Working Memory Identifies Reasoning Limits in Language Models

Chunhui Zhang **Yiren Jian** **Zhongyu Ouyang** **Soroush Vosoughi***
Department of Computer Science, Dartmouth College
{chunhui.zhang.gr, yiren.jian.gr, zhongyu.ouyang.gr, soroush.vosoughi}@dartmouth.edu

Abstract

This study explores the inherent limitations of Large Language Models (LLMs) from a scaling perspective, focusing on the upper bounds of their cognitive capabilities. We integrate insights from cognitive science to quantitatively examine how LLMs perform on *n*-back tasks—a benchmark used to assess working memory, which involves temporarily holding and manipulating information. Our findings reveal that despite increased model size, LLMs still face significant challenges in holding and processing information effectively, especially under complex task conditions. We also assess various prompting strategies, revealing their diverse impacts on LLM performance. The results highlight the struggle of current LLMs to autonomously discover optimal problem-solving patterns without heavily relying on *manually* corrected prompts. To move beyond these constraints, fundamental improvements in the planning and search of LLMs are essential for them to reason autonomously. Improving these capabilities will reduce the reliance on external corrections and enable LLMs to become more autonomous in their problem-solving processes.

1 Introduction

Working memory is integral to human reasoning, serving as the cognitive system that temporarily holds and manipulates information for complex tasks like learning, reasoning, and comprehension (Atkinson and Shiffrin, 1968; Buehner et al., 2005). Understanding working memory is pivotal not only for grasping human cognitive processes but also for exploring the capabilities and limitations of LLMs. Cognitive limits represent the thresholds where the efficiency of information processing and retention starts to wane, critical for pushing the boundaries of what LLMs like InstructGPT and ChatGPT can achieve (Baddeley, 1992; OpenAI, 2023; Ouyang et al., 2022a).

Models	Param. (B)	Token (B)	ZettaFLOP
`InstructGPT`			
text-ada-001	0.35	$\sim$ 300	$\sim$ 0.64
text-babbage-001	1.3	$\sim$ 300	$\sim$ 2.3
text-curie-001	6.7	$\sim$ 300	$\sim$ 12
text-davinci-001	175	$\sim$ 300	$\sim$ 315
text-davinci-002	175	$\sim$ 300	$\sim$ 315
`ChatGPT`			
gpt-3.5-turbo-0613	N.A.	N.A.	N.A.
gpt-4-0613	N.A.	N.A.	N.A.

Table 1: Details of the scaling parameters for LMs from Wei et al. (2023) and Ouyang et al. (2022a). Our study also covers *new **open-source** reasoning LMs* in App.G

As models like InstructGPT and ChatGPT evolve (see Tab.1), their enhanced capabilities necessitate a comparison to human cognitive processes, especially regarding working memory and cognitive limits (OpenAI, 2023; Ouyang et al., 2022a). LLMs have excelled in various natural language tasks, often surpassing domain-specific models in areas such as machine translation and visual QA (Brown et al., 2020; Alayrac et al., 2022; Chowdhery et al., 2023). This success is attributed to their advanced reasoning abilities, honed through innovative prompting methodologies such as the chain of thought (CoT), self-consistency, and automated reasoning (Brown et al., 2020; Wei et al., 2022a,c; Wang et al., 2023c; Zhang et al., 2023).

In this study, we explore the boundaries of LLMs' working memory, focusing on their performance in handling information under various prompting conditions. Our analysis reveals that despite their advanced capabilities, LLMs struggle with specific complex tasks from the Big-bench Hard (BBH) dataset, highlighting a stark contrast to human problem-solving abilities (Suzgun et al., 2023). This discrepancy challenges the prevailing notion of LLMs' autonomy and self-corrective reasoning (Wang et al., 2023c; Zhang et al., 2023; Huang et al., 2024).

*Corresponding author

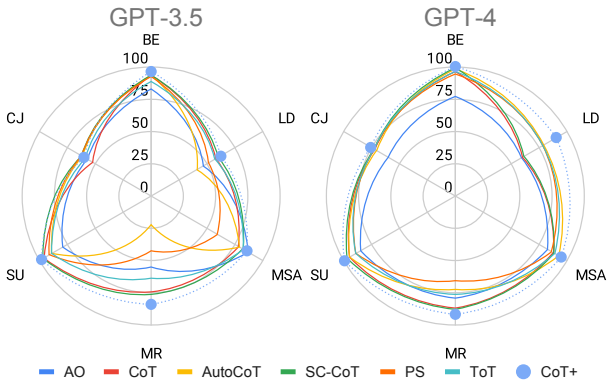

Figure 1: Categorization of BBH tasks showcasing different cognitive emphases: *logic* (**B**oolean **E**xpressions, **L**ogical **D**eduction, **M**ulti-**S**tep **A**rithmetic) and *world knowledge* (**M**ovie **R**ecommend, **S**ports **U**nderstand, **C**ausal **J**udgement). CoT+ denotes enhancements to CoT correcting prior evaluation errors.

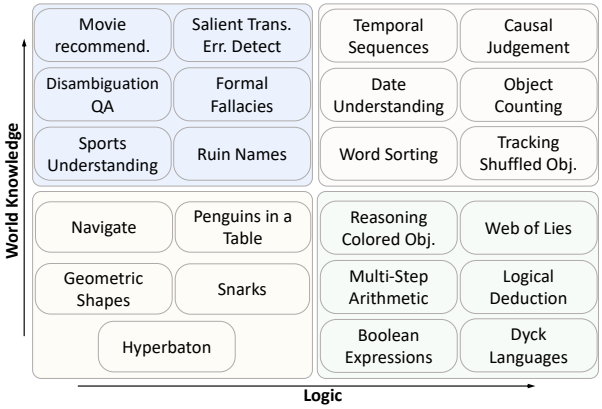

Figure 2: Classification of reasoning tasks into two dimensions: world knowledge and logic. Refer to App.C for the rationale behind this categorization.

Our comparative analysis reveals significant reasoning limitations in scaled models, especially when compared to the general intelligence of humans. These limitations manifest primarily in an autonomy deficit within prompts, rendering them ineffective for enabling LLMs to autonomously uncover problem-solving patterns, and a lack of prompt effectiveness in enhancing comprehension of **world knowledge** and **logic**, key components for reasoning through answers effectively.

In response to these challenges, our study proposes potential remedial strategies. As illustrated in Fig.1, despite a variety of prompt designs, their effectiveness is significantly inferior compared to our refined **CoT+** prompts, which are derived from correcting mistakes. These CoT+ prompts incorporate specific problem-solving patterns that our analysis identified as crucial for addressing and rectifying common errors, thus demonstrating their superior efficacy. Moreover, the process of scaling LLMs, such as the transition from GPT-3.5 to GPT-4, significantly enhances their reasoning abilities in both **world knowledge** and **logic**, independent of the complexity of the prompts used.

By anchoring our analysis in cognitive science, we establish parallels between *(i)* **human working memory**—the system for temporarily retaining and manipulating information during cognitive tasks—and *(ii)* **the intrinsic capabilities of LLMs** (Baddeley, 1992). Utilizing the *n*-back dataset (Gong et al., 2024), a well-recognized cognitive science benchmark, we explore the performance of various scaled LLMs. Our results reveal a significant discrepancy between the working memory capacities of LLMs and humans, emphasizing the ongoing challenges LLMs face in complex problem-solving tasks, regardless of their scale or the prompting methods utilized.

To address these challenges, *(i)* we must focus on **optimizing working memory utilization** to enhance LLM performance. CoT+ prompts, refined from error analysis, not only improve question comprehension but also enhance the reasoning process by effectively utilizing the limited working memory capacity. This approach incorporates cognitive-science strategies such as chunking, logical step arousal, and rehearsal techniques. *(ii)* Beyond these practical solutions, our analysis strongly advocates for expanding the working memory capacity of LLMs as a crucial step toward their development into truly autonomous problem-solvers. Such advancements are expected to significantly improve their capacity to independently **plan and search** through complex logical constructs and world knowledge, thereby deepening their reasoning autonomy. Our code is available: `https://github.com/chunhuizng/working-memory-limits`

2 Fundamentals of Working Memory

Working memory is fundamental to reasoning, acting as a dynamic reservoir that temporarily holds and manipulates information essential for complex cognitive tasks (Kyllonen and Christal, 1990; Buehner et al., 2005). This capability is crucial for fluid reasoning—a core cognitive skill that involves deducing relationships and identifying patterns in tasks that require minimal prior knowledge.

Contemporary research has focused on enhancing attention mechanisms within deep learning models, striving for a more refined understanding

of cognitive processing (Guo et al., 2020; Li et al., 2023b). However, these efforts often deviate from traditional conceptualizations of working memory. Distinctly, (Gong et al., 2024) introduced the *n*-back dataset as a benchmark for assessing working memory, tailored specifically for evaluating cognitive capacities.

Our study diverges from conventional approaches by examining working memory within LLMs from a perspective that integrates both cognitive science principles and computational scalability, particularly in the context of reasoning. Furthermore, we investigate how scaling up models affects their working memory capabilities and propose strategies to mitigate these burdens. These strategies aim to enhance the complex reasoning abilities of LLMs. *For a detailed discussion on scaled language models and their implications on working memory, see App.A.*

3 Cognitive Limits of LLMs as Reasoners

Tasks To assess the reasoning capabilities of LLMs, we utilize the BBH set containing 23 tasks, which are categorized into two main dimensions of reasoning: world knowledge and logic. This categorization is visually represented in Fig.2.

Models We deploy a range of models from InstructGPT to ChatGPT, as detailed in Tab.1 and App.B. Models range from text-ada-001 with 0.35B parameters to text-davinci-002 with 175B parameters, each trained on approximately 300 billion tokens. Notably, the text-davinci-001 and text-davinci-002 models share similar parameters but differ in code training and supervised instruction tuning (Ouyang et al., 2022b). ChatGPT builds upon these foundations, incorporating enhancements in conversational modeling and alignment with human values.

Prompts As a baseline, we use answer-only (AO) prompts where LLMs directly provide answers without intermediate steps. We also employ advanced prompting techniques such as CoT, which guides LLMs through a step-by-step information processing (Wei et al., 2022c); SC-CoT, which ensures consistent reasoning (Wang et al., 2023c); PS, aiding structured problem-solving (Wang et al., 2023a); ToT, organizing reasoning hierarchically (Yao et al., 2023a); and Auto-CoT, automating the reasoning flow (Zhang et al., 2023). Moreover, we introduce CoT+, an enhancement of the basic CoT that incorporates lessons learned from previous errors.

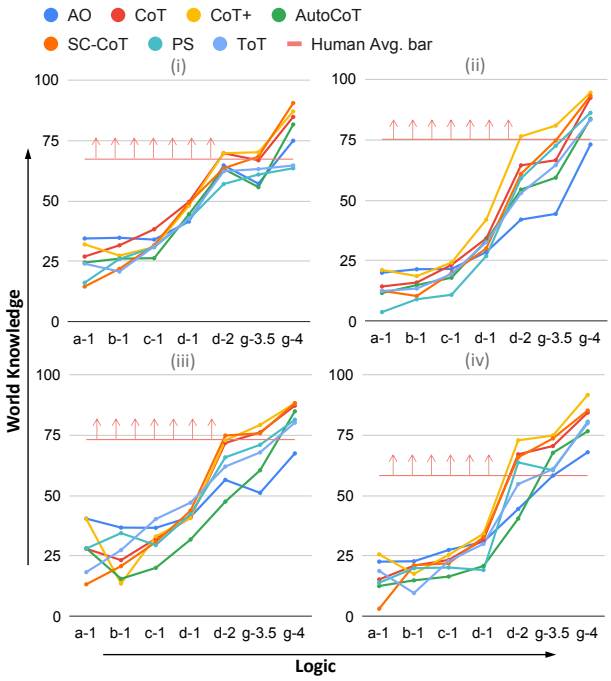

Figure 3: Scaling effects of different prompts across language models: text-ada-001 (a-1), text-ada-001 (b-1) text-curie-001 (c-1) text-davinci-001 (d-1) text-davinci-002 (d-2) GPT-3.5 (g-3.5) GPT-4 (g-4). ↑ on the human Avg. bar indicates the maximum potential human accuracy of 100%. Task performance is divided into four clusters reflecting the dual dimensions of reasoning: world knowledge and logic, as detailed in Fig.2.

