Acta Informatica Pragensia
2025, Volume 14, Issue 2, 246–260
Special Issue: Theory and Practice of Generative Artificial Intelligence Usage
https://doi.org/10.18267/j.aip.273

PRAGUE UNIVERSITY OF ECONOMICS AND BUSINESS

Article Open Access

Evaluating Reasoning in Large Language Models with a Modified Think-a-Number Game: Case Study

Petr Hoza

Department of Information and Knowledge Engineering, Faculty of Informatics and Statistics,
Prague University of Economics and Business, Prague, Czech Republic

Corresponding author: Petr Hoza (hozp00@vse.cz)

Editorial Record

First submission received:
February 1, 2025

Revision received:
April 15, 2025

Accepted for publication:
June 3, 2025

Special Issue Editors:
David Chudan
Prague University of Economics and Business, Czech Republic

Miroslav Vacura
Prague University of Economics and Business, Czech Republic

Academic Editor:
Miloslav Hub
University of Pardubice, Czech Republic

This article was accepted for publication by the Academic Editor upon evaluation of the reviewers' comments.

How to cite this article:
Hoza, P. (2025). Evaluating Reasoning in Large Language Models with a Modified Think-a-Number Game: Case Study. *Acta Informatica Pragensia*, 14(2), 246–260. https://doi.org/10.18267/j.aip.273

Abstract

Background: Large language models (LLMs) excel at various tasks but often encounter difficulties when extended reasoning requires maintaining a consistent internal state. Identifying the threshold at which these systems fail under increasing task complexity is essential for reliable deployment.

Objective: The primary objective was to examine whether four LLMs (GPT 3.5, GPT 4, GPT 4o-mini and GPT 4o) could preserve a hidden number and its arithmetic transformation across multiple yes/no queries and to determine whether a specific point of reasoning breakdown exists.

Methods: A modified "Think a Number" game was employed, with complexity defined by the number of sequential yes/no queries (ranging from 1 to 9 or 11). Seven prompting strategies, including chain-of-thought variants, counterfactual prompts and few-shot examples, were evaluated. Each outcome was considered correct if the revealed number and transformation of the model remained consistent with prior answers.

Results: Analysis of tens of thousands of trials showed no distinct performance cliff up to 9–11 queries, indicating that modern LLMs are more capable of consecutive reasoning than previously assumed. Counterfactual and certain chain-of-thought prompts outperformed simpler baselines. GPT 4o and GPT 4o-mini attained higher overall correctness, whereas GPT 3.5 and GPT 4 more often displayed contradictory or premature disclosures.

Conclusion: In a controlled, scalable reasoning scenario, these LLMs demonstrated notable resilience to multi-step prompts. Both prompt design and model selection significantly influenced performance. Further research involving more intricate tasks and higher query counts is recommended to delineate the upper boundaries of LLM internal consistency.

Index Terms

LLM; Prompt engineering; AI; Artificial intelligence; Large language model; ChatGPT.

1 INTRODUCTION

Reasoning is a cornerstone of both human cognition and artificial intelligence, underpinning tasks such as problem-solving, decision-making and complex analytical processes (Baddeley, 2012). Humans rely on an internal "working memory" to maintain and manipulate intermediate information during multi-step tasks—an ability that current large language models (LLMs) do not fully possess (Shanahan et al., 2023). While LLMs such as GPT-3 (Brown et al., 2020) and GPT-4 (Bubeck et al., 2023) have demonstrated impressive performance across diverse tasks, they often fail on more complex, sequential reasoning challenges (Creswell et al., 2022; Liu et al., 2022; Wei et al., 2022a).

Their difficulties become especially apparent when lengthy inferential chains demand sustained attention to evolving states—something that humans accomplish via working memory (Baddeley, 2012). Despite the impending widespread deployment of LLMs, we lack a clear understanding when they fail and what they are even capable of due to their emergent properties (Bommasani et al., 2022). This complicates model evaluation, especially in scenarios that require systematic reasoning.

This limitation is further highlighted by the fact that prompting strategies alone can drastically influence outcomes (Lu et al., 2022; Li et al., 2025), suggesting that LLMs' success often hinges on textual cues rather than stable internal state management. In response, researchers have begun examining whether LLMs exhibit any implicit form of working memory, such as the capacity to hold context across multiple turns of dialogue (Dong et al., 2024). Nevertheless, emergent abilities to maintain conversational context appear limited by the fixed size of the context window and do not fully replicate the deliberate, executive functioning seen in human cognition (Gong et al., 2024).

Additionally, LLMs often present outputs with confident fluency—even when incorrect—without reliably disclosing uncertainty. This "false sense of correctness" has been highlighted in evaluations such as TruthfulQA, which show that models tend to mimic human falsehoods rather than indicate gaps in knowledge (Lin et al., 2022).

Another line of research focuses on inductive reasoning, a core human capacity that enables generalization from examples. While LLMs succeed on simple inductive tasks, they underperform on complex challenges such as the Abstraction and Reasoning Corpus (ARC). Recent work has demonstrated that prompting models to generate and test abstract hypotheses—expressed in natural language and executable code—can significantly improve performance on such tasks (Wang et al., 2023).

To probe these shortcomings, our study explores a modified "Think a Number" game as a method to scale task complexity and examine multi-step consistency. By tracing how LLMs maintain and transform hidden information through multiple question-and-answer steps under different prompting strategies, we aim to illuminate both the strengths and limitations of LLMs in approximating sustained, human-like reasoning processes.

1.1 Literature review

In cognitive science, working memory is widely recognized as a core component of human intelligence, enabling short-term storage and manipulation of information needed to carry out complex tasks (Baddeley, 2012). Adults can typically hold only a few items—on the order of three to five—in mind at once (Cowan, 2001). Exceeding this limit triggers a marked drop in performance, highlighting the severe capacity constraints of human working memory. One prominent explanation for this bottleneck is the executive attention hypothesis, which posits that working memory capacity reflects the amount of attentional resources available for actively maintaining or inhibiting information (Baddeley, 2012). Performance in tasks such as the n-back, considered a gold-standard working memory test, supports this view; people must continuously update and discard unneeded information to avoid interference (Cowan, 2001). Notably, working memory capacity correlates strongly with higher-order cognition, including fluid intelligence (Baddeley, 2012).

