap of 4. To focus on the reasoning gap within depths, we report the average values of each discrepancy for examples where the average score for $Parent(q^k)$ and $Child(q^k)$ is larger than 4. This approach ignores cases where models failed to provide appropriate answers.

**Models** We investigate 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 (OpenAI, 2022) to compare the performance of these open-source models against a proprietary model.

### 4.2 Depthwise Knowledge Reasoning Results

**Overall results** Table 3 presents the overall depthwise reasoning performance of LLMs. As anticipated, solving questions in $D_3$ is the most challenging, while $D_1$ yields higher performance across all models. Among the models, LLaMA 3 70B Instruct demonstrates the best performance across all depths, with Mixtral 8x7B Instruct achieving the second-best results. LLaMA 3 70B Instruct also exhibits the lowest discrepancies for both forward and backward discrepancy metrics, effectively solving questions at all depths with minimal discrepancies. Conversely, the least capable model, LLaMA 2 7B Chat, shows the lowest performance along with the highest forward and backward discrepancies. Note that the relatively low forward discrepancy from $D_1 \leftrightarrow D_2$ for LLaMA 2 7B Chat is due to its low performance at $D_2$. This observation hightlight the varying capabilities of different LLMs in handling depthwise reasoning.

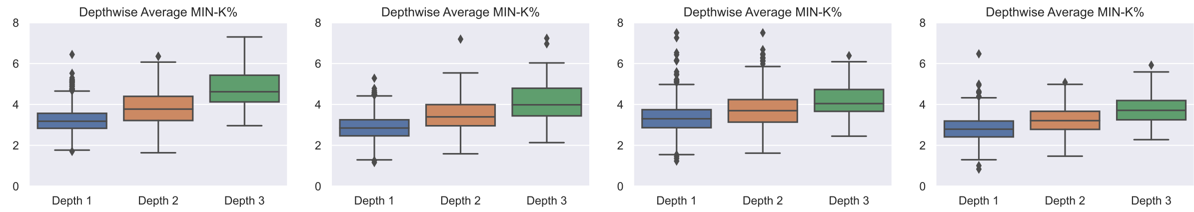
**Discrepancy patterns** We observe distinct patterns by analyzing the trends in forward and backward discrepancies separately. When considering discrepancies as the product of intensity (*i.e.*, the value of discrepancies) and frequency (*i.e.*, the ratio of questions with positive discrepancy), forward discrepancy tends to occur more frequently but with lower intensity. For instance, LLaMA 3 8B Instruct exhibits an intensity of 0.225 with a frequency of 41.44%. In contrast, backward discrepancy happens less frequently but is more significant when it occurs. LLaMA 3 8B Instruct shows an intensity of 0.323 with a frequency of 23.32% for backward discrepancy.[7] This results show the distinct nature of forward and backward discrepancies in model performance.

### 4.3 Memorization in Depthwise Knowledge Reasoning

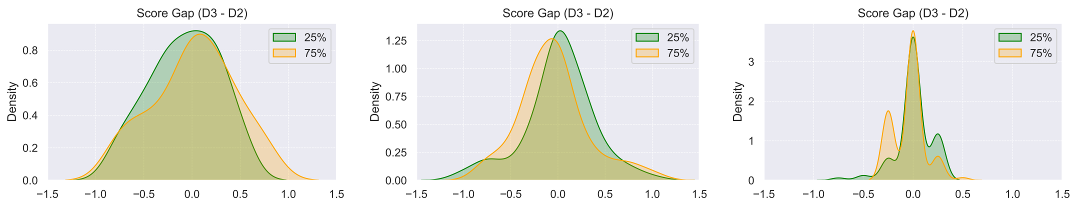
#### 4.3.1 Depthwise Memorization

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

[7]The intensity and frequency for all models are provided in Appendix E.