Reasoning performance of LMs We assess the impact of various prompts on the performance of scaled LLMs, focusing on scenarios that demand substantial world knowledge and logical reasoning. Figure 3 displays the performance of LLMs across the 23 Big-bench Hard (BBH) tasks, categorized into four groups based on their reliance on world knowledge and logic, as outlined in Figure 2. These categories are: *(i)* pronounced world knowledge but limited logic, *(ii)* balanced world knowledge and logic, *(iii)* deficient in both world knowledge and logic, and *(iv)* robust logic but limited world knowledge.

Further analysis in Appendix D (Figures 11–14) explores LLMs' proficiency in integrating world knowledge and logical reasoning. We observe that despite the use of advanced prompting techniques, LLMs occasionally struggle to fully integrate extensive world knowledge or to derive and apply essential logical patterns effectively, particularly in overcoming typical errors. Additionally, these models face challenges in the temporary retention

and logical manipulation of information, which are critical for successfully applying both knowledge and reasoning in task completion.

Two distinct trends emerge across the task categories. First, the effectiveness of advanced prompts varies with the model size. Smaller models, such as text-curie-002, and those with fewer parameters, often show decreased accuracy with advanced prompts compared to basic answer-only prompts. In contrast, larger models, beginning with text-davinci-001, exhibit significant performance improvements with advanced prompting strategies. This pattern aligns with findings from Wei et al. (2022b); Suzgun et al. (2023), suggesting that emergent capabilities from complex prompting strategies are more pronounced in larger models, while smaller models may not benefit as substantially, potentially due to distractions caused by intermediate reasoning steps.

A second trend underscores the persistent gap in reasoning capabilities between LLMs and humans. Despite advancements, as depicted in Figure 3 and supported by Suzgun et al. (2023), LLMs have not consistently surpassed human intelligence, even with the latest and most sophisticated models like d-2. This observation raises a critical question: **What intrinsic factors limit the cognitive abilities of LLMs as reasoners?** Addressing this question is essential for understanding the reasoning capabilities of LLMs and how they compare to human cognition.

4 Working Memory's Pivotal Role

As LLMs struggle to recognize essential pattern of problem solving, their limitations invite an in-depth discussion from the perspective of *working memory*—**a crucial intrinsic ability that entails temporary storage and manipulation of information for reasoning within cognitive science** (see Fig.4). This capacity is central not only to human cognition but also to the functioning of artificial intelligence systems.

The comparison between LLM and human cognitive processes reveals gaps and offers heuristic strategies to mitigate the current limitations of LLM. This insight underpins the call for further developmental strides aimed fundamentally at enhancing problem-solving capabilities in LLMs. Using our understanding of working memory, we can devise interventions that improve the efficiency and accuracy of LLMs in tasks that require complex cognitive operations.

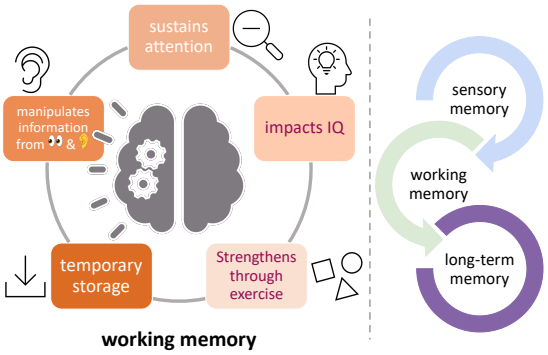

Figure 4: Illustration of working memory's role in the human cognitive system. See details in App.A.3.

4.1 Quantifying Working Memory

n-back benchmark The n-back task is a standard cognitive assessment in psychology used to evaluate working memory (Kirchner, 1958). Participants are challenged to identify when a current stimulus matches one from n steps earlier in a sequence. With variations such as $n = 1, 2,$ and3, the task's complexity increases with higher n values, testing the ability to temporarily store and manipulate information.

This task has been adapted for LMs based on a dataset from Gong et al. (2024), assessing LMs' working memory by evaluating their ability to recognize matches using internal representations. The task uses ASCII characters in grids of different sizes to explore spatial scaling abilities, mirroring challenges found in human cognitive processes.

d': metric of sensitivity d' is a crucial metric for evaluating response accuracy in the n-back task, measuring the ability to distinguish relevant information from distractions (Haatveit et al., 2010). For LLMs, d' quantifies the precision of identifying correct matches, reflecting their working memory and attentional capabilities. Details on the formulation of d' can be found in App.E.2.

4.2 Working Memory in Scaling LMs

U-shaped performance curve of working memory We begin by analyzing the changes in model responses that contribute to a decrease in the d' metric. As illustrated in Fig.6, the smallest model, text-ada-001, initially generates responses that alternate between 'm' and '-,' followed by new QA templates. This pattern indicates that text-ada-001 primarily engages in a basic, repetitive replication of the input QA structure, suggesting a lack of

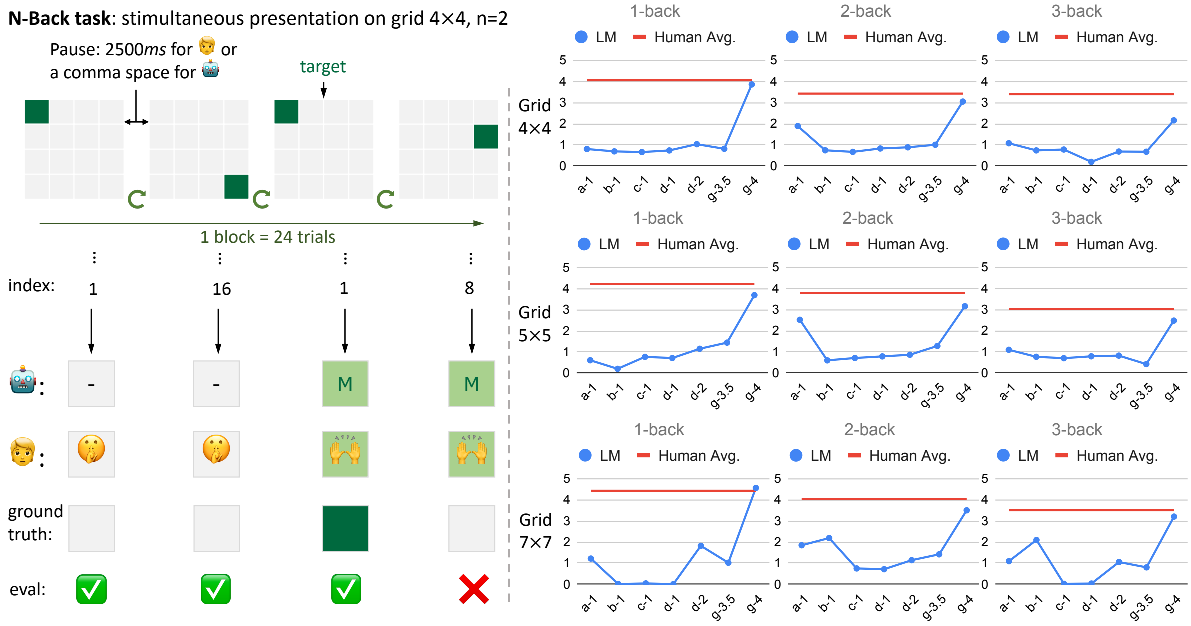

Figure 5: (Left) Visualization of the n-back task at $n = 2$, with further details in App.E. (Right) Performance comparisons in n-back tasks, illustrated using the d' metric for both LMs (blue curve) and human subjects (red line) at $n = 1, 2,$ and 3 across grid sizes of 4×4, 5×5, and 7×7. Model scaling from a-1 to g-4 on the x-axis represents increasing scales. Human benchmarks are used to evaluate LMs' working memory capacities.

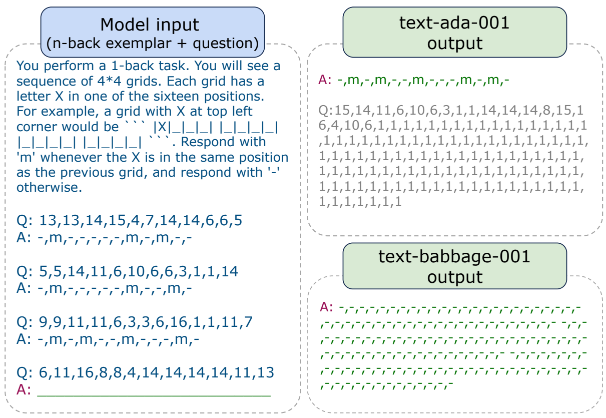

Figure 6: Results in n-back tasks on smaller LMs: a-1 replicates the input format, whereas b-1 diverges from replication but exhibits a collapsed response pattern.

meaningful processing or understanding of the actual intent of the input. This behavior points to the model's extremely limited capacity, as it focuses primarily on matching the format of the input rather than interpreting its content.

We then shift our focus to intermediate-sized models, text-babbage-001 and text-curie-001. Their outputs deviate from the 'm' and '-' mixture, instead displaying a continuous stream of either response. These models do not follow up with the `Question` and `Answer` templates noted in the smallest model. This indicates that while these intermediate-sized models demonstrate a basic grasp of the input's intent, they still fail to fully capture the complex patterns required in n-back tasks, resulting in a reduction of d'.

In contrast, larger models such as text-davinci-001 and beyond show a significant improvement in working memory capabilities. These models not only understand but also accurately respond to n-back tasks, indicating a deeper comprehension and processing ability.

This observed trend is supported by findings from other studies, such as those by Wei et al. (2023) and McKenzie et al. (2023), which suggest that larger models are capable of overcoming simplistic heuristics and the interference from distractor tasks that often impede smaller models. Further discussion on this topic can be found in App.F.

Code training enhances Text-Davinci-002's working memory Figure 3 illustrates significant improvements in working memory capabilities as the model transitions from text-davinci-001 (d-1) to text-davinci-002 (d-2). A key distinction between these versions is the enhancement of d-2 with additional code training, beyond the supervised instructions used in d-1. While d-1 adequately responds to human instructions, d-2 excels in working memory tasks due to its advanced training.

Text-davinci-002 demonstrates a superior ability to *chunk* information, a cognitive science strategy for organizing data into manageable units for more straightforward processing and retrieval (Miller,

1956). This capability mimics human cognitive processes in handling complex tasks, where procedural programming in d-2 aligns with sequential task resolution strategies, and object-oriented programming approaches reflect the decomposition of complex tasks into simpler components. The enhanced code training in text-davinci-002 not only bolsters its internal reasoning but also makes it particularly adept at performing n-back working memory evaluations. This training leads to clearer reasoning patterns and effectively reduces the working memory load in LMs pretrained with code, corroborating recent findings by Li et al. (2023a).

4.3 Comparisons: Humans and Scaled LMs

LMs emerge with human-level working memory via scaling We assess the performance of working memory across both LMs and human participants in varying complexities ($n = 1, 2,$ and 3). Human participants consistently maintain high d' scores, such as in the 4×4 grid setting, where d' ranges between 3.39 and 4.05, indicating robust and consistent working memory capabilities.

In contrast, LMs display varied performance levels. For instance, smaller models like text-ada-001 and text-babbage-001 exhibit d' scores ranging from 0.01 to 2.51, generally falling below the human average, particularly at higher n-back levels. Conversely, as the model size increases to GPT-3.5 and GPT-4, their d' scores improve, suggesting a positive correlation between model size and working memory enhancement. This trend is particularly pronounced in the largest model, GPT-4, where d' scores in 1-back tasks in the 7×7 grid setting can rival or even surpass the human average, highlighting the potential of larger models to match or exceed human working memory capabilities.