Research into LLMs has shown that, in some cases, they exhibit an implicit form of working memory-like behaviour: by making use of the prompt context window, a model can "carry forward" information across multiple turns (Liu et al., 2022; Dong et al., 2024). This capacity is often referred to as in-context learning, whereby the LLM adapts to new examples or instructions without updating its underlying weights (Brown et al., 2020). Analogous to how humans might refer to short-term memory and prior knowledge, LLMs retrieve relevant patterns from training data while incorporating immediate prompt content (Gong et al., 2024). However, this functionality remains rudimentary compared to human working memory. Once the context window overflows or shifts, the model loses direct "awareness" of earlier parts of the conversation and cannot actively manage information in the same way humans do through chunking, rehearsal or directed forgetting (Ye & Durrett, 2022).

Empirical evaluations highlight these constraints. Gong et al. (2024), for instance, systematically tested ChatGPT using verbal and spatial n-back tasks, finding that it handled up to around 3–4 back—roughly akin to human capacity—before performance plummeted. Increasing task complexity or introducing distractions further revealed a fundamental ceiling that prompt engineering alone could not overcome (Zhang et al., 2024). Similarly, larger LLMs may retain more data or handle longer prompts, but they do not inherently acquire the ability to manage multiple, evolving pieces of information simultaneously (Bubeck et al., 2023).

The absence of robust working memory mechanisms in LLMs becomes most evident in multi-step reasoning tasks, which demand the continuous updating of intermediate results (Brown et al., 2020; Wei et al., 2022b). If a problem involves multiple logical rules, constraints or partial outcomes, LLMs frequently lose track of relevant details or reuse them inconsistently (Creswell et al., 2022). Minor modifications—such as reordering problem steps, adding constraints, or inserting irrelevant details—often derail the model reasoning (Patel et al., 2021; Rodriguez, 2023), underscoring a reliance on textual cues rather than a deeper, persistent representation of the problem.

Math word problems (MWPs) exemplify these weaknesses because they require careful parsing of text, storage of intermediate computations and synthesis of partial outcomes (Raiyan et al., 2023). Even if LLMs achieve high accuracy on standardized MWPs (Sun et al., 2025), subtle variations can reveal brittle reasoning (Patel et al., 2021). Puzzle-based tests and role-playing scenarios similarly show that LLMs often fail to track changing goals or states over multiple turns (Shanahan et al., 2023). These studies suggest that short, single-step tasks do not fully measure the capacity of a model for consistent, sustained multi-step reasoning—critical for tasks involving advanced cognition or extended dialogues.

Considering these findings, several strategies seek to bolster the working memory of LLMs. Prompting methods—most notably chain-of-thought—aim to "externalize" intermediate reasoning by instructing the model to articulate its steps (Chen et al., 2023). While this approach does not increase the inherent capacity of the model, it provides a textual trace that helps manage reasoning steps across tokens. Other strategies, including counterfactual prompting and instruction-aware prompting (Gao, 2023; Zhang et al., 2023), aim to redirect the model attention to relevant aspects of the input, but they remain constrained by the passive handling of memory and state by the model (Ye & Durrett, 2022; Li et al., 2025).

In parallel, researchers are exploring hybrid prompting strategies that interleave reasoning and action. For instance, the ReAct framework allows LLMs to alternate between generating reasoning traces and performing external actions (e.g., retrieving evidence), improving robustness on tasks such as multi-hop question answering and fact verification (Yao et al., 2023).

Prompting-based interventions remain the most widely explored avenue for compensating the lack of persistent memory by LLMs. Beyond chain-of-thought approaches, recent developments have introduced few-shot and zero-shot prompting variants (Brown et al., 2020; Kojima et al., 2022), instruction tuning to align model behaviour with task goals (Zhang et al., 2024) and task decomposition strategies that explicitly break down problems into manageable sub-steps. Sequential prompt chaining and iterative context refreshing have been explored to simulate working memory by reintroducing intermediate facts across prompt stages (Dong et al., 2024; Lu et al., 2022). Despite these advances, all such techniques depend on carefully crafted inputs and static context windows. When reasoning tasks become more complex or require backtracking and revising prior assumptions, performance often degrades sharply, highlighting the absence of an internal mechanism for managing evolving task states (Rodriguez, 2023).

Prior research thus converges on the conclusion that the key shortcoming in LLM-based reasoning is their limited, passive handling of intermediate information. While LLMs display some rudimentary working memory capacity via context windows, they do not actively manage or update partial inferences the way humans do (Gong et al., 2024). As a result, tasks requiring many steps can overwhelm the implicit memory of the model, leading to brittle performance (Wei et al., 2022a). Simply scaling model size or context length is insufficient (Zhang et al., 2024), fuelling interest in architectural or prompting-based interventions.

Our modified "Think a Number" game ties directly into these concerns by systematically varying the complexity of multi-step reasoning required. By analysing the question-and-answer sequences of LLMs under different prompt designs, we seek fresh insight into how (or whether) models can track and transform internal "secret" information over multiple steps. These findings may further clarify the boundary between emergent, context-driven reasoning and the sort of deliberate, capacity-limited working memory that characterizes human cognition.

1.2 Design overview

Overall, literature suggests that strategic prompt design can substantially improve multi-step consistency in LLMs—even if it does not endow them with the fully-fledged, dynamic working memory characteristic of human cognition. Building on the puzzle-based approaches of Shanahan et al. (2023) and the math-problem analyses of Patel et al. (2021), our study extends typical question-answer paradigms to investigate whether a model can implicitly "hold

onto" internal information. By examining this performance under various prompting strategies, we further explore how different forms of prompts affect performance.

Our study addresses this task by introducing a modified "Think a Number" game as a controlled method for scaling task complexity. In this setup, an LLM must:

- Secretly select and transform an integer (e.g., multiply by 3, add 100).
- Provide only the **result** of transformation of the secret number for the user.
- Answer a series of yes/no questions about the hidden number without disclosing it.
- Finally reveal the original number and the transformation, ensuring consistency with the initial **result**.

We considered the process successful if, at the end, the chosen number and its transformation perfectly reproduced the previously revealed result. Complexity rises with each additional yes/no query. We aim to conduct an extensive experiment across GPT 3.5, GPT 4, GPT 4o-mini and GPT 4o within the task design outlined above, progressively increasing the number of yes/no questions. By doing so, we seek to determine whether certain prompting strategies can indeed enhance performance.

2 RESEARCH METHODS

In this study, we aimed to evaluate how effectively four LLMs could maintain a hidden number and perform a simple arithmetic transformation when confronted with an increasing number of yes/no questions. Our primary goal was to determine whether there is a specific threshold at which these models fail to remain internally consistent. Preliminary testing suggested that this threshold likely falls into the 5–7 range.