(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 average Min-K% probability at each depth. (e)-(g) present the score difference 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.

memorization. Following Shi et al. (2023), we compare the Min-K% probability within models. Higher values suggest a lesser possibility of predictions directly existing in the training data, compared to those with lower values. More specifically, Min-K% probability averages the negative log-likelihood of the K% tokens with the lowest probabilities in the model's predictions. If a given prediction was included in the training, outlier words with low probabilities would be less frequently appeared, resulting in high probabilities for the Min-K% tokens. Since Min-K% probability is the average negative log-likelihood, the resulting value would be lower if the text was present during training.[8]

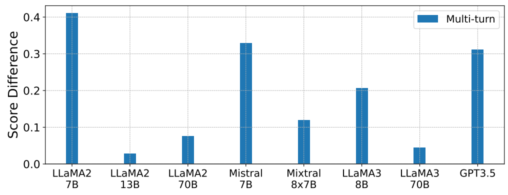
**Models rely less on memorization for complex questions.** Figure 3 (a)-(d) present the depthwise average of the Min-K% probability for four models. We observe that as the depth increases, the Min-K% probability also increases for all models. This indicates that answering questions based on simple conceptual 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.

[8]For our calculations, we set $k$ to 20 and used a sequence length of 128.

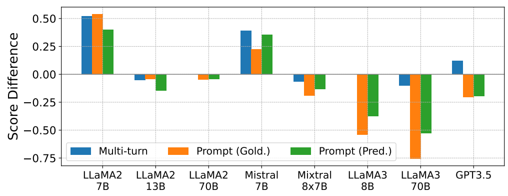
#### 4.3.2 Memorization Gap between Depths

Further analysis of questions in the bottom 25% and top 75% quantiles of the Min-K% probability distribution provides additional insights. Figure 3 (e)-(g) show the score difference between neighboring questions ($D_2 \leftrightarrow D_3$), whose Min-K% probability is in the bottom 25% or top 75%. The gap is calculated 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.

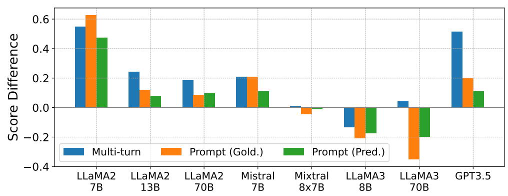
**Failure modes for discrepancies** We observe that the model with small capacity, LLaMA 2 7B Chat, exhibits large variances in both directions, showing significant forward and backward discrepancies. In contrast, models with larger capacities, such as LLaMA 2 70B Chat and LLaMA 3 70B Instruct, demonstrate smaller variances. Additionally, models with larger capacities tend to show relatively higher forward discrepancies for the top 75% examples, which rely less on memorization. On the other hand, the bottom 25% show a slight shift towards positive values, indicating relatively more backward discrepancies. This suggests that failures in knowledge reasoning result in forward discrepancies, while failures due to reliance on memorization may lead to backward discrepancies. The depth-




(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 $D_1$ do not have shallower questions.

wise Min-K% probability and score difference for other models are provided in Appendix F.

### 4.4 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 reasoning (Wei et al., 2022b; Press et al., 2023; Zhou et al., 2023) have shown that enforcing LLMs to explicitly reason through intermediate steps 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 inputs 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 input example for each approach is provided in Appendix G.

**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 simple 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).

**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, while more capable models favor self-prediction results. This preference may stem from the alignment of self-generated inputs with the models' internal reasoning when they are already proficient. 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, constructed to evaluate the knowledge reasoning abilities of LLMs, 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 intricate questions that necessitate advanced reasoning skills and the scarcity of existing datasets that meet criteria for (1) real-world relevance, (2) long-form question answering to assess deep knowledge reasoning, and (3) 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 establish more robust benchmarks that provide a larger and more varied set of questions to better assess the knowledge reasoning capabilities of LLMs.

**GPT-4 data generation and evaluation** All questions except for $D_3$ and reference answers in DEPTHQA is generated by GPT-4 Turbo, which may introduce inaccuracies due to potential errors in the decomposition process or unverified knowledge produced by the model. 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, and manual rewriting and verification (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. Furthermore, we utilize GPT-4 Turbo to assess the correctness of model predictions. Following established 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 research objectives.

## 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.

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 *Neur*, 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.

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*.

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. 2023. 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*.

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*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

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 *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

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*.