These observations indicate that LMs has clear scalability in working memory capacity, with larger models like GPT-4 more closely approximating human-level performance. However, even the most sophisticated GPT-4 models do not consistently outperform humans, especially in tasks of increasing complexity. While scaling up model size does enhance working memory capacity, a discernible gap remains between machine and human cognitive processing, warranting further exploration.

Enhancing working memory: cognitive strategies from human participants Through interviews with human participants, we explored cognitive strategies that alleviate the burden on working memory during n-back tasks.

(i) Rehearsal (Craik and Watkins, 1973; Baddeley et al., 1984): Participants employed their inner voice for sub-vocalization or spoke aloud to continuously repeat sequences, such as a string of letters. For visuospatial tasks, they rehearsed by mentally or visually retracing the locations of the squares. These strategies help maintain and update information in working memory, enhancing the ability to recall and process current stimuli.

(ii) Chunking (Miller, 1956; Simon, 1974): Participants grouped individual elements into larger, more manageable units or "chunks," simplifying the information processing. By reducing the discrete elements to remember, chunking decreases the immediate cognitive load, making the task appear more manageable even though it does not increase the overall capacity of working memory.

(iii) Arousal (*the state of being physiologically alert and mentally focused, enhancing the ability to process and recall critical information*) (Libkuman et al., 2004): Participants concentrated their attention on specific sequences or item groups, focusing on relevant strings or patterns rather than updating their working memory with each item. This selective attention allowed them to better recall the most relevant stimuli for responding to the n-back tasks. Detailed descriptions of the interview setup and methodologies can be found in App. E.3.

5 Applying Cognitive Strategies to Mitigate Reasoning Limits in LLMs

To address reasoning challenges in LLMs due to their limited working memory capacity, we incorporate cognitive science principles aimed at enhancing their ability to effectively utilize working memory. This approach improves LLMs in recalling world knowledge and executing logical manipulations. These strategies and **open-source reasoning models** are detailed at App.H & G.

The development of these strategies is currently manual, prompting an important question for future research: *How can we develop foundation models with greater intrinsic autonomy to automatically overcome these reasoning limitations?*

5.1 Insights from Working Memory of LMs

Small LMs are best suited to simple prompts While mitigation strategies discussed previously prove effective for larger LMs with advanced emergent abilities, small LMs encounter distinct limitations. As highlighted in §4.1, the U-shaped scal-

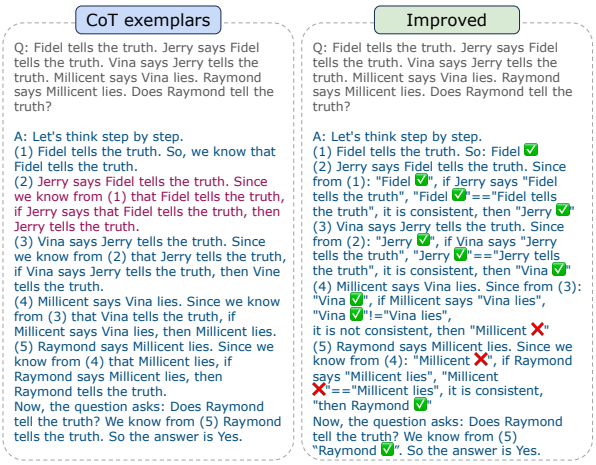

Figure 7: Comparison of vanilla CoT and enhanced CoT+ in *Web of Lies*. Steps where LLMs falter in replication and extrapolation are highlighted in red, contrasting with the improved steps shown in CoT+.

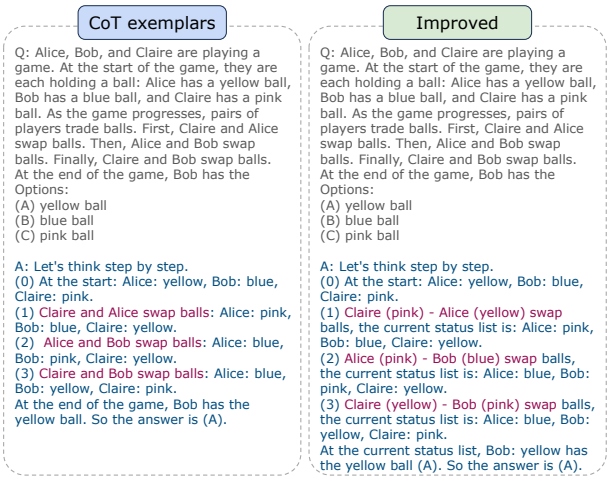

Figure 8: Comparison between CoT and improved CoT+ on *Track Shuffled Objects*.

ing phenomenon in working memory reveals that smaller models struggle with replicating complex reasoning patterns present in sophisticated prompts. Consequently, smaller LMs are suited to focusing on accurately formatting outputs rather than attempting to emulate complex reasoning steps.

Fig.3 demonstrates that smaller LMs perform more effectively with straightforward, answer-only prompts compared to more complex ones like AutoCoT, PS, and ToT. This finding indicates that prompts for smaller LMs should be designed to embrace simpler reasoning patterns and avoid overly intricate reasoning that can overwhelm their processing capabilities. For instance, by integrating the phrase *"after thinking carefully, the answer is..."* into the answer-only prompt for the *Tracking Shuffled Objects*, we increase the accuracy of text-ada-001 to 25.4%, a 15.8% improvement over the standard CoT prompt. Thus, our customized CoT+ not only cater to the capabilities of smaller models but also outperform traditional answer-only prompts by sidestepping the counterproductive complexities of other advanced prompts.

Code-style prompts facilitate logical reasoning Identifying common errors in logic-intensive tasks, such as those found in *Web of Lies*, we noted that errors frequently arose during attempts to imitate recursive truth assessments, such as *"if Jerry says that Fidel tells the truth, then Jerry tells the truth"*, which are marked in red in Fig.7.

Drawing on insights from §4.1, we have enhanced the performance of LLMs on logic tasks by incorporating code training. This training approach significantly boosts the working memory capabilities of models like text-davinci-002, given that the structured nature of code naturally aligns with logical patterns. For example, re-implementing recursive truth assessments using programming constructs such as "if-else" statements and equality checks ("==") helps to clearly represent logical operations. Following this methodological enhancement, text-davinci-002 recorded an accuracy increase to 96.4% on the task, a notable improvement of 4.8% over previous attempts.

5.2 Insights from §4.3 Human Cognition

Optimizing memory utilization through chunking According to failure cases in the *Tracking Shuffled Objects* task, LLMs often struggle to retain the color associated with each person's ball, especially in segments prominently highlighted in red in Fig.8. This task challenges the model to store three distinct data points (name, color, and swap action) simultaneously, placing considerable demand on the working memory.

As we explored previously (§4.1), consolidating these individual data points into a single, cohesive chunk can dramatically alleviate working memory load. To implement this, we revised the prompt structure to attach each person's ball color directly next to their name in the format: name(color)–name(color)–swap. This chunking approach resulted in text-davinci-002 achieving a marked accuracy improvement to 96.4%, a rise of 35.4% over the original prompt.

Utilizing rehearsal to counteract forgetting During our study of the *Object Counting* task, we

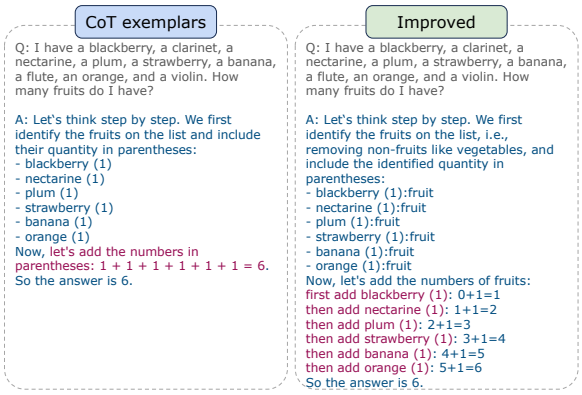

Figure 9: Comparison between CoT and improved CoT+ on *Object Counting*.

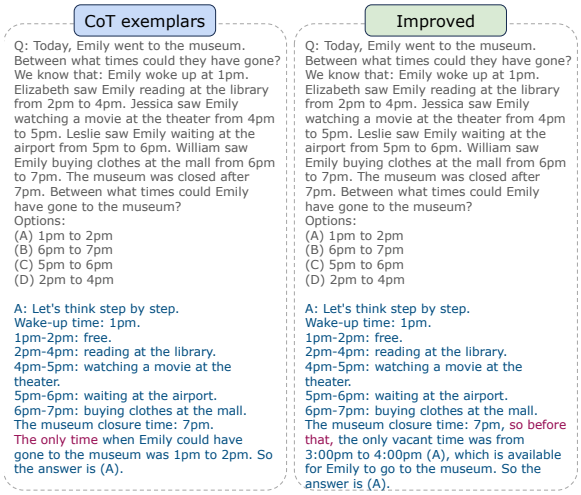

Figure 10: Comparison between CoT and improved CoT+ on *Temporal Sequence*.

noticed that LLMs frequently miscalculated the total quantities of different items, particularly highlighted in the red section of Fig.9. LLMs tended to overlook certain item types when attempting to compute all quantities in one step.

Given the complexity of amalgamating various datapoints, we adjust the prompt to enable separate counting for each item type. This is accomplished by introducing a rehearsal pattern that encourages LLMs to repeatedly recite the item name alongside its count. This method establishes a mnemonic trigger, enhancing the LLM's ability to consistently track and sum the quantities of each item type. With this modified rehearsal-triggered prompt, text-davinci-002's performance on the *Object Counting* task increased to 94.8%, reflecting an 18% improvement over the previous approach.

Enhancing LLM focus with arousal triggers Our analysis of the *Temporal Sequence* task revealed that LLMs often struggle to maintain focused attention on each duration, particularly in determining available time slots (see Fig.10). Updating the LLM's working memory for each time duration continuously proved overly demanding. Initially, using standard CoT prompts led to frequent oversight of free durations.

To address this, we introduced a specific trigger, *"so before that"*, into the prompt after reviewing all durations. This cue is designed to refocus the LLM's attention and prompt a comprehensive reassessment to identify any overlooked free durations. This technique reduces the chance of forgetting and lightens the cognitive load by structuring the response process more effectively. After integrating this attention-directing trigger, GPT-3.5's accuracy on the *Temporal Sequence* task improved dramatically to 96%, an enhancement of 32.8% compared to the original prompt.

6 Conclusion and Future Work

We explore the working memory capacities and reasoning limitations of LLMs within the context of complex cognitive tasks. Despite advancements in model scaling and the development of sophisticated prompting techniques such as CoT+ prompts, LLMs continue to exhibit significant discrepancies in performance compared to human cognitive capabilities. Our findings underscore the fundamental limitations of LLMs in autonomously recognizing and applying complex problem-solving patterns without direct human intervention. The intrinsic capacity constraints of LLMs, particularly in terms of working memory, pose a critical challenge. These limitations are not merely technical but are also a reflection of the current state of AI development, which still heavily relies on human-engineered solutions. Our mitigation strategies, while effective to some extent, primarily enhance performance in well-defined, controlled scenarios and do not universally translate to the diverse, unstructured challenges present in real-world applications.

As we move forward, it is imperative to focus on developing foundation models that can independently **plan and search** through complex logical constructs and world knowledge. Enhancing the working memory capacities of LLMs and reducing their reliance on human-crafted prompts are essential steps toward achieving this goal. Such advancements will not only bridge the gap between artificial and human intelligence but also expand the **autonomy** of LLMs in complex, dynamic, real-world environments.

Limitations

In this study, the measurement of working memory does not imply that the inherent capacity constraints of prompted language models as reasoners have been fully identified:

First, although the n-back task used is classical, it does not reflect working memory 100%. For example, some cognitive science research (Jaeggi et al., 2010) also criticizes the validity of the n-back dataset for measuring working memory because it has a weak correlation with complex working memory tasks because it relies on a different mix of cognitive processes.