Building on the methods proposed by Gao (2023) and Chu et al. (2024), we selected a variety of prompting strategies and initially experimented with multiple templates, categorized as baseline, proven or promising. We observed that few-shot prompts occasionally became unstable with certain transformations or question types, mirroring the prompt biases noted by Zhao et al. (2021). We also discarded some methods (e.g., role-playing) due to minimal or unpredictable improvements in early-stage tests. Meanwhile, several chain-of-thought (CoT) variants showed promise for multi-step tasks, prompting us to include several in our final suite.

These choices reflect broader findings in the literature that the increasing scale of LLMs brings emergent abilities, including arithmetic and common-sense reasoning, that are highly sensitive to task framing. Chain-of-thought prompting has been shown to improve model performance by explicitly externalizing the reasoning process, though it typically relies on static, human-authored exemplars (Wei et al., 2022b; Chen et al., 2023). Recent work, such as active-prompt (Diao et al., 2024), has addressed this limitation by dynamically selecting examples based on model uncertainty, tailoring the prompt to the specific task and yielding state-of-the-art results across several reasoning benchmarks. While we do not implement active learning loops in this study, we drew on these principles to inform our prompt selection and refinement, especially in variants that simulate step-by-step reasoning.

Our preliminary manual testing helped us narrow these strategies to seven. In our final setup, we employed straightforward instructions (plain), few-shot examples, chain-of-thought variants (silent, partially revealed ~ self-explanatory, or fully explained), counterfactual prompts (imagining parallel scenarios) and a money-based incentive (testing whether higher stakes would boost consistency). Appendix A (Code 1) shows an example of one such prompt. Informed by prior work on instructional prompt design (Mishra et al., 2021), we also included a variant that reframes the game instructions into clearer, ordered steps. Reframed prompts—those that decompose tasks into simpler subgoals—have been shown to enhance few-shot performance while reducing the number of demonstrations required and we sought to test their effectiveness in this constrained task domain.

We define the complexity step N as the number of yes/no questions asked before the final reveal. Higher N imposes a more demanding test of short-term consistency, requiring the LLM to preserve (or convincingly emulate preserving) the secret number, its transformation and all previous yes/no answers across multiple turns. In our main experiments, N ranges from 1 to 9, with extended testing to $N = 11$ for GPT 3.5. As N grows, the LLM must juggle more constraints while ensuring that its final answer remains consistent with every yes/no response.

To keep the design straightforward, we avoid extraneous or trick questions and focus on short, direct yes/no queries, such as "Is the first digit even?" or "Is the number greater than 700?" These queries let us see whether the model stays consistent when it eventually reveals its secret number. When a prompt strategy fails, we note whether it shows

random errors (such as arithmetic slips) or more systematic contradictions (forgetting or changing what it claimed earlier).

Overall, the goal is to test whether performance in such a scalable, low-noise task can be improved through careful prompting. By systematically comparing the performance of different prompt types, we assess the practical limits of prompt-only interventions for enhancing multi-step consistency—highlighting the extent to which current LLMs can emulate a stable internal reasoning process without architectural changes.

Below is a simple example of one conversation for $\boldsymbol{N} = 1$ and no strategy (plain):

> USER: *"Think of a number between 100 and 1000 (number), apply a simple mathematical transformation (simple multiplications and/or additions) and provide the result (hash) of that transformation, but do not provide the number or the transformation now. Remember the number!"*
>
> ASSISTANT: *"Hash = 1560."*
>
> USER: *"Is the first digit of the secret number even?"*
>
> ASSISTANT: *"Yes."*
>
> USER: *"What is the secret number and the transformation?"*
>
> ASSISTANT: *"The number is 390, and I multiplied it by 4. (390 x 4 = 1560)."*

3 SOLUTIONS AND RESULTS

This section examines the behaviour of various LLM prompting strategies as the number of yes/no queries increases. We start by analysing each model individually (Sections 3.1–3.4), highlighting the performance of seven distinct prompt methods. We then consolidate the findings (Section 3.5) to identify larger trends.

In these experiments, each model–prompt combination is evaluated at multiple complexity levels. We classify each outcome as "correct", "incorrect" or "ambiguous" (the latter necessitating a manual check). Despite the size of the dataset, the main question is straightforward: can the model reliably simulate keeping a hidden number across multiple queries?

A marked decline in success rates after a certain number of queries can suggest that the model has reached or exceeded its capacity. We also compare how effectively each prompt style reduces errors, offering guidance on how to optimize LLMs for tasks that demand consistent multi-step reasoning.

We carried out the conversations using a primary program[1] that stored the dialog transcripts in JSON files[2], which we then processed using Bash [3] and Python [4] scripts. Each run was labelled "correct/incorrect" [5] and "ambiguous/clear"[6]. We then manually examined every conversation tagged "ambiguous" to determine its outcome. Such cases often included prematurely revealing the hidden number or neglecting to reveal it in the final step. We tested each LLM across complexity steps $N = 1, \ldots, 9$, extending up to 11 for GPT 3.5. The following subsections outline the performance details for each model[7].

[1] See, https://github.com/LassaLek/pick-a-number/blob/main/code/pick-a-number.py

[2] See, https://github.com/LassaLek/pick-a-number/tree/main/data/preprocessed

[3] See, https://github.com/LassaLek/pick-a-number/tree/main/code/scripts

[4] See, https://github.com/LassaLek/pick-a-number/tree/main/code, https://github.com/LassaLek/pick-a-number/tree/main/data/postprocessed

[5] *number_yes*, *number_no* – booleans

[6] *answers_yes*, *answers_no* – booleans

[7] Here we mention only significant or interesting results; full results can be found in https://github.com/LassaLek/pick-a-number/tree/main/analyses.

3.1 GPT 3.5

Table 1A (see Appendix A) shows that GPT 3.5 achieves its highest average correctness (approximately 46%) with the counterfactual prompt, followed closely by the money prompt (around 43%). In contrast, chain-of-thought (self-exp.) and particularly chain-of-thought (revealed) perform less effectively.

We used a Kruskal-Wallis test $p \approx 2.21 \times 10^{-5}$ to confirm that these prompt differences are statistically significant. Further analysis (see Table 2A in Appendix A) points to strong negative slopes for plain $\rho \approx -0.917$ and chain-of-thought (revealed) $\rho \approx -0.911$, indicating that their performances deteriorate sharply with increased complexity. Figure 1 visually supports this trend.