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 14. 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[9] (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 $\geq 0.9$ 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 20 and 21) to augment questions filling the gap of knowledge. Table 5 illustrates how an augmented question successfully replaces a previously redundant question, enhancing comprehensiveness.

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

| |
|---|
| *Top-1 before deduplication (similarity = 0.97)*<br>$D_2$: How do you calculate the determinant of a matrix?<br>$D_1$: How do you find the determinant of a matrix? |
| *Top-1 after deduplication (similarity = 0.93)*<br>$D_2$: What does it mean for two vectors to be orthogonal, and how can you verify this property?<br>$D_1$: 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?<br>2. How is a remainder defined in division?<br>3. What does it mean for two numbers to be congruent modulo a number?<br>4. What does the term 'congruence modulo a number' mean? $\Rightarrow$ 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-

| *Original question:* | *Flipped 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 | 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 binary questions into non-binary 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 8: Distribution of reasoning types for $D_3$ and $D_2$ in a subset of DEPTHQA. Multiple reasoning types can be included in one instance.

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 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.

## C Reasoning Type Analysis

In Table 8, 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. We provide question deconstructions examples in Table 11 and Table 12 where each showcases distinct reasoning types and knowledge.

## D Details in Experiments

### D.1 Model Inference

To inference LLMs used in our experimental setup (Section 4.1), we use a standardized API from OpenRouter[10] to access LLMs and use the complementary LiteLLM[11] 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 on 1 NVIDIA A6000 GPU. We use the default sampling parameters suited for each model. The specific prompt templates used to induce reasoning paths are organized in Appendix G.2. The inference on the whole pass of DEPTHQA finishes within 10 minutes. We report single-run results.

[10] openrouter.ai
[11] litellm.vercel.app/docs/providers/openrouter

### D.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 25. We report single-run results. 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.

## E Discrepancy Results

To separately observe how frequently each discrepancy occurs and its intensity when it happens, Table 9 and Table 10 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).

## F Overall Results with Min-K% Probability

### F.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.

### F.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 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.

## G Prompts

### G.1 Data construction

We provide the prompts used to classify TutorEval questions (Table 14), generate $D_3$ answers (Table 15), generate $D_2$ or $D_2$ answers (Table 16), generate questions at $D_2$ (Table 18) and $D_1$ (Table 19), and augment questions at $D_2$ (Table 20) and $D_1$ (Table 21). For generating or augmenting any question at $D_2$ or $D_1$, we use the same system prompt (Table 17) that describes the definitions of depths of knowledge.

### G.2 Inference

We provide the prompts used for zero-shot (Table 22), Prompt (Gold) and Prompt (Pred.) (Table 23), and multi-turn (Table 24) inference.

### G.3 Evaluation

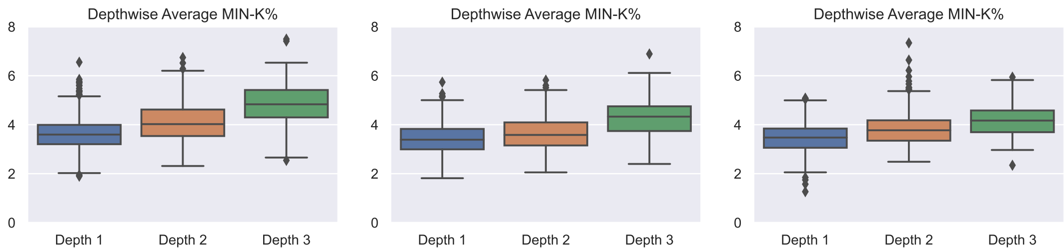
The prompt used for LLM-as-a-Judge evaluation is in Table 25.

| Model | Average Forward Discrepancy | | | Value | | | Frequency (%) | | |
|---|---|---|---|---|---|---|---|---|---|
| | $D_2 \leftrightarrow D_3$ | $D_1 \leftrightarrow D_2$ | Overall | $D_2 \leftrightarrow D_3$ | $D_1 \leftrightarrow D_2$ | Overall | $D_2 \leftrightarrow D_3$ | $D_1 \leftrightarrow D_2$ | 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 9: Average intensity and frequency of forward discrepancy.