Second, the n-back task is also limited for evaluating LMs when adapted to them (Gong et al., 2024). For example, while an ideal analysis of human performance on n-back tasks would include response latency, response latency is less applicable to LMs. When applied to LMs, response times are often affected by factors unrelated to biocognitive processing, such as LM deployment and API backend queue status. Also, the n-back dataset is originally proposed by (Gong et al., 2024), and the demonstration prompt of this n-back dataset may not be fully understandable for all LMs to execute the n-back. Therefore, the measured quantitative results are heuristic for working memory comparison.

Third, human participants exhibit significant variability in n-back task performance due to individual differences in physical and mental states. These are difficult to fully account for in comparative analyses with LMs. In addition, humans improve their performance through practice within a single task and receive visual-auditory-mental stimuli, in contrast to LMs that operate with static pretrained weights and do not experience such dynamic learning during testing.

In addition, our contribution goes beyond proposing a new prompt method as our experiments on BBH (23 hard tasks) show the following arguments: when CoT+ outperforms other prompts, it demonstrates that LLMs, even with advanced prompts, lack autonomy in discovering optimal problem-solving patterns. Human-tailored CoT+ shows that humans can automatically identify and design targeted prompts for specific problems, while LLMs cannot exhaustively capture diverse specific patterns autonomously (Fig.11-14 and Tab.11-14). Therefore, although CoT+ tailored from LLM mistakes requires manual crafting and cannot be automatically discovered by LLMs, this should not be seen as a weakness. Instead, it verifies the most important motivation of our study.

Acknowledgements

This research was partially funded by a Google Research Award.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*.

Richard C Atkinson and Richard M Shiffrin. 1968. Human memory: A proposed system and its control processes. *Psychology of learning and motivation*.

Alan Baddeley. 1992. Working memory. *Science*.

Alan Baddeley, Vivien Lewis, and Giuseppe Vallar. 1984. Exploring the articulatory loop. *The Quarterly Journal of Experimental Psychology Section A*.

Alan D. Baddeley and Graham Hitch. 1974. Working memory. *Psychology of Learning and Motivation*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*.

Markus Buehner, Stefan Krumm, and Marion Pick. 2005. Reasoning = working memory ≠ attention. *Intelligence*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric Poe Xing. 2023. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*.

Fergus IM Craik and Michael J Watkins. 1973. The role of rehearsal in short-term memory. *Journal of verbal learning and verbal behavior*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

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

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Annual Meeting of the Association for Computational Linguistics*.

Dongyu Gong, Xingchen Wan, and Dingmin Wang. 2024. Working memory capacity of ChatGPT: an empirical study. In *AAAI Conference on Artificial Intelligence*.

Fengyu Guo, Ruifang He, Jianwu Dang, and Jian Wang. 2020. Working memory-driven neural networks with a novel knowledge enhancement paradigm for implicit discourse relation recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*.

Beathe C Haatveit, Kjetil Sundet, Kenneth Hugdahl, Torill Ueland, Ingrid Melle, and Ole A Andreassen. 2010. The validity of d prime as a working memory index: results from the "bergen n-back task". *Journal of clinical and experimental neuropsychology*.

Xiaotian Han, Yiren Jian, Xuefeng Hu, Haogeng Liu, Yiqi Wang, Qihang Fan, Yuang Ai, Huaibo Huang, Ran He, Zhenheng Yang, et al. 2024. Infimm-webmath-40b: Advancing multimodal pre-training for enhanced mathematical reasoning. *arXiv preprint arXiv:2409.12568*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*.

Susanne M Jaeggi, Martin Buschkuehl, Walter J Perrig, and Beat Meier. 2010. The concurrent validity of the n-back task as a working memory measure. *Memory*.

Yiren Jian, Chongyang Gao, and Soroush Vosoughi. 2022. Non-linguistic supervision for contrastive learning of sentence embeddings. In *Advances in Neural Information Processing Systems*.

Yiren Jian, Chongyang Gao, and Soroush Vosoughi. 2024a. Bootstrapping vision-language learning with decoupled language pre-training. In *Advances in Neural Information Processing Systems*.

Yiren Jian, Tingkai Liu, Yunzhe Tao, Chunhui Zhang, Soroush Vosoughi, and Hongxia Yang. 2024b. Expedited training of visual conditioned language generation via redundancy reduction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

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

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. *mistral.ai*.

Wayne K Kirchner. 1958. Age differences in short-term retention of rapidly changing information. *Journal of experimental psychology*.

Patrick C. Kyllonen and Raymond E. Christal. 1990. Reasoning ability is (little more than) working-memory capacity?! *Intelligence*.

Chengshu Li, Jacky Liang, Fei Xia, Andy Zeng, Sergey Levine, Dorsa Sadigh, Karol Hausman, Xinyun Chen, Li Fei-Fei, and brian ichter. 2023a. Chain of code: Reasoning with a language model-augmented code interpreter. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2023b. Large language models with controllable working memory. In *Findings of the Association for Computational Linguistics*.

Terry Libkuman, Charles Stabler, and Hajime Otani. 2004. Arousal, valence, and memory for detail. *Memory*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024.

Let's verify step by step. In *International Conference on Learning Representations*.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2024a. Chain of hindsight aligns language models with feedback. In *International Conference on Learning Representations*.

Haogeng Liu, Quanzeng You, Yiqi Wang, Xiaotian Han, Bohan Zhai, Yongfei Liu, Wentao Chen, Yiren Jian, Yunzhe Tao, Jianbo Yuan, et al. 2024b. Infimm: Advancing multimodal understanding with an open-sourced visual language model. In *Findings of the Association for Computational Linguistics ACL*.

Ruibo Liu, Jason Wei, Shixiang Shane Gu, Te-Yen Wu, Soroush Vosoughi, Claire Cui, Denny Zhou, and Andrew M. Dai. 2023. Mind's eye: Grounded language model reasoning through simulation. In *International Conference on Learning Representations*.

Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V. Chawla. 2024c. Can we soft prompt llms for graph learning tasks? In *Companion Proceedings of the ACM Web Conference 2024*.

Ian R. McKenzie, Alexander Lyzhov, Michael Martin Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Xudong Shen, Joe Cavanagh, Andrew George Gritsevskiy, Derik Kauffman, Aaron T. Kirtland, et al. 2023. Inverse scaling: When bigger isn't better. *Transactions on Machine Learning Research*.

George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*.

OpenAI. 2023. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022a. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022b. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Zhongyu Ouyang, Chunhui Zhang, Shifu Hou, Shang Ma, Chaoran Chen, Toby Li, Xusheng Xiao, Chuxu Zhang, and Yanfang Ye. 2024. Symbolic prompt tuning completes the app promotion graph. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*.

Piotr Piekos, Mateusz Malinowski, and Henryk Michalewski. 2021. Measuring and improving BERT's mathematical abilities by predicting the order of reasoning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. *OpenAI blog*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Fobo Shi, Peijun Qing, Dong Yang, Nan Wang, Youbo Lei, Haonan Lu, Xiaodong Lin, and Duantengchuan Li. 2024. Prompt space optimizing few-shot reasoning success with large language models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Herbert A Simon. 1974. How big is a chunk? by combining data from several experiments, a basic human memory unit can be identified and measured. *Science*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics*.

Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. 2024a. Personalized pieces: Efficient personalized large language models through collaborative efforts. In *Conference on Empirical Methods in Natural Language Processing*.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2024b. Democratizing large language models via personalized parameter-efficient fine-tuning. In *Conference on Empirical Methods in Natural Language Processing*.

Yijun Tian, Huan Song, Zichen Wang, Haozhu Wang, Ziqing Hu, Fang Wang, Nitesh V Chawla, and Panpan Xu. 2024. Graph neural prompting with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

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

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

PeiFeng Wang, Aaron Chan, Filip Ilievski, Muhao Chen, and Xiang Ren. 2023b. Pinto: Faithful language reasoning using prompt-generated rationales. In *International Conference on Learning Representations*.

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

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Najoung Kim, Yi Tay, and Quoc V Le. 2023. Inverse scaling can become u-shaped. In *Conference on Empirical Methods in Natural Language Processing*.

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. 2022b. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022c. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*.

Chunhui Zhang, Chao Huang, Youhuan Li, Xiangliang Zhang, Yanfang Ye, and Chuxu Zhang. 2022. Look twice as much as you say: Scene graph contrastive learning for self-supervised image caption generation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *International Conference on Learning Representations*.

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 complex reasoning in large language models. In *International Conference on Learning Representations*.

A Additional Related Works

A.1 LMs as Reasoners

Language models (LMs) encounter significant hurdles in system-2 tasks, such as logical reasoning, which necessitate complex cognitive processes (Wang et al., 2023c; Huang et al., 2024). Initial efforts aimed at enhancing reasoning capabilities focused on specialized tasks (Geva et al., 2020; Piekos et al., 2021, *inter alia*). More recent strategies involve using prompts as a straightforward yet scalable method, enabling LLMs to address a wider array of reasoning tasks without the need for further tuning (Wei et al., 2022c; Zhou et al., 2023; Wang et al., 2023a; Yao et al., 2023a; Zhang et al., 2023; Shi et al., 2024, *inter alia*). Furthermore, advanced techniques for reasoning have been developed, such as continually training LMs on rationales and intermediate steps (Wang et al., 2023b; Liu et al., 2024a), enhancing LMs with external knowledge sources (Liu et al., 2023; Guu et al., 2020), and crafting LMs to function as actionable agents (Yao et al., 2023b; Hao et al., 2023; Tan et al., 2024a,b). Recent work has also explored language models for multi-modal understanding and reasoning tasks (Zhang et al., 2022; Jian et al., 2022, 2024b,a; Liu et al., 2024b; Han et al., 2024), expanding the capabilities of traditional language models to perform reasoning based on visual input. Also, some attempts from relational learning use language models to analyse graph data (Tian et al., 2024; Liu et al., 2024c; Ouyang et al., 2024). In our research, we focus on the simplistic yet expansible CoT prompts to delve into the intrinsic reasoning abilities of LLMs.

A.2 Scaled Language Models

Scaling up language models, primarily based on self-attention (Vaswani et al., 2017), has significantly improved natural language capabilities (Chowdhery et al., 2023). This includes bidirectional encoders such as BERT (Devlin et al., 2019) and encoder-decoders such as T5 (Raffel et al., 2020). However, decoder-only models such as the GPT series have shown superior scalability in open text generation (Radford et al., 2018, 2019; Brown et al., 2020). To better align these models with human-like interaction capabilities (such as ChatGPT), proposed techniques such as instruction tuning (Ouyang et al., 2022b; Wei et al., 2022a) and preference optimization (OpenAI, 2023; Rafailov et al., 2023) have significantly improved their utility and reliability. In this work, we analyze decoder-only language models as reasoners systematically from a scaling-up perspective.

A.3 Cognitive Chain

There is a cognitive chain that describes the process by which sensory input is transformed into long-term memory, passing through several stages: *sensory memory, short-term (or working) memory, and long-term memory.* This model, originally proposed by Atkinson and Shiffrin (1968), compares human memory processing to the information processing system of a computer.

In short, sensory memory is the initial stage where sensory input is captured for a very short period of time. When attention is paid to this information, it can be transferred to short-term memory (STM), where it is consciously processed. Short-term memory acts as a temporary storage area for information and has a limited capacity. Through the process of rehearsal — repeatedly thinking or practicing information — short-term memory can encode information into long-term memory (LTM). The detailed explanations of these concepts are listed below:

- **Sensory Memory:** The first stage involves the short-term storage of sensory information, such as sights, sounds, and tastes, lasting up to a few seconds. This stage filters vast amounts of sensory data, discarding what is deemed unimportant and transferring valuable information to short-term memory. The selective nature of sensory memory plays a crucial role in determining what information enters our cognitive system.

- **Short-Term Memory:** Often used interchangeably with working memory, short-term memory (STM) processes incoming sensory memory. While it is a component of working memory, STM specifically refers to the temporary storage system that holds information for approximately 15 to 30 seconds. It can link new sensory information with existing long-term memory. The process of rehearsal is critical at this stage because it helps move information from STM to long-term memory. Rehearsal can be active, in which information is consciously repeated, or elaborative, in which new data is linked to already known information.