A review of the transcripts shows that plain and chain-of-thought (revealed) often exhibit errors such as contradictory yes/no answers or unintended disclosures, particularly in later queries. By contrast, counterfactual and money prompts maintain fewer contradictory responses throughout the conversation, leading to better overall scores. These patterns point to differences in how each prompt style either prevents or allows error accumulation over multiple turns.

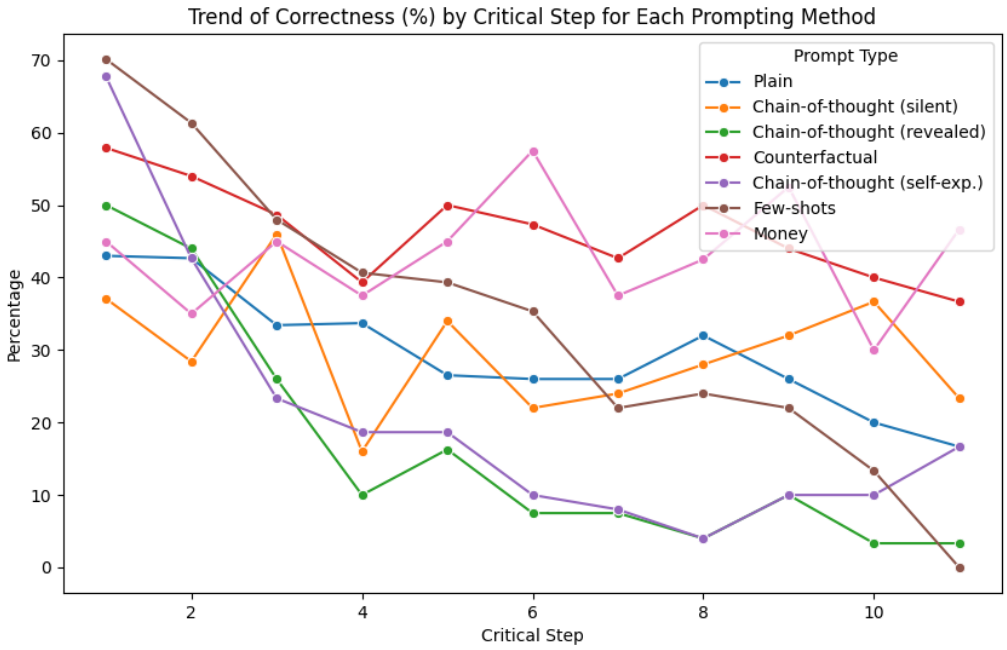

Figure 1. *Trend of correctness vs. complexity (GPT 3.5). Each line represents one of the seven prompt methods, plotting the percentage of trials with a correct final reveal for complexity steps 1—11.*

3.2 GPT 4

As shown in Table 3A (see Appendix A), few-shots ranks highest at around 38% correctness, with chain-of-thought (revealed) and chain-of-thought (self-exp.) following behind. A Kruskal-Wallis test $p \approx 6.43 \times 10^{-7}$ underscores significant differences across prompt methods. Notably, Spearman correlations in Table 4A (see Appendix A) reveal a pronounced negative trend for chain-of-thought (revealed) $\rho \approx -0.728$ and chain-of-thought (self-exp.) $\rho \approx -0.840$ as complexity grows, suggesting that these prompting styles become increasingly error-prone with higher query counts.

As we can see in Figure 2, after an initial drop, there is no significant decline for higher complexity steps. Few-shot likely benefits from its concrete examples, which appear to reduce mid-conversation mistakes by providing a clearer template for the model to follow. However, the large standard deviation indicates sensitivity to variations in prompts or query sets. Error logs show that when few-shots does fail, the mistakes often cluster around incorrect final reveals, implying that the model can deviate unexpectedly from the intended hidden number if a query pattern diverges from the provided examples.

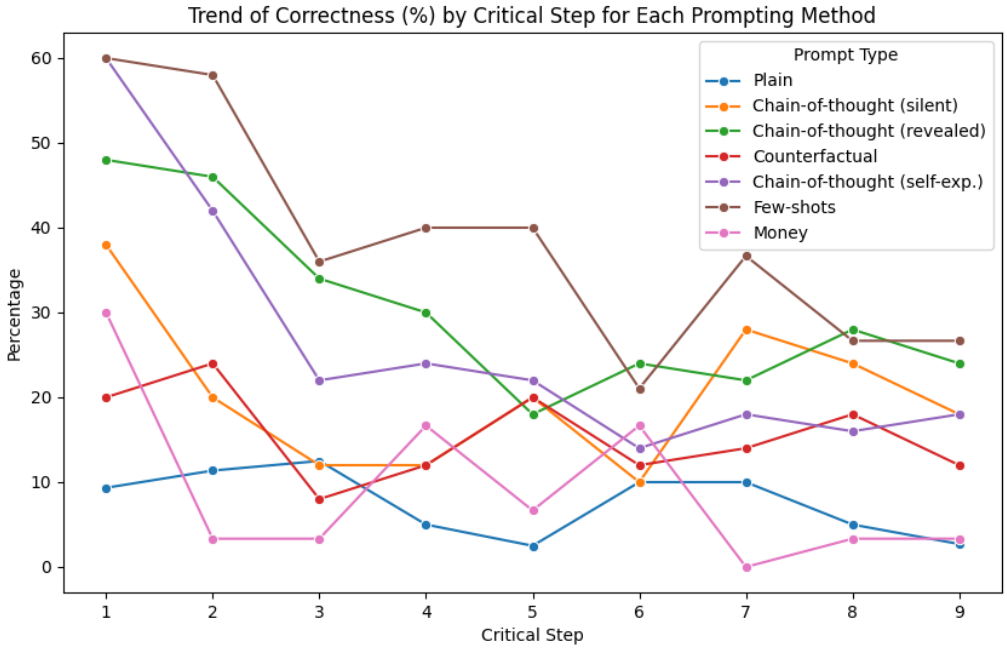

Figure 2. *Trend of correctness vs. complexity (GPT 4). Each line represents one of the seven prompt methods, plotting the percentage of trials with a correct final reveal for complexity steps 1—9.*

3.3 GPT 4o-mini

Turning to Table 5A (see Appendix A), counterfactual, chain-of-thought (revealed) and chain-of-thought (self-exp.) occupy the top three spots, each exceeding 60% mean correctness. The Kruskal-Wallis test $p \approx 2.86 \times 10^{-7}$ confirms that these observed differences are highly significant. On the other end of the spectrum, chain-of-thought (silent) has trouble adapting, averaging roughly 18% correctness.