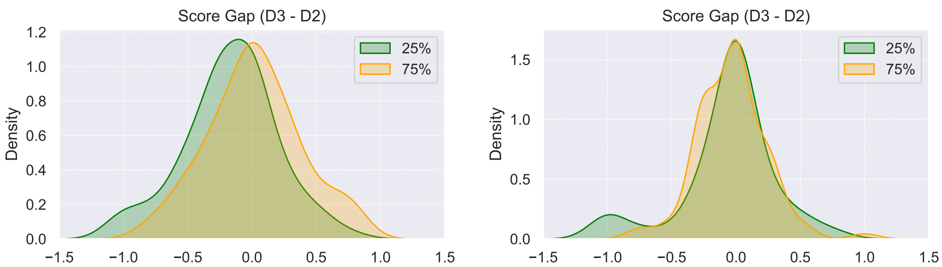
| Model | Average Backward Discrepancy | | | Value | | | Frequency (%) | | |
|---|---|---|---|---|---|---|---|---|---|
| | $D_2 \leftrightarrow D_3$ | $D_1 \leftrightarrow D_2$ | Overall | $D_2 \leftrightarrow D_3$ | $D_1 \leftrightarrow D_2$ | Overall | $D_2 \leftrightarrow D_3$ | $D_1 \leftrightarrow D_2$ | 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 10: Average intensity and frequency of backward discrepancy.

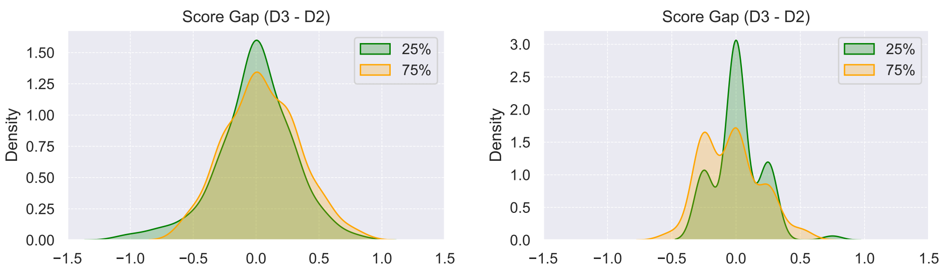



(a) LLaMA 2 13B Chat (b) Mistral 8B Instruct (c) Mixtral 8x7B Instruct

Figure 5: Average Min-K% probabilaity 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.

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

Table 11: 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.

| | | |
|---|---|---|
| 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? | | |
| | How do horses and donkeys differ genetically and what makes them separate species? | |
| | What is the definition of a viable offspring in biological terms? | |
| | Can you explain the concept of hybrid vigor and how it applies to cross-species offspring such as mules? | |
| | | What is the definition of hybrid vigor? |
| | | What is a mule? |
| | | How is a mule produced? |
| | | What are the parent species of a mule? |
| | What are the implications of sterility in hybrids like mules for their classification as members of a species? | |

Table 12: 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:**<br>I'm studying ideal and real gasses in engineering thermodynamics. Why can't we measure temperature in Fahrenheit? |
| **Predicted answer:**<br>In engineering thermodynamics, it's common to work with ideal and real gases, and temperature plays a crucial role in characterizing their behavior.<br><br>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.<br><br>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.<br><br>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.<br><br>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.<br><br>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:**<br>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:**<br>4 |

Table 13: Example LLaMA 8B Instruct zero-shot inference on a $D_3$ question followed by GPT-4 Turbo evaluation.

**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 14: 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 15: Prompt for generating reference answer for a $D_3$ question.

| |
|---|
| **System prompt:**<br>You are a helpful assistant that accurately answers complex questions. Ensure that your answer is focused and compact. |
| **User prompt:**<br>`{question}` |

Table 16: Prompt for generating reference answer for a $D_1$ or $D_2$ question.

| |
|---|
| **System prompt:**<br>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:<br><br>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.<br>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.<br>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 17: 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 18: 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 19: 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 20: 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 21: 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 22: 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 23: 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 24: 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 25: Prompt for LLM-as-a-Judge evaluation with an accuracy score rubric.