- **Working Memory:** This stage involves the active manipulation of information stored in short-

term memory. Working memory is not just a passive storage system, but is also involved in processing and organizing information. Models such as that of Baddeley and Hitch (1974) propose different components of working memory, such as the visuospatial sketchpad, the episodic buffer, and the phonological loop, which are managed by a central executive. This system plays a key role in reasoning, comprehension, and learning, because it serves as the connector through which sensory inputs are filtered, processed, and encoded for short-term use or consolidated into the long-term memory for future recall, thereby connecting sensory inputs to long-term memory storage. Its primary functions include focusing attention, manipulating information through visual and auditory channels, and temporary storage of information.

- **Long-Term Memory:** The final stage, long-term memory (LTM), is where information is stored indefinitely. The capacity of LTM is believed to be limitless, encompassing everything we remember from more than a few minutes ago. Long-term memory is organized into semantic networks in which related concepts are linked, and the strength of these links depends on the frequency of association. Memory retrieval involves the process of spreading activation, in which the recall of one concept partially activates related concepts, facilitating their retrieval. Long-term memory can be either explicit (declarative) or implicit. Explicit memories are those that we consciously recall and include episodic (personal experiences) and semantic (facts and knowledge) memories. Implicit memory includes skills and routines that are performed automatically without conscious recall.

B Model Selection

Our selection of models, particularly the InstructGPT series, follows the choices made in studies by Suzgun et al. (2023) and Wei et al. (2023). The former is notable for being the first study to extract the Big-Bench Hard (BBH) reasoning tasks from the full Big-Bench dataset. Both studies use InstructGPT to investigate the scaling behavior of LMs with CoT prompts. In addition to the InstructGPT series, our study incorporates the ChatGPT series to provide a comprehensive overview of the scaling evolution of current transformative LLMs.

The InstructGPT and ChatGPT series exemplify a clear and systematic scaling evolution in the development of LLMs: recent advances in decoder-only LLMs, marked by the emergence of new capabilities, are mainly due to scaling up to billions of parameters, complemented by instruction tuning and Reinforcement Learning from Human Feedback (RLHF) techniques. This clear and systematic scaling evolution picture on the InstructGPT and ChatGPT series contrasts with other models (such as LLaMA (Touvron et al., 2023a,b), Vicuna (Chiang et al., 2023), Mistral (Jiang et al., 2023), or Mixtral (Jiang et al., 2024)), whose more convoluted and less clear methodological evolution paths make it difficult to delineate a clear and systemic scaling evolution picture.

The detailed list of models is text-ada-001, text-babbage-001, text-curie-001, text-davinci-001, text-davinci-002, GPT-3.5 (gpt-3.5-turbo-0613), and GPT-4 (gpt-4-0613), as shown in Tab.1. GPT-3's scaling to 175 billion parameters resulted in the original Davinci model. Subsequent instruction tuning refined this model into text-davinci-001. The next level model, text-davinci-002, includes additional code training. GPT-3.5 and GPT-4, part of the ChatGPT series, differ significantly from their predecessors in that they are tuned under Reinforcement Learning from Human Feedback (RLHF), designed to improve conversational modeling and alignment with human values.

C Cards of BBH Task Categorization

The BBH (Big-Bench Hard) tasks, initially shown in Figures 2 and 3, can be categorized into four distinct clusters based on two dimensions of reasoning: world knowledge and logic. In Tables 2 through 5, we present detailed cards of descriptions and rationales for each task within these four categories, offering insights into the specific reasoning challenges they encompass. For the officially released BBH repository, please refer to the original BBH paper (Suzgun et al., 2023) and its link[1], which includes the raw data for each task.

Note that while these rationales may not represent an absolute standard, these divisions nonetheless reflect a thoughtful approach to understanding the complexities and nuances of each task, highlighting key aspects of world knowledge and logic inherent in the BBH dataset, and thereby enabling further observation of LLMs' reasoning behaviors.

[1] https://github.com/suzgunmirac/BIG-Bench-Hard/tree/main

D LLMs as Reasoners: World Knowledge and Logic Assessment

D.1 Category *(i)*: Extensive World Knowledge

For category *(i)*, tasks demand extensive world knowledge — factual and general information about the world — but require less in terms of logical reasoning. These tasks present unique challenges, as advanced prompts designed to improve performance through pattern recognition are less effective compared to categories *(ii)*, *(iii)*, and *(iv)*. This is because these tasks necessitate a breadth of specific knowledge rather than logical acumen. For instance, tasks like *Sports Understanding* and *Movie Recommendation* in this category require in-depth knowledge of sports rules and history, or detailed facts about movies. Similarly, Ruin Names demands an understanding of human perception, humor in English, and familiarity with Western cultural references, such as artists, bands, and movies. Notably, the benefits from various novel prompts in GPT-3.5 are less significant when compared to scaling up the model: the average accuracy of five prompt baselines (CoT, AutoCoT, SC-CoT, PS, and ToT) is 62.93%, which exceeds the 57.1% accuracy achieved with response-only in GPT-3.5. However, this advantage of human-designed prompts diminishes emergently when scaled up to GPT-4. In this larger model, answer-only yields a much higher accuracy of 74.8%. This highlights a key limitation: prompts cannot directly imbue models with this extensive world knowledge, which is a critical bottleneck in the models' ability to infer accurate answers.

D.2 Category *(ii)*: Moderate World Knowledge and Logic

In category *(ii)*, while tasks that require a moderate level of world knowledge, they also require more logic than *(i)*. Therefore, LMs on the tasks in this category are observed to benefit a bit more from advanced prompts than *(i)*. Tasks such as *Temporal Sequences* and *Date Understanding*, which fall into this category, demand a basic understanding of time-related concepts and straightforward logical manipulation. These tasks typically involve solving simple mathematical problems related to time. Especially in models equal to or larger than text-davinci-002, advanced prompts have been shown to effectively structure the reasoning process of LLMs. Since category *(ii)* requires moderate world knowledge, the benefits of innovative prompts in GPT-3.5 are slightly diminished when models are scaled up. The average accuracy across five prompt baselines (CoT, AutoCoT, SC-CoT, PS, and ToT) is 67.6%, surpassing the 44.4% accuracy obtained with an answer-only approach in GPT-3.5. However, this advantage decreases in GPT-4, where the answer-only achieves a higher accuracy of 72.8%. This structured approach helps the recall and storage of relevant factual knowledge, thereby facilitating the logical deductions necessary to arrive at the correct answer.

D.3 Category *(iii)*: Beyond the World Knowledge and Logic

Category *(iii)* includes tasks that are beyond the realm of world knowledge and logic. These tasks often involve unique features that make them less amenable to improvement through conventional prompt patterns. Tasks such as *Snarks* require emotional intelligence, including the perception and interpretation of emotions in snarky scenarios. Similarly, tasks such as *Navigate* and *Geometric Shapes* require spatial intelligence, including spatial judgment and reasoning skills. Because LLMs lack the capacity for mental visualization, they struggle with spatial tasks under language-only but non-visual conditions, such as translating tactile sensations into spatial understanding. In addition, LLMs face significant challenges in emulating innate human abilities, particularly emotional and spatial intelligence, using explicit prompt patterns alone. The degree of improvement observed in category *(iii)* tasks using advanced prompts is notably lower (an increase of 18.74% from an answer-only accuracy of 51.4% to an average accuracy of 70.144% with advanced prompts in GPT-3.5). This is in contrast to category *(ii)* tasks involving moderate world knowledge and logic, where the improvement is more remarkable (an increase of 23.2% from an answer-only accuracy of 44.4% to an average accuracy of 67.6% with advanced prompts in GPT-3.5).

D.4 Category *(iv)*: Strong and Clear Logic

Category *(iv)* includes tasks based on strong and clear logic, where advanced prompts are particularly effective. Tasks in this category, such as *Boolean Expressions*, *Multi-Step Arithmetic*, and *Logical Deduction*, require different levels of arithmetic and logical reasoning. Because these tasks do not rely heavily on extensive world knowledge,

Task	World Knowledge	Logic
Disambiguation QA	Grammar and ambiguity knowledge	Limited
Formal Fallacies	Common sense and formal fallacies knowledge	Some understanding of formal fallacy patterns
Movie Recommendation	Movie-related knowledge (titles, themes, actors)	Minimal
Ruin Names	Understanding humor in English names (artists, bands, movies)	Limited
Salient Translation Error Detection	Knowledge of German and English	Basic understanding of grammar patterns
Sports Understanding	Sports knowledge (news, history, rules, athletes)	Limited

Table 2: Task descriptions for category *i* (extensive world knowledge) on BBH.

Task	World Knowledge	Logic
Causal Judgement	Understanding causality and common sense	Reasoning with induction and deduction
Date Understanding	Knowledge of the Western calendar and common sense	Date calculations (addition and subtraction)
Object Counting	Knowledge of categories and common sense	Completing patterns based on given instructions
Temporal Sequences	Understanding time and common sense	Identifying free slots by eliminating events in a calendar
Tracking Shuffled Object	Knowledge of taxonomy and common sense	Ordering items after exchanges
Word Sorting	Knowledge of the English alphabet and words	Separating and sorting letters by size from words and sentences

Table 3: Task descriptions for category *ii* (moderate world knowledge and logic) on BBH.

Task	World Knowledge & logic
Geometric Shapes	Requires some spatial imagination ability with minimal world knowledge and logic.
Hyperbaton	Requires a vague sense of language with minimal world knowledge and logic.
Navigate	Requires spatial imagination ability with minimal world knowledge and logic.
Penguins in a Table	Primarily requires counting ability.
Snarks	Requires a sense of humor and emotional intelligence with minimal world knowledge and logic.

Table 4: Task descriptions for category *iii* (beyond the world knowledge and logic) on BBH.

Task	World Knowledge	Logic
Boolean Expressions	Minimal	Working with Boolean values
Dyck Languages	Minimal	Identifying patterns in Dyck-n sequences
Logical Deduction	Limited	Deducing the order of a sequence of objects
Multi-Step Arithmetic	Basic understanding	Performing basic arithmetic operations
Reasoning about Colored Objects	Basic understanding of color and common items	Finding items by their color from a list
Web of Lies	Basic understanding	Working with Boolean values in a chain of transmission

Table 5: Task descriptions for category *iv* (strong and clear logic) on BBH.

they lend themselves well to the structured approaches of Advanced Prompts. These prompts help break down complex problems into smaller, more manageable sub-problems, making it easier to solve complicated, multi-step problems. Notably, tasks within category *(iv)*, including *Boolean Expressions*, *Web of Lies*, and *Reasoning about Colored Objects*, achieve (near) perfect accuracies ranging from 98.8 to 100%, underscoring the effectiveness of prompt-based assistance in improving LLM performance.

E Details of Working Memory Evaluation

E.1 n-back Test Description

The *n*-back test involves presenting participants with a sequence of stimuli. The primary task is to identify when a current stimulus matches the one that appeared n steps earlier in the sequence. The difficulty of the task is modulated by altering the load factor, denoted as n, which changes the number of steps back participants must remember.

For further clarification, the visual *n*-back test can be compared to the classic memory game of Concentration. However, there are key differences. In contrast to Concentration, where multiple items are placed at fixed locations, the *n*-back test involves a single item whose position changes with each turn. For instance, a '1-N' level necessitates remembering the position from one step back, while a '2-N' level involves recalling the position from two steps back, and so on.

Consider the simplest example of an auditory 3**-back test**, where an experimenter recites a sequence of letters to the participant. The sequence might be as follows:

G P X C H O **C** Q L **C** K **L** W M Y V B B K,

where participants are required to respond to letters that are underlined. In this test, participants need to identify when a letter, such as those highlighted in bold, matches the one that was presented three steps earlier.

The n-back task is specifically structured to actively engage the working memory system. For example, in a two-back (n = 2) test, it is not enough to simply remember the items recently presented. Participants must dynamically update their working memory buffer to accurately track and compare the current stimulus. This task requires both the maintenance and the manipulation of information within working memory.