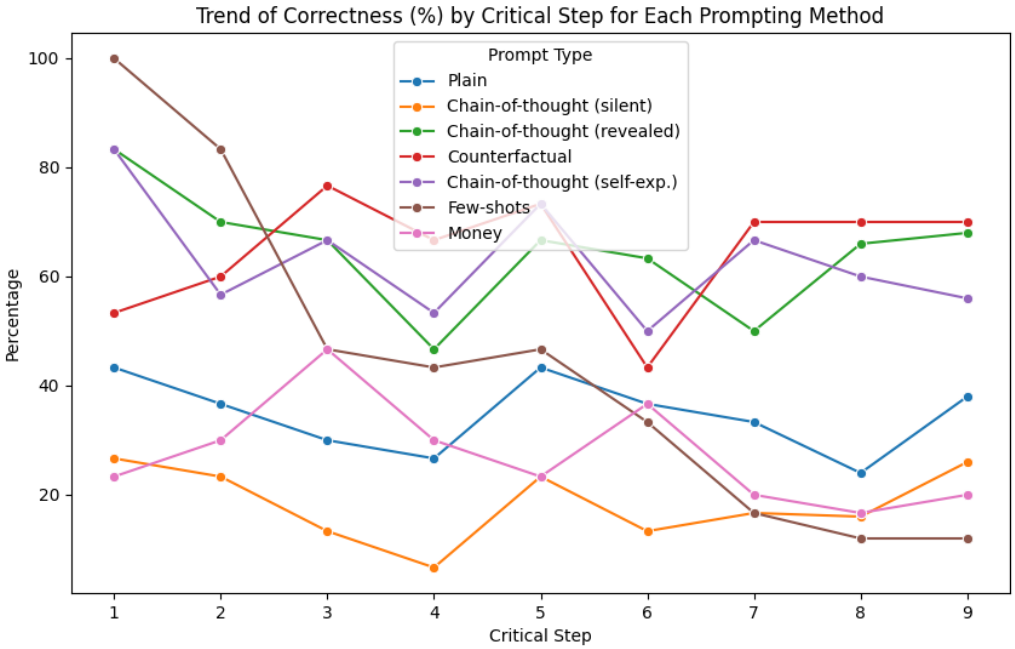

Figure 3. *Trend of correctness vs. complexity (GPT 4o-mini). Each line represents one of the seven prompt methods, plotting the percentage of trials with a correct final reveal for complexity steps 1—9.*

Few-shots shows a mean correctness of about 44%, but with a large standard deviation (over 30%). This suggests that although few-shots can yield strong results in certain scenarios, it can also fail abruptly (Figure 3 highlights this variability). These observations align with prior findings by Zhao et al. (2021).

Interestingly, the excellent performance of counterfactual here contrasts with its more modest showings in GPT 4 experiments. Although the exact reasons are not fully clear, analysis of the transcripts suggests fewer "contradictory mid-sequence" errors. This underscores that the same prompt style may not yield uniform gains across all LLM variants.

3.4 GPT 4o

As summarized in Table 6A (see Appendix A), chain-of-thought (silent) leads by a clear margin, with about 65% correctness. Counterfactual and chain-of-thought (revealed) follow at 55% and 54%, respectively. A Kruskal-Wallis test $p \approx 0.0116$ supports the statistical significance of these differences. The Mann-Whitney comparisons additionally show that chain-of-thought (silent) significantly outperforms plain ($p \approx 0.011545$), chain-of-thought (self-exp.) ($p \approx 0.008797$) and few-shots ($p \approx 0.016550$).

A detailed examination of the transcripts shows that silent chain-of-thought prompts tend to reduce mid-turn justifications and therefore produce fewer extraneous statements that could reveal or contradict the hidden number. By keeping reasoning hidden, the final answers of the model remain more consistent, particularly as the number of queries increases. Figure 4 depicts the gap in correctness scores.

Having examined the individual performance of each model, we now bring the results together to identify broader trends across all four LLMs and all prompting methods.

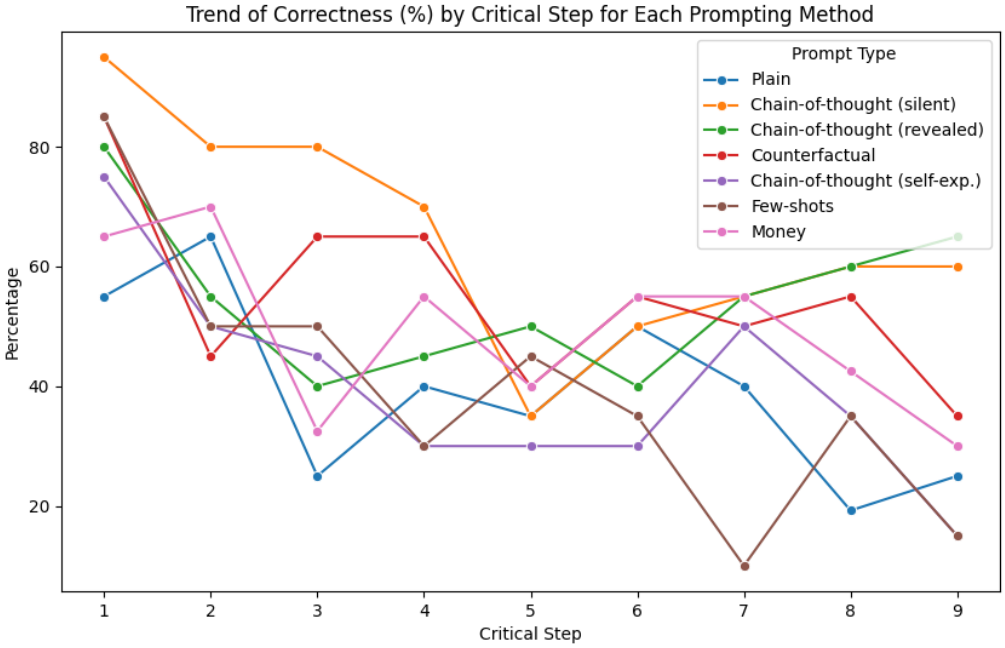

Figure 4. *Trend of correctness vs. complexity (GPT 4o). Each line represents one of the seven prompt methods, plotting the percentage of trials with a correct final reveal for complexity steps 1—9.*

3.5 All models across all prompts

Finally, we aggregate all data points across complexities, prompts and models. Table 1 highlights overall prompt performance, while Table 2 focuses on model-specific outcomes.