E.2 Formulation of d'

To reveal limited working memory capacity in LLMs, we use a quantitative approach to evaluate the working memory capacity of the LMs, particularly from a scaling-up perspective. The metric for working memory evaluation on n-back test is D-prime (d').

D-prime (d') is a metric derived from signal detection theory that quantifies an individual's ability to discriminate between signal (correct matches) and noise (incorrect matches). It is calculated based on the rates of hits (correct recognitions) and false alarms (incorrect recognitions), providing a measure of sensitivity or discriminability. The formula for d' is:

$$d' = Z(\text{Hit Rate}) - Z(\text{False Alarm Rate}). \quad (1)$$

where $Z(\cdot)$ is the inverse of the cumulative distribution function for a standard normal distribution. The Hit Rate is the proportion of true positive cases that are correctly identified, while the False Alarm Rate is the proportion of negative cases that are incorrectly classified as positive.

To represent the hit rate and false alarm rate using equations, we first need to define the terms used in these equations. In the signal detection theory, the relevant terms are:

- **True Positives (TP)**: Correct recognitions of the signal.
- **False Negatives (FN)**: Failures to recognize the signal when it is present.
- **False Positives (FP)**: Incorrect recognitions of the signal when it is actually not present.
- **True Negatives (TN)**: Correct rejections of non-signal events.

With these definitions in mind, the equation for hit rate is:

$$\text{Hit Rate (HR)} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (2)$$

where the hit rate is the proportion of actual positive cases (signals) that are correctly identified. Then, the equation for the false alarm rate is:

$$\text{False Alarm Rate (FAR)} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \quad (3)$$

where the false alarm rate is the proportion of negative cases (non-signals) that are incorrectly classified as positive (signals). We use d' to compare the

working memory of OpenAI LMs and eight human users via their response results on the tasks in this paper.

E.3 Basic Information about Human Participants

The study involved a total of 8 graduate students accepted to/enrolled in US graduate schools, all of whom have passed the GRE exam. The participants were volunteers who were randomly selected to partake in the evalution. The average age of the participants was 24.5 years. The group consisted of 2 female participants, 1 participant who preferred not to declare their gender, and 5 male participants. The participants were offered the chance to win Amazon gift cards as the compensation.

F Discovery from Working Memory of Scaled LMs

F.1 Why is Working Memory Scaling U-shaped?

Beyond prior most direct findings, we also investigate atypical scaling behaviors, such as those differing from power-law or linear scaling, as reported in other studies. Wei et al. (2023) suggests that the inverse scaling law may emerge when tasks include simpler distractor activities that divert the LMs' attention from the primary, more challenging tasks. Further, McKenzie et al. (2023) identifies additional factors contributing to inverse scaling, including a tendency for models to repeat memorized sequences rather than adhering to new in-context instructions; replicating undesirable patterns in the training data; and the use of few-shot learning demonstrations that, while technically correct, may misguide the model's task performance. Inspired by these discussions and returning to our study, the factors of the initial decline in the U-shaped curve may be caused by *overemphasis on simpler heuristics:* Similar to the distractor tasks mentioned by Wei et al. (2023), the smallest text-ada-001 may prioritize simpler, more familiar patterns (e.g., a basic, repetitive replication of the input QA structure) over the more complex patterns (e.g., understanding the intention of inputs to output the responses) of the n-back tasks. In contrast, the factors underlying the latter improvement in the U-shaped curve, especially when examining larger models (text-davinci-001 and beyond), may be due to increased working memory. They handle cognitive load more easily, allowing for more complex internal representations and better memory retention. These emergent abilities are essential for n-back, which requires holding and manipulating information over time.

F.2 Larger spatial fields demand less working memory

Fig.5 demonstrates a notable improvement in working memory performance as the size of the spatial field increases, both for humans and LMs. In detail, a consistent trend observed is the increase in d' scores for both humans and LMs across the growing grid sizes (from 4×4 to 7×7). Taking the 2-back as an example, human d' scores increase from 3.42 to 3.78 to 4.04, and for GPT-4 from 3.04 to 3.15 to 3.49. Interviews from participants show that moving to larger spatial fields (larger display areas on projectors) enhances their overall experience: *firstly, it reduces eye fatigue; moreover, projectors with larger display areas are more immersive, enabling participants to become more engrossed in the task. Additionally, they can use their limbs to track and assist in remembering where targets appear, enabling an embodied environment.*

This improvement can be attributed to the cognitive benefits of spatial positioning. The larger spatial field provides better sensory affordance, naturally *triggers* participants to move their eyes and mentally shift their gaze. This movement facilitates the association of each stimulus with a distinct spatial location. As a result, the human brain instinctively records these spatial locations, effortlessly mapping new information to them, thereby reducing the load on working memory. Consequently, this spatial mapping process enhances the memorability of information and simplifies its recall. In particular, spatial positioning from larger fields triggers humans' intuitive and effortless navigation when recalling specific details about the n-back scenes. In contrast, smaller fields lack this trigger, leading to direct mapping of information in the brain, which can be more cognitively demanding and prone to distortion. Overall, a larger spatial field more clearly polarizes the relative position of objects on the grid, then triggers LLMs to track the flow of information more effectively within limited working memory.

F.3 GPT-3.5 Affected by Human-value Alignment Tax

In Fig.5, there is a decrease in working memory when moving from text-davinci-002 to GPT-3.5. When comparing text-davinci-002 and GPT-3.5,

a significant difference is that GPT-3.5, which belongs to the ChatGPT series, is tuned under Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022b; OpenAI, 2023). It is designed to enhance conversational modeling and alignment with human values, but it can also inadvertently harm other pre-existing capabilities, such as world knowledge and logical reasoning, within the model. This counter-productive phenomenon, known as the *negative alignment tax,* is first proposed by Lightman et al. (2024).

In our evaluation of working memory, since the primary expectation of the n-back task is that the model accurately recognizes input patterns and recalls recent stimuli to make correct decisions, we infer that human-value alignment in GPT-3.5 appears to detract from this working memory function, which can be recognized as the "alignment tax". When compared to the much larger GPT-4, the reasoning abilities of LMs with smaller scales, such as the aligned GPT-3.5, are more inescapably affected by this alignment tax.

G Extension on open-source LMs: Compute-Optimal Scaling Laws and Synthetic Data

Our conclusions on the foundational LM-evolving roadmap (*scaled pretraining* $\rightarrow$ *code training enhancement* $\rightarrow$ *instruction tuning* $\rightarrow$ *RLHF fine-tuning*) for the InstructGPT family are extended with findings from current open-source models, too: specifically, the newly included factors, *data-centric/compute-optimal scaling laws* and *synthetic data*, have further impacts on working memory and reasoning abilities:

When conducting experiments on open-source LLaMA-family models (Touvron et al., 2023a,b; Dubey et al., 2024), we observe that scaling along the data dimension (*data-centric/compute-optimal scaling laws*) enables the **70-billion** model to perform comparably to the GPT-4 model (which is originally larger). According to Table 7, on the logical tasks, LLaMA3-70B has a similar performance to GPT-4; on the world knowledge tasks, LLaMA3-70B underperforms compared to GPT-4 in the movie recommendation task. We attribute this to the reason that compared with GPT-4, although LLaMA3-70B contains some certain knowledge with sufficient details, there is still room to balance its knowledge proportion from the data engineering perspective.

According to Tables 8, 9, and 10, when investigating them directly from a working memory perspective, our N-back test shows that according to llama-family, the *data-centric/compute-optimal scaling laws* guiding their training model size have a significant improvement in working memory capacity, which inspires scaling compute-efficient models with more well-structured data to unlock their performance on downstream/multimodal tasks.

Due to the use of *synthetic data*, LLaMA3 models show significant improvement compared to LLaMA and LLaMA2. This positive impact on relatively small-scale LMs indicates that synthetic data enhances the model's ability to automatically decompose and solve problems, leveraging the model's reasoning ability. According to Tables 8, 9, and 10, our N-back test shows that *synthetic data* used for training the llama-3 family condenses the appropriate response behavior while facing test questions and responding in a certain format. This suggests the potential for synthetic data to guide small-scale models toward desired behavioral outcomes during training. Unlike natural data, which primarily reflects the **distribution of the results of thinking and reasoning**, synthetic data contains **the process of thought and reasoning** more comprehensively. This distinction enables the model to better simulate and internalize the steps involved in problem-solving.

H Chain-of-Thought Strikes back

As we analyzed earlier (case studies in §5), LLMs with CoT prompts cannot spontaneously discover reasoning patterns to solve the frequent failure cases. Meanwhile, even LLMs utilizing more advanced prompts are not immune to these shortcomings. Our analysis of their failure cases also indicates that the philosophies of these advanced prompts frequently fail to capture the most essential patterns required to solve these failures effectively. This deficiency in LLMs can be traced back to their fundamental limitations, specifically their lack of autonomy.

These limitations have not yet been fundamentally addressed and require further transformative changes (such as continual scaling, incorporating multisensory abilities, and interactive embodied environments). Nevertheless, by implementing our mitigation strategies inspired by cognitive science, LLMs that use the CoT+ prompt can effectively target and address key reasoning patterns especially

BBH	# Shots	Llama 2 7B	Llama 2 70B	Llama 3 8B	Llama 3 70B
zero-shot	0	32.6	51.2	28.6	50.1
CoT	3	38.2	64.9	61.1	81.3

Table 6: Performance of LLaMA-family models on the BIG-Bench Hard task. The results are reported in terms of accuracy (%).

Benchmark	Category	Llama 2 7B	Llama 2 70B	Llama 3 8B	Llama 3 70B
BBH	*i*	56.85	70.00	61.50	75.06
BBH	*ii*	33.50	68.13	65.15	82.76
BBH	*iii*	30.00	62.36	55.75	79.65
BBH	*iv*	30.02	59.06	62.00	87.73

Table 7: Detailed performance of LLaMA-family models with *CoT* on the BIG-Bench Hard task. Four categories are: *(i)* pronounced world knowledge but limited logic, *(ii)* balanced world knowledge and logic, *(iii)* deficient in both world knowledge and logic, and *(iv)* robust logic but limited world knowledge. Note that each llama model is trained with well-structured data (coda data) as their preprint mentioned, even though the model size is relatively small, their performance is still competitive on logic task *(iv)*.

Model	Task	n=1	n=2	n=3
text-ada-001	grids_4	0.80	1.89	1.08
text-babbage-001	grids_4	0.69	0.74	0.74
text-curie-001	grids_4	0.66	0.67	0.78
text-davinci-001	grids_4	0.74	0.83	0.20
text-davinci-002	grids_4	1.03	0.89	0.69
gpt-3.5	grids_4	0.82	1.01	0.68
gpt-4	grids_4	3.85	3.04	2.16
llama2-7b	grids_4	0.93	0.92	0.71
llama2-70b	grids_4	3.06	2.35	1.54
llama3-8b	grids_4	1.10	0.96	0.82
llama3-70b	grids_4	3.18	2.40	1.56

Table 8: Performance of LLaMA-family models on the N-back task (Config in 4x4 grid field). The results are reported in terms of d' sensitivity.

Model	Task	n=1	n=2	n=3
text-ada-001	grids_5	0.61	2.51	1.10
text-babbage-001	grids_5	0.21	0.60	0.77
text-curie-001	grids_5	0.77	0.71	0.70
text-davinci-001	grids_5	0.72	0.79	0.79
text-davinci-002	grids_5	1.15	0.86	0.82
gpt-3.5	grids_5	1.44	1.28	0.42
gpt-4	grids_5	3.68	3.15	2.48
llama2-7b	grids_5	1.14	0.85	0.84
llama2-70b	grids_5	3.23	2.15	2.06
llama3-8b	grids_5	1.26	1.08	0.95
llama3-70b	grids_5	3.12	2.48	2.19

Table 9: Performance of LLaMA-family models on the N-back task (Config in 5x5 grid field). The results are reported in terms of d' sensitivity.

Model	Task	n=1	n=2	n=3
text-ada-001	grids_7	1.22	1.85	1.10
text-babbage-001	grids_7	0.01	2.19	2.10
text-curie-001	grids_7	0.05	0.75	0.03
text-davinci-001	grids_7	0.01	0.72	0.04
text-davinci-002	grids_7	1.82	1.14	1.05
gpt-3.5	grids_7	1.02	1.42	0.80
gpt-4	grids_7	4.55	3.50	3.20
llama2-7b	grids_7	1.76	1.28	1.53
llama2-70b	grids_7	3.40	2.94	2.39
llama3-8b	grids_7	1.97	1.22	1.76
llama3-70b	grids_7	3.13	3.19	2.25

Table 10: Performance of LLaMA-family models on the N-back task (Config in 7x7 grid field). The results are reported in terms of d' sensitivity.

the identified common failures of BBH tasks. CoT+ allows LLMs to cover these essential patterns more accurately, as the changes from CoT to CoT+ are based on our detailed analysis and observed enhancements in addressing failure cases within BBH tasks.

H.1 Changes in CoT+ to Avoid Mistakes

In Tables 11 through 14, we provide a comprehensive overview of the prevalent failures observed in the CoT prompts across LLMs (text-davinci-002, GPT-3.5, and GPT-4) and detail the corresponding enhancements made in the CoT+ prompts for each task. Our cognitive science-inspired mitigation strategies, namely CoT+ prompts, are developed as enhancements to the classic CoT prompts, which serve as a fundamental baseline in our study.

Despite the inclusion of other more advanced

prompts (SC-CoT (Wang et al., 2023c), Plan-and-Solve (Wang et al., 2023a), tree-of-thought (Yao et al., 2023a), and AutoCoT (Zhang et al., 2023)) in our paper, these too fall short in automatically addressing the subtle and challenging patterns inherent in these common mistakes (according to our provided experimental log) and subsequently cannot bring the improvement as good as CoT+ (see Tables 11 through 14). To facilitate further research and replication, we have made the prompts and code used in our experiments available in an anonymous code repository.

H.2 Empirical Improvements of CoT+ per Task

We present the empirical results for each task, as shown in Figures 11 through 14. These figures detail the improvements observed in the transition from CoT (the vanilla implementation provided in the BBH paper (Suzgun et al., 2023)) to CoT+ (improved with our cognitive science-inspired mitigation strategies). The results demonstrate the superiority of CoT+ prompts over other advanced prompts such as SC-CoT, Plan-and-Solve, Tree-of-Thought, and AutoCoT. These improvements are particularly pronounced for tasks that require clear and structured logic.

Chain-of-Thought	**Before** Vanilla (Suzgun et al., 2023)	**After** CoT+ with cognitive science inspirations
Disambiguation QA	Phrases like *"This case makes sense because of the implicit causality of the sentence. Y was the ..., but Y ..."* actually present the results first and then explain the reasons, which confuse LMs due to the opposite of the order of cause and effect.	Modify it to align with human cognitive process on causality (i.e., the order of cause and effect) in code format, like *"If we use 'X sent a message to Y, but Y didn't reply yet,' then it means Y ..., but Y ... Therefore, it is causality. So it is True."*
Formal Fallacies	Natural language expressions can be vague, and they often involve functions with verbose names, such as *"(1) Every infrequent user of Paul Mitchell shampoo is either a rare consumer of Nioxin shampoo or a loyal buyer of Caress soap, or both: If X = infrequent-user(Paul Mitchell), then X = rare-consumer(Nioxin) OR X = loyal-buyer(Caress)."*	Break down complex information into more manageable parts, using simpler language for clarity. For example, *"1. Interpret premise 1: if a person ... 2. Interpret premise 2: No one who is a regular user... 3. Evaluate the Conclusion: Does it logically follow that ...? 4. Analyze Premise 1: ... 5. Analyze Premise 2:... 6. Conclusion: Without direct overlap between ... Thus, it is invalid."*
Movie Recommendation	Steps like *"Amongst all the options, the only movie similar to these ones seems to be The Princess Bride (1987).* do not pay attention to discussing each option's information and therefore may miss the right one.	Modify it as *"Next, consider the movies in the options: (A) They Shoot Horses (drama; 1969) and (B) Don't They? (drama; 1969). Among the options, the one that best matches ..."* to rehearse LMs to consider information about each option for better selection.
Ruin Names	Steps like *" 'One of our dinosaurs is pissing' is indeed a very whimsical and mischievous edit. This change truly ruins the original title of the movie."* lack a clear format judgment about whether they constitute a ruin.	Modify it to provide a clear trigger, such as *"... The title's meaning is changed. So it's a pun ..."*, which triggers LMs to make a clear format judgment (arousal) regarding whether it is considered a ruin and compare it with other options.
Salient Translation Error Detection	LMs often struggle to individually identify translation mistakes within sentence concepts such as Modifiers or Adjectives, Numerical Values, Negation or Antonyms, Named Entities, and Dropped Content.	To elicit inherent multilingual knowledge from LMs, we use triggers like *correct_translation = 'Artemisia is a genus of plants in the family Asteraceae."'* as an additional reference for comparison behind the given translation. This comparison helps the models identify inconsistencies more effectively, allowing them to locate the match choice without getting lost in iterating through each choice's concept.
Sports Understanding	Statements like *"Santi Cazorla is a soccer player. Touchdown is part of American football and rugby. So the answer is no."* do not straightforwardly lead to a conclusive determination of consistency.	Reformulate it from a sports knowledge QA into a logical one, as follows: *"Santi Cazorla: a soccer player. Touchdown: part of American football and rugby. Since both are different sports, it is inconsistent"* via explicit triggers like *'it is (in)consistent'*

Table 11: The patterns of common mistakes and the corresponding fixes for tasks in category *(i)*.

Chain-of-Thought	**Before** Vanilla (Suzgun et al., 2023)	**After** CoT+ with cognitive science inspirations
Causal Judgement	Natural language passages like *"A typical person would assume that this passage suggests that Frank T. had no intention of shooting and injuring someone, and that the bullet accidentally hit the neighbor's body"* can cause LMs to become lost while trying to comprehend a series of events in question.	Step-by-step, deduce the causal relation by well-organizedly rehearsals of the events from the question, such as *first, the result is 'hit the neighbor's body, causing significant injury.' Then, it is caused by 'the bullet bouncing off a large boulder several feet away'.*
Date Understanding	Natural language phrases like *"10 days before today is December 14, 1937"* are too vague to represent dates for calculation.	Use a clear trigger such as *"We check/convert the date to MM/DD/YYYY format, which is 12/24/1937. 10 days before today is 12/14/1937."* to clearly represent the date for accurate calculation.
Object Counting	Steps like *"Now, let's add the numbers in parentheses: 1 + 1 + 1 + 1 + 1 + 1 = 6."* are challenging for LMs to calculate all items at once.	Modify it as *"First, add blackberry (1): 0+1=1; then add nectarine (1): 1+1=2; then add plum (1): 2+1=3; then add strawberry (1): 3+1=4; then add banana (1): 4+1=5; finally, add orange (1): 5+1=6."* for concise chunks of `object-count` that helps LMs effectively rehearse the overall object information.
Temporal Sequences	Steps like *"... The museum closure time: 7pm. The only time when Emily could have gone to the museum was 1pm to 2pm"* can confuse LMs with numerous time durations when determining if it's a free period.	Inject the trigger *"so before that"* into the prompt after checking all the durations to redirect the LLM's attention to identify any missed free durations.
Tracking Shuffled Objects	Steps like *"(1) Claire and Alice swap balls: Alice: pink, Bob: blue, Claire: yellow."* are verbose and not compact enough in representing the `name-color` relation.	Modify it as *"(1) Claire (pink) and Alice (yellow) swap balls, updating the status list: Alice: pink, Bob: blue, Claire: yellow."* for a more compact and concise chunks of the `name-color` relation for each name's appearance.
Word Sorting	Steps like *"We now have: (3) ["costume" ? "counterpart"] < (15) "oven". Now let's sort this subpart ["costume" ? "counterpart"] by second letters."* is difficult for LMs to maintain consistent (small to large) sorting order.	Inject the trigger *"Let's sort the letters by serial number from smallest to largest."* into the prompt to explicitly rehearse LMs to maintain the small-to-large sorting order consistently.

Table 12: The patterns of common mistakes and the corresponding fixes for tasks in category *(ii)*.

Chain-of-Thought	**Before** Vanilla (Suzgun et al., 2023)	**After** CoT+ with cognitive science inspirations
Geometric Shapes	Natural language passages such as *"(1) M 41.00,43.00: Move the current point to 41.00,43.00. (2) L 37.00,34.00: Create a line from 41.00,43.00 to 37.00,34.00."* can confuse LMs when attempting to understand geometric changes.	Use concise code format and symbols as triggers to represent directions for a more intuitive understanding of geometric changes, like *"(1) M 41.00,43.00: (0,0) → (41.00,43.00). (2) L 37.00,34.00: (41.00,43.00) → (37.00,34.00).".*
Hyperbaton	Natural language representations like *"(A) has the following adjective order: [7. material] [1. opinion] (or, in numeric terms, 7 1)."* are not concise enough for sorting adjectives in ordered chunks.	Use a code format to represent it as *"(A): "rubber terrible ship": (1) rubber → material: [7. material] (2) terrible → opinion: [1. opinion] (3) ship → noun 7 < 1? False. So, (A) is False."* This concise transformation shifts from a language-based representation to a logic-based QA format, making it easier to sort adjectives in order.
Navigate	Natural language passages like *"... (1) Turn left: (0, 0), facing the negative x-axis. (2) Turn around: (0, 0), facing the positive x-axis."* can confuse LMs when trying to understand spatial concepts.	Use concise language and symbols to represent directions for a more intuitive understanding of spatial concepts, like *"... then Turn left: (0, 0), ←; then Turn left: (0, 0), ↓".*
Penguins in a Table	Natural languages like *"Vincent is 9 years old, and Gwen is 8 years old. We add James to table: James is 12 years old."* are not concise enough to record/recall information about "name, age, height, weight".	Use a code format to represent it as *"Vincent: [9, 60, 11] Gwen: [8, 70, 15] James: [12, 90, 12]"* for a more compact and concise chunks that facilitates recall and manipulation.
Snarks	Prompt like *"... it likens the consistency in the league's punishments with that in morality. Discussing the consistency of the league's punishments in the context of morality, ethics, or law makes sense and does not appear to make a satirical point about anything."* do not have a straightforward conclusion regarding consistency in given statements.	Reformulate it from an emotional question into a logical one, as in *"In (A), 'Avoiding ad hominem attacks' is often useful and helpful, so it is GOOD. Then, 'really help your case' is GOOD. GOOD != NOT GOOD, so it is inconsistent."* The explicit contrast expressions are also effective arousals/triggers.

Table 13: The patterns of common mistakes and the corresponding fixes for tasks in category *(iii)*.