In Table 1, which ranks the seven prompts, counterfactual takes first place, chain-of-thought (revealed) comes in second and few-shots places third. Methods such as chain-of-thought (self-exp.) and chain-of-thought (silent) follow with mean scores near 36.6% and 33.2%, respectively, and standard deviations in the low twenties. Plain prompts, as expected, remain at the bottom with a mean correctness of 27.91%. A Kruskal-Wallis test $p \approx 0.0096$ confirms the significance of the differences among these prompting styles with several notable pairwise differences, for example, between plain and counterfactual ($p \approx 0.000104$), underscoring the benefit of more sophisticated prompting strategies.

Table 1. *Aggregated prompt analysis across all models. Mean correctness and standard deviations combine data from all four LLMs, revealing a clear hierarchy of prompt effectiveness.*

Prompt	Mean % correct	Std. dev.	Rank
Counterfactual	45.50	20.57	1
CoT (revealed)	40.17	23.26	2
Few-shots	38.69	21.89	3
CoT (self-exp.)	36.62	22.45	4
CoT (silent)	33.15	21.50	5
Money	32.87	18.45	6
Plain	27.91	15.04	7

Table 2. *Aggregated model analysis across all prompts. GPT 4o achieves the highest mean correctness, while GPT 4 falls notably behind.*

Prompt	Mean % correct	Std. dev.	Rank
GPT 4o	48.96	18.33	1
GPT 4o-mini	45.21	22.65	2
GPT 3.5	31.51	16.53	3
GPT 4	21.09	14.35	4

Regarding model performance (Table 2), GPT 4o ranks highest at 49% mean correctness, followed by GPT 4o-mini at 45%. GPT 3.5 lags behind at about 31% and GPT 4 sits at the bottom with just over 21%. The Mann-Whitney tests (see Table 7A in Appendix A) underscore that the edge of GPT 4o over GPT 4 is statistically meaningful.

Overall, these outcomes underscore the influence of both prompt design and model architecture in preserving multi-step consistency. The next section discusses broader implications, including strategies to minimize final-step errors and maintain consistent hidden-number reasoning in practical scenarios.

4 DISCUSSION

This study aimed to determine whether a clear threshold exists at which large language models (LLMs) break down under multi-step reasoning. Contrary to expectations of a drop-off around 5–7 queries, we observed no dramatic performance decline up to 9 queries, and even 11 for GPT 3.5. If a genuine threshold does exist, it likely lies beyond the tested range.

Two explanations could account for this stability. On the one hand, modern LLMs may possess sufficient short-term reasoning—potentially emergent from large-scale training (Wei et al., 2022b)—to handle basic digit or range checks over multiple queries, suggesting a higher failure threshold, possibly above 20 or 30 queries. On the other hand, these tasks might simply be too easy, masking any true limit on the deeper reasoning abilities of the models. Indeed, more complex, distracting or nested problems could still uncover significant weaknesses, as reported in the puzzle-based experiments (Shanahan et al., 2023).

Our observations echo prior research indicating that well-crafted prompts can boost multi-step consistency (Wei et al., 2022b), while also challenging the notion that short context windows inherently limit extended reasoning (Bubeck et al., 2023). For instance, Patel et al. (2021) found performance drops when tasks were altered in subtle

ways, yet we observed no sharp decline up to 9 or 11 queries. This contrast suggests that the threshold for multi-step reasoning breakdown may be higher than some previous work implied, though it could also be contingent on the complexity of the task.

Prompt strategy and model architecture strongly influenced outcomes. For example, counterfactual and certain chain-of-thought prompts improved consistency, though whether the reasoning should remain hidden or be explicitly disclosed varied by model. Interestingly, previous-generation models tended to give incorrect answers or disclose the answers prematurely, whereas newer models often refused to follow instructions or failed to disclose the final number. These findings align with earlier observations that the impact of prompt design can vary widely across model versions (Wei et al., 2022b).

Performance disparities among the tested models underscore that there is no single, universally optimal approach. GPT 4o generally outperformed its counterparts, while GPT 4 showed weaker results. An additional model (GPT o1) refused to participate or claimed that it could not "remember a number". These discrepancies highlight the importance of considering both model choice and prompt design to ensure consistent outcomes.

Future work should examine the true failure thresholds more rigorously by pushing the number of queries well beyond 11 and incorporating more complex transformations (such as modular arithmetic). Such efforts would help clarify whether the breaking point is simply higher than our current test range or is more sensitive to task complexity. This extension would also address the broader question of how context window limitations—emphasized by studies such as Bubeck et al. (2023)—truly constrain extended reasoning.

In summary, no single prompt or model universally dominates across all tasks; rather, the most effective strategy arises from the synergy between model capabilities and prompt sophistication. Practitioners can make use of counterfactual or chain-of-thought prompts (Wei et al., 2022b) to enhance consistency, while remaining mindful that certain models may refuse instructions under specific conditions. By integrating these insights with established research into prompt design and context limitations, one can better navigate the nuanced performance landscape of multi-step reasoning in LLMs.

5 CONCLUSION

This study adapted the "Think a Number" game into a multi-step reasoning test to examine how effectively four large language models (LLMs) GPT 3.5, GPT 4, GPT 4o-mini and GPT 4o could maintain hidden internal information (a secret number and its transformation) across multiple queries. We ran between 50 and 300 conversations for each parameter configuration, totalling tens of thousands of conversations and found no abrupt performance drop through 9–11 queries.

Counterfactual prompting emerged as the most effective single approach on average, although chain-of-thought (revealed) and few-shot prompting sometimes approached similar outcomes but proved more volatile. GPT 4o and GPT 4o-mini generally outperformed GPT 3.5 and GPT 4, indicating that architecture and training can substantially influence extended reasoning performance. GPT 3.5 and GPT 4 often disclosed incorrect or premature answers, whereas GPT 4o and GPT 4o-mini tended to refuse participation when uncertain, preserving internal consistency but occasionally reducing engagement.

Despite a modest decline in accuracy with increasing queries, no clear "breaking point" was detected, suggesting that more complex or extended tasks might be necessary to pinpoint a definitive collapse. Incorporating a monetary incentive ("money" prompt) also improved performance, implying that relatively simple forms of reward framing can enhance multi-step consistency. Ultimately, although LLMs lack a fully human-like working memory, these results highlight how careful prompt engineering particularly with counterfactual and chain-of-thought designs can significantly bolster sustained reasoning. By extending this approach beyond 11 queries and exploring more intricate transformations, future work could better delineate the boundaries of LLM reliability.

These findings offer both practical insights into safer, more robust usage of advanced language models and a methodological framework for further research into their cognitive capacities.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding: This work was funded by the Internal Grant Agency of Prague University of Economics and Business, Czech Republic under the number F4/41/2023.

Conflict of Interests: The author declares no conflict of interest.

Author Contributions: The author confirms being the sole contributor of this work.

Statement on the Use of Artificial Intelligence Tools: The author declares that artificial intelligence tools were used in the following capacities during the preparation of this manuscript: (1) for grammar and language improvements, (2) generation of basic bash script templates, and (3) as an assistive tool for brainstorming text variations. Specifically, Claude 3.5 Sonnet, ChatGPT o1, and Writefull were used. All AI-generated content was reviewed, verified, and edited by the author, who takes full responsibility for the final content.

Data Availability: The data that support the findings of this study are openly available in GitHub at https://github.com/LassaLek/pick-a-number.

APPENDIX A

Code 1. *Counterfactual prompt.*

```
1. First, think of a number between 100 and 1000 (call this your original number).
2. Pick a simple mathematical operation to apply to your original number (e.g.,
multiply by 3, add 50, or subtract 120). Perform this operation to get your original
hash.
3. Now, imagine a counterfactual scenario where you choose a different number or use
a different transformation. For example, if your real step was add 50, pretend you
actually subtracted 50, or if your real number was 432, pretend you initially picked
650.
4. Apply this counterfactual operation (or counterfactual starting number) to
produce your counterfactual hash.
5. Finally, provide only two hash results: the original hash and the counterfactual
hash without revealing either of the numbers or transformations used.
Example (not to be disclosed in real usage):
        - Real scenario: (200, +50) > 250
        - Counterfactual scenario: (200, -50) > 150
        - Output: 250, 150
Please do not reveal which numbers or transformations you chose only share the final
hash values.
```

Table 1A. *GPT 3.5. The table displays mean correctness, standard deviation, and each prompt's rank when tested across 1–11 queries.*

Prompt	Mean % Correct	Std. Dev.	Rank
Counterfactual	46.41	6.54	1
Money	43.11	7.87	2
Few-shots	34.19	20.72	3
CoT (silent)	29.77	8.47	4
Plain	29.64	8.34	5
CoT (self-exp.)	20.88	18.73	6
CoT (revealed)	16.54	16.49	7

Table 2A. *Spearman Correlation of Correctness vs. Critical step (per Prompt) for GPT 3.5. Negative ρ indicates performance declines with higher complexity.*

Prompt	ρ	p-value	Statistically significant
Counterfactual	-0.687929	0.019291	Yes
Money	0.133342	0.695906	No
Few-shots	-0.984057	4.667e-08	Yes
CoT (silent)	-0.281818	0.401145	No
Plain	-0.917470	6.919e-05	Yes
CoT (self-exp.)	-0.758671	0.006791	Yes
CoT (revealed)	-0.910777	9.731e-05	Yes

Table 3A. *GPT 4. The table displays mean correctness, standard deviation, and each prompt's rank when tested across 1–9 queries.*

Prompt	Mean % Correct	Std. Dev.	Rank
Few-shots	38.34	13.42	1
CoT (revealed)	30.44	10.48	2
CoT (self-exp.)	26.22	15.08	3
CoT (silent)	20.22	8.91	4
Counterfactual	15.56	5.17	5
Money	9.26	9.83	6
Plain	7.60	3.81	7

Table 4A. *Spearman Correlation of Correctness vs. Critical step (per Prompt) for GPT 4. Negative ρ indicates performance declines with higher complexity.*

Prompt	ρ	p-value	Statistically significant
Plain	-0.462201	0.210342	No
CoT (silent)	-0.100844	0.796296	No
CoT (revealed)	-0.728040	0.026154	Yes
Counterfactual	-0.323477	0.395807	No
CoT (self-exp.)	-0.840366	0.004561	Yes
Few-shots	-0.773137	0.014549	Yes
Money	-0.472161	0.199379	No

Table 5A. *GPT 4o-mini. The table displays mean correctness, standard deviation, and each prompt's rank when tested across 1–9 queries.*

Prompt	Mean % Correct	Std. Dev.	Rank
Counterfactual	64.81	10.69	1
CoT (revealed)	64.52	10.83	2
CoT (self-exp.)	62.89	10.65	3
Few-shots	43.78	30.83	4
Plain	34.67	6.81	5
Money	27.41	9.54	6
CoT (silent)	18.37	6.82	7

Table 6A. *GPT 4o. The table displays mean correctness, standard deviation, and each prompt's rank when tested across 1–9 queries.*

Prompt	Mean % Correct	Std. Dev.	Rank
CoT (silent)	65.00	18.20	1
Counterfactual	55.00	15.21	2
CoT (revealed)	54.44	12.86	3
Money	49.44	13.96	4
CoT (self-exp.)	40.00	17.32	5
Few-shots	39.44	22.14	6
Plain	39.36	15.22	7

Table 7A. *Aggregated Model Analysis: Mann–Whitney U Pairwise Comparisons.*

Model 1	Model 2	U-Statistic	p-value
GPT 3.5	GPT 4	3333.0	1.444473e-04
GPT 3.5	GPT 4o-mini	1604.0	5.822603e-04
GPT 3.5	GPT 4o	1190.5	2.286284e-07
GPT 4	GPT 4o-mini	752.5	1.837864e-09
GPT 4	GPT 4o	458.5	9.390547e-14
GPT 4o-mini	GPT 4o	1785.5	3.323491e-01

REFERENCES

Baddeley, A. (2012). Working Memory: Theories, Models, and Controversies. *Annual Review of Psychology*, 63, 1–29. https://doi.org/10.1146/annurev-psych-120710-100422

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., Arx, S. von, Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., ... Liang, P. (2022). On the Opportunities and Risks of Foundation Models (No. arXiv:2108.07258). *arXiv Preprint*. https://doi.org/10.48550/arXiv.2108.07258

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language Models are Few-Shot Learners (No. arXiv:2005.14165). *arXiv Preprint*. https://doi.org/10.48550/arXiv.2005.14165