Chain-of-Thought	**Before** Vanilla (Suzgun et al., 2023)	**After** CoT+ with cognitive science inspirations
Boolean Expressions	Steps like *"We first simplify this expression "Z" as follows: "Z = True and False and not True and True = A and B" where "A = True and False" and "B = not True and True." Let's evaluate A: A = True and False = False. Let's evaluate B: B = not True and True = not (True and True) = not (True) = False."* are recursive forms of reasoning, which can confuse LLMs when executing substitution operations in recursion.	Modify it from recursion to a more straightforward step-by-step pattern with chunked information like *"The order of operations: "not" > "and" > "or." 1. Evaluate 'not' first: - 'not True' becomes 'False' So now the expression is: 'True and False and False and True' 2. Evaluate 'and' from left to right: - 'True and False' becomes 'False' - ... - ... Since the 'and' operation is associative, after the first 'False' is encountered, the entire expression will evaluate to 'False'."*
Dyck Languages	Steps like *"0: empty stack (1) [; stack: [(2) { ; stack: [{"* are not clear enough to convey the concept of a stack.	Utilize a code format and symbols as chunks to represent the stack procedure, such as *"0: null (1) null ← [= [(2) [← { = [{"*, to provide a clearer explanation with less ambiguous natural language.
Logical Deduction	Steps like *"... (2) Eli finished below Amy: "(above) ? Amy ? Eli ? (below)". (3) Combining (1) and (2), we get the following ordering: "(above) Eve Amy Eli (below)"."* are vague in natural language format to represent the position relations.	Use code format to more clearly chunk the position relations, such as *"...* `status = [None, "Amy", None]` *Since Eve is above Amy, she cannot be last or in the middle, so she must be first:* `status[0] = "Eve"` `status = ["Eve", "Amy", None]` *..."*
Multi-Step Arithmetic Boolean Expressions	Steps like *"This equation can be written as "A * B", where A = (-5 + 9 * -4 - 0) and B = (4 + -7 + 0 * -5). Let's calculate A = (-5 + 9 * -4 - 0) = (-5 + (9 * -4) - 0) = (-5 + (-36) - 0) = (-5 + -36 - 0) = -5 - 36 = -41."* are recursive forms of calculation, which can confuse LLMs when executing substitution operations in recursion.	Modify it from recursion to a more straightforward step-by-step calculation for each chunked sub-problem like *" 1. First set of parentheses: - Multiply '-9*7' to get '-63' - ... - ... So the first part simplifies to: '3969' 2. Second set of parentheses: - Multiply '4*-9' to get '-36' - ... - ... So the second part simplifies to: '-40'"*
Reasoning about Colored Objects	The answering process has no effect on the reasoning process, as in *"According to this question, the color of the stress ball is blue."*	Use code formatting to concisely chunk information about objects and their colors for easy manipulation, as shown in *"First, initialize a list = ["pencil": "red","mug": "purple"] If you remove all the red objects, the updated list will be: ["mug": "purple"]"*.
Web of Lies	The natural language to represent the web of truth or false, *"Jerry says Fidel tells the truth. Since we know from (1) that Fidel tells the truth, if Jerry says that Fidel tells the truth, then Jerry tells the truth"*, is verbose and can confuse LMs.	Use simple symbols to chunk the web of truth/false, such as *"Jerry says Fidel tells the truth. Since from (1): "Fidel ✓", if Jerry says "Fidel tells the truth", "Fidel ✓" == "Fidel tells the truth", it is consistent, then "Jerry ✓"*. It is more concise and compact in chunking the relation between `name-truthiness`.

Table 14: The patterns of common mistakes and the corresponding fixes for tasks in category *(iv)*.

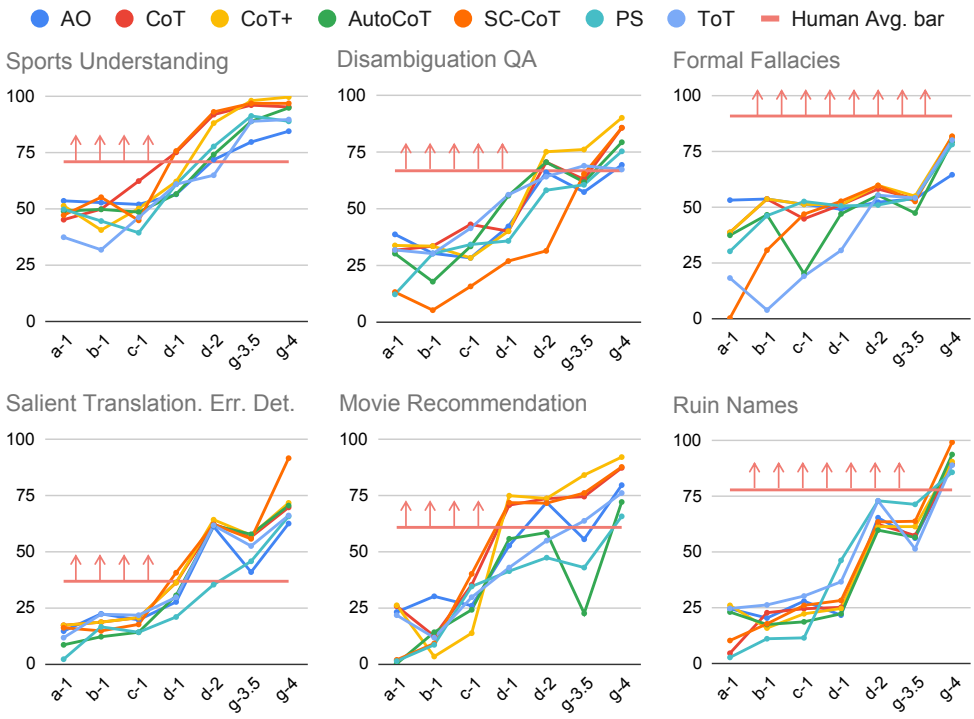

Figure 11: Accuracy results on category *(i)*: tasks requiring extensive world knowledge.

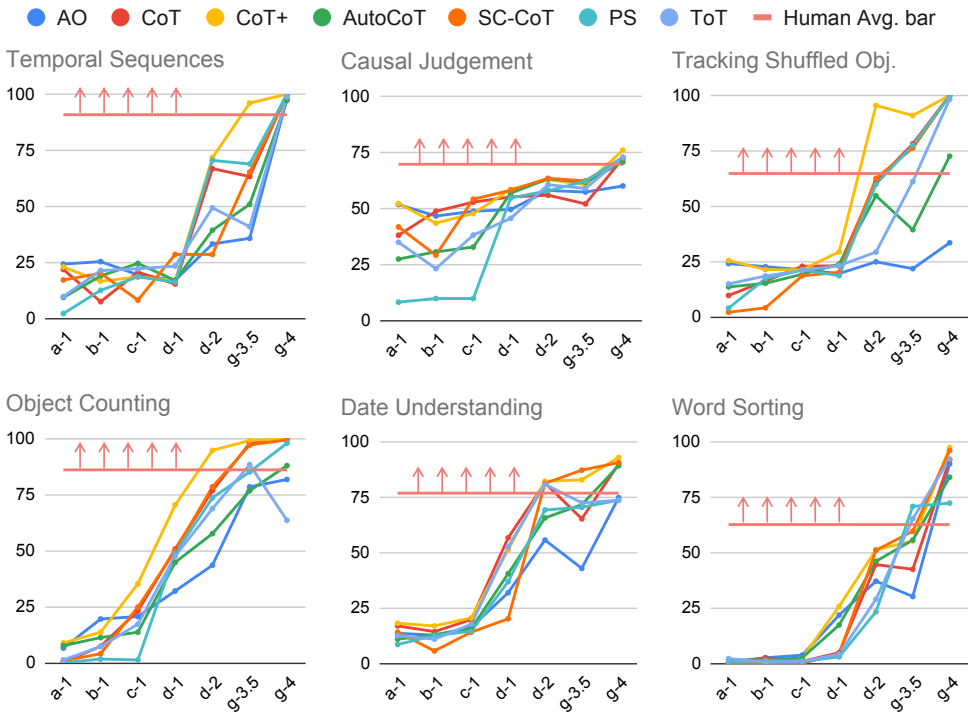

Figure 12: Accuracy results on category *(ii)*: tasks requiring moderate world knowledge and logic.

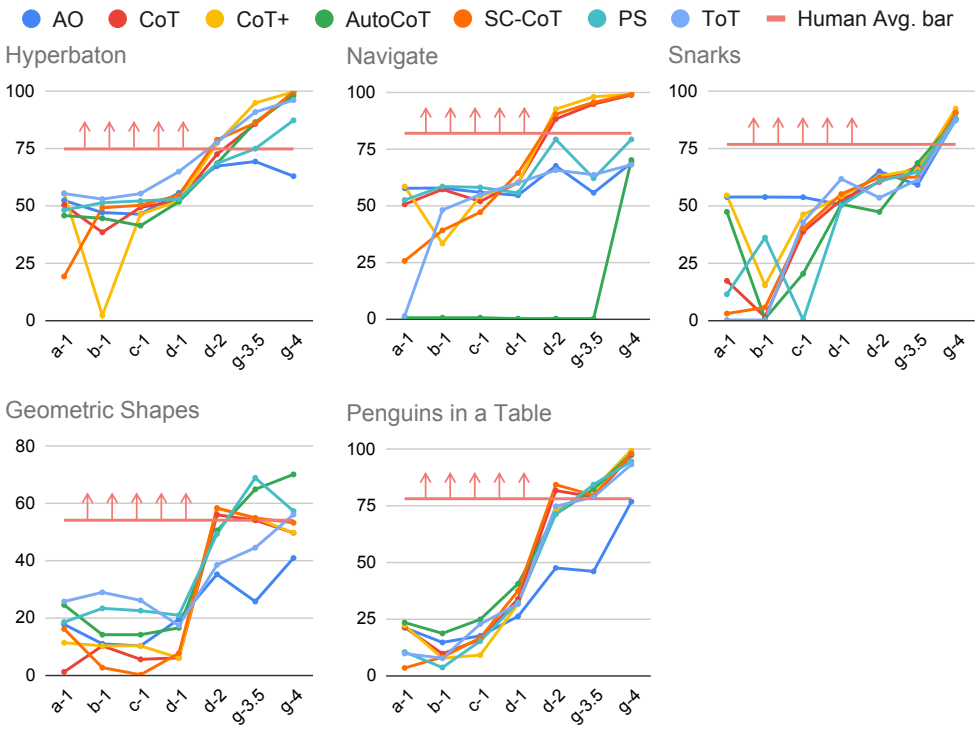

Figure 13: Accuracy results on category *(iii)*: tasks beyond the world knowledge and logic (e.g., emotional/spatial intelligence).

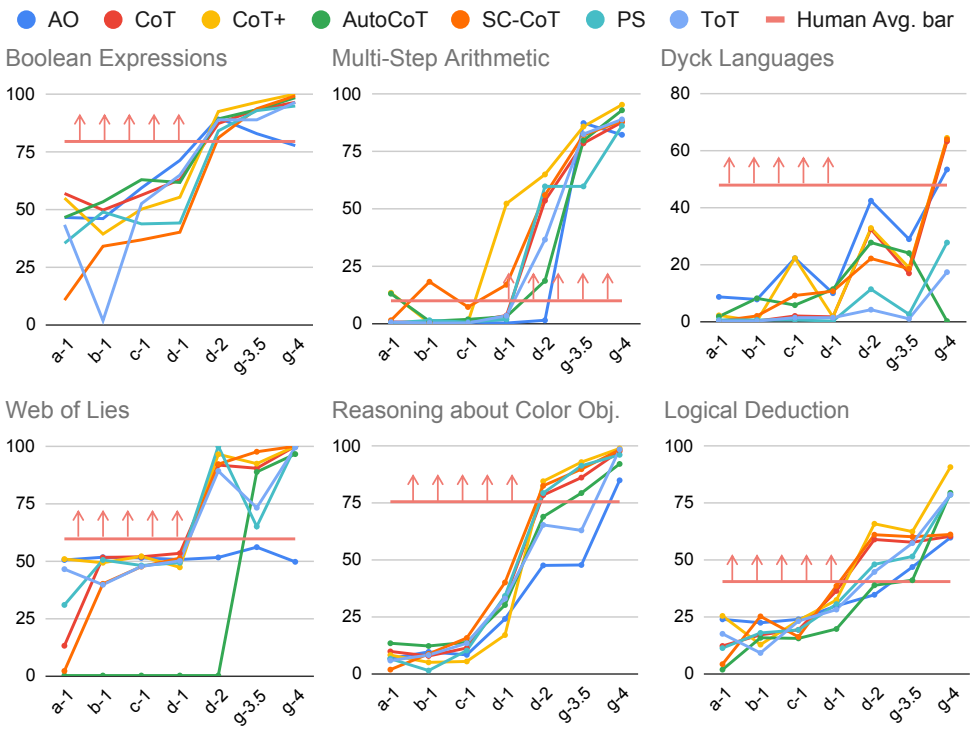

Figure 14: Accuracy results on category *(iv)*: tasks requiring strong and clear logic.