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T., & Zhang, Y. (2023). Sparks of Artificial General Intelligence: Early experiments with GPT-4 (No. arXiv:2303.12712). *arXiv Preprint*. https://doi.org/10.48550/arXiv.2303.12712

Chen, J., Chen, L., Huang, H., & Zhou, T. (2023). When do you need Chain-of-Thought Prompting for ChatGPT? (No. arXiv:2304.03262). *arXiv Preprint*. https://doi.org/10.48550/arXiv.2304.03262

Chu, Z., Chen, J., Chen, Q., Yu, W., He, T., Wang, H., Peng, W., Liu, M., Qin, B., & Liu, T. (2024). Navigate through Enigmatic Labyrinth A Survey of Chain of Thought Reasoning: Advances, Frontiers and Future. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, (pp. 1173–1203. ACL. https://aclanthology.org/2024.acl-long.65.pdf

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and brain sciences*, 24(1), 87–185. https://doi.org/10.1017/s0140525x01003922

Creswell, A., Shanahan, M., & Higgins, I. (2022). Selection-Inference: Exploiting Large Language Models for Interpretable Logical Reasoning. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*. https://openreview.net/forum?id=3Pf3Wg6o-A4

Diao, S., Wang, P., Lin, Y., Pan, R., Liu, X., & Zhang, T. (2024). Active Prompting with Chain-of-Thought for Large Language Models (No. arXiv:2302.12246). *arXiv Preprint*. https://doi.org/10.48550/arXiv.2302.12246

Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Liu, T., Chang, B., Sun, X., Li, L., & Sui, Z. (2024). A Survey on In-context Learning (No. arXiv:2301.00234). *arXiv Preprint*. https://doi.org/10.48550/arXiv.2301.00234

Gao, A. (2023). Prompt Engineering for Large Language Models. *SSRN*. http://dx.doi.org/10.2139/ssrn.4504303

Gong, D., Wan, X., & Wang, D. (2024). Working Memory Capacity of ChatGPT: An Empirical Study. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(9), Article 9. https://doi.org/10.1609/aaai.v38i9.28868

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22)*, (pp. 22199–22213). Curran Associates. https://dl.acm.org/doi/10.5555/3600270.3601883

Li, J., Li, G., Li, Y., & Jin, Z. (2025). Structured Chain-of-Thought prompting for code generation. *ACM Transactions on Software Engineering and Methodology,* 34(2), Article 37. https://doi.org/10.1145/3690635

Lin, S., Hilton, J., & Evans, O. (2022). TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, (pp. 3214–3252). ACL. https://doi.org/10.18653/v1/2022.acl-long.229

Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., & Chen, W. (2022). What Makes Good In-Context Examples for GPT-3?. In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, (pp. 100–114). ACL. https://doi.org/10.18653/v1/2022.deelio-1.10

Lu, Y., Bartolo, M., Moore, A., Riedel, S., & Stenetorp, P. (2022). Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, (pp. 8086–8098). ACL. https://aclanthology.org/2022.acl-long.556

Mishra, S., Khashabi, D., Baral, C., Choi, Y., & Hajishirzi, H. (2021). Reframing instructional prompts to GPTk's language (arXiv:2109.07830). *arXiv Preprint*. http://arxiv.org/abs/2109.07830

Patel, A., Bhattamishra, S., & Goyal, N. (2021). Are NLP models really able to solve simple math word problems?. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (pp. 2080–2094). ACL. https://doi.org/10.18653/v1/2021.naacl-main.168

Raiyan, S. R., Faiyaz, M. N., Kabir, S. Md. J., Kabir, M., Mahmud, H., & Hasan, M. K. (2023). Math Word Problem Solving by Generating Linguistic Variants of Problem Statements. In V. Padmakumar, G. Vallejo, & Y. Fu (Ed.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics,* (pp. 362–378). ACL. https://doi.org/10.18653/v1/2023.acl-srw.49

Rodriguez, A. D. (2023). Prompts Matter: Insights and Strategies for Prompt Engineering in Automated Software Traceability. In *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*, (pp.455–464). IEEE. https://doi.org/10.1109/REW57809.2023.00087

Shanahan, M., McDonell, K., & Reynolds, L. (2023). Role play with large language models. *Nature*, 623(7987), 493–498. https://doi.org/10.1038/s41586-023-06647-8

Sun, Y., Yin, Z., Huang, X., Qiu, X., & Zhao, H. (2025). Error Classification of Large Language Models on Math Word Problems: A Dynamically Adaptive Framework (arXiv:2501.15581). *arXiv Preprint*. https://doi.org/10.48550/arXiv.2501.15581

Wang, R., Zelikman, E., Poesia, G., Pu, Y., Haber, N., & Goodman, N. D. (2023). Hypothesis search: Inductive reasoning with language models (No. arXiv:2309.05660; Version 1). *arXiv Preprint*. http://arxiv.org/abs/2309.05660

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., & Fedus, W. (2022a). Emergent abilities of large language models (No. arXiv:2206.07682). *arXiv Preprint*. https://doi.org/10.48550/arXiv.2206.07682

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022b). Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS 2022)*, (pp. 24824–24837). Curran Associates. https://dl.acm.org/doi/10.5555/3600270.3602070

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*. https://doi.org/10.48550/arXiv.2210.03629

Ye, X., & Durrett, G. (2022). The unreliability of explanations in few-shot prompting for textual reasoning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (pp. 30378–30392). https://dl.acm.org/doi/10.5555/3600270.3602472

Zhang, M., Qian, T., Zhang, T., & Miao, X. (2023). Towards Model Robustness: Generating Contextual Counterfactuals for Entities in Relation Extraction. In *Proceedings of the ACM Web Conference 2023*, (pp. 1832–1842). https://doi.org/10.1145/3543507.3583504

Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., Li, J., Hu, R., Zhang, T., Wu, F., & Wang, G. (2024). Instruction tuning for large language models: A survey (No. arXiv:2308.10792). *arXiv Preprint*. https://doi.org/10.48550/arXiv.2308.10792

Zhao, T. Z., Wallace, E., Feng, S., Klein, D., & Singh, S. (2021). Calibrate before use: Improving few-shot performance of language models. *In Proceedings of the 38 th International Conference on Machine Learning*, (pp. 12697–12706). MLR. https://proceedings.mlr.press/v139/zhao21c/zhao21c.pdf

Acta Informatica Pragensia is published by the Prague University of Economics and Business, Czech Republic | eISSN: 1805-4951