In-Context Alignment at Scale: When More is Less

Neelabh Madan [1] **Lakshminarayanan Subramanian** [1]

Abstract

In-context instructions are a widely used and accessible method for aligning model behavior through human feedback. However, as users increasingly expect LLMs to perform multiple tasks or exhibit diverse behaviors, the number of such instructions in the prompt scales rapidly. In this work, we investigate how LLMs *scale* in their ability to accurately incorporate new information or rules provided purely in context—especially when such information contradicts the model's prior beliefs or behaviors, and when the amount of such in-context information increases. We conduct experiments using controlled open-source benchmarks such as `NewNews`, which poses questions about hypothetical unseen news events, and we also introduce a synthetic benchmark that injects explicit rules into the prompt. These rules are designed to be easy to evaluate and must be followed by the model in order to generate the correct response. Our analysis reveals several key insights: (1) larger models generally perform better at incorporating new information, though their accuracy degrades as the number of new facts increases– which is expected; (2) prompt depth has limited overall effect, although in tasks involving similar rules, information placed at the beginning and end of the prompt is more reliably attended to; and (3) LLMs often "cheat" by exploiting superficial cues, and struggle when true logical inference is required—highlighting the need for more robust evaluation protocols. These findings offer critical insight into the current limitations of in-context behavior alignment in LLMs at scale.

[1]New York University. Correspondence to: Neelabh Madan <nm3171@nyu.edu>.

ICML 2025 Workshop on Models of Human Feedback for AI Alignment, Vancouver, Canada.

1. Introduction

Large Language Models (LLMs) have emerged as powerful tools for natural language understanding and generation, capable of tackling a wide range of tasks in zero-shot and few-shot settings. Their success is attributed to extensive pretraining on massive corpora, resulting in models that encode vast world knowledge and general-purpose reasoning abilities. However, training such models requires significant computational resources, making it infeasible to fine-tune or retrain them for every user or application or alignment towards a particular new behavior. As a result, in order to align the model to human preferences/feedbacks there is growing reliance on *in-context learning*—where models are expected to adapt to new tasks or domains using only the information provided at inference time. This paradigm of in-context alignment (ICA) is widely used as it is the most accessible method for all users. ICA is particularly appealing in real-world automation scenarios, where users employ LLMs as general-purpose agents under a variety of constraints, such as "be concise," "use this new knowledge," "follow this format," or "do not use bullet points." With the advent of models capable of handling long contexts—reaching up to 1 million tokens—there is a strong temptation to "just throw everything into the prompt" and expect the model to seamlessly incorporate all relevant rules, preferences, and information. This observation raises a fundamental question: **Does the ability to provide extensive context necessarily imply that it is beneficial to do so?**

In practice, users often find that models fail to honor constraints or incorporate new facts as intended, especially in settings involving synthetic data generation or rule-based data annotation. Designing prompts that reliably elicit the desired behavior frequently requires extensive iteration, revealing underlying fragilities in how LLMs interpret and internalize in-context information. These failures suggest a gap between our mental models of LLMs and their actual behavior—one that mirrors phenomenon studied in cognitive psychology, such as recency effects and memory interference.

In this work, we investigate the extent to which LLMs can *revise or augment their beliefs* using new information provided solely in the prompt, particularly when this information introduces novel concepts or contradicts prior model

knowledge and when this information is given at scale. We refer to this ability as **in-context belief updating/alignment**, and we study its limits through controlled synthetic benchmarks. Each prompt contains a set of never seen before statements (defining new operations, facts, or concepts) or explicit instructions, followed by questions that can only be answered correctly if the model makes use of these injected facts/instructions.

By varying the *number of new facts/instructions*, the *degree of difficulty*, *the depth* of the new fact in the corpus and the *model size*, we uncover how performance scales with prompt complexity. We measure not only accuracy, but also failure modes such as shallow pattern matching. Our findings enable us to observe of empirical trends that characterize how well LLMs update beliefs and follow injected rules as a function of model capacity and contextual load.

Ultimately, our goal is to build a more principled understanding of how LLMs process and prioritize new contextual information. This has direct implications for improving prompt design, enhancing model interpretability, and guiding the development of more robust and adaptive language agents. Our paper presents the following novel findings:

- **Performance Degradation with Context Complexity:** LLMs show performance drops as the number of behaviors (N) increases, with up to 50% accuracy loss from $N = 1$ to $N = 2$, even in large models.
- **Task-Specific Depth Effects:** Model performance varies with the depth of key information, showing that depth effects are task-dependent and contradict trends from long-context benchmarks.
- **Emergence of Cheating Behaviors:** Models relying on shallow pattern-matching degrade substantially when substitutions disrupt cues, highlighting limits in generalization and reasoning.
- **Independent Task Complexity Affects Performance:** Increasing the complexity of the task which require additional skills tend to degrade the performance more as the number of behaviors to align for increases.

2. Related Work

We place our work at the intersection of language model in-context alignment and scaling. Our focus is to understand how well large language models internalize and utilize new information provided at scale during inference, particularly when it conflicts with their prior knowledge.

2.1. Scaling Laws in Language Models

Early work by Kaplan et al. (2020) established empirical scaling laws that relate model performance to parameter count, dataset size, and compute. This was later refined by Hoffmann et al. (2022), who emphasized compute-optimal training under the Chinchilla paradigm. While these studies focus on perplexity and task accuracy under i.i.d. conditions, our work investigates how the capacity for online belief updating scales with model size and contextual load. Moreover, these studies inspire our research to provide empirical scaling trends for in-context alignment.

2.2. Belief Modeling and Revision

LLMs are known to internalize factual and commonsense beliefs during pretraining. Lin et al. (2022) showed that models often reproduce plausible but incorrect statements, highlighting the challenge of belief misalignment. The ELK framework proposes evaluating models' latent beliefs beyond their surface outputs. On the editing front, approaches like ROME (Meng et al., 2023), MEMIT (Mitchell et al., 2022), and others (Dai et al., 2022) aim to modify internal representations directly, requiring weight updates. In contrast, our method provides new facts in context and studies how this affects downstream reasoning—without model modification.

2.3. In-Context Learning and Prompt Adaptation

In-context learning (ICL) emerged with GPT-3 (Brown et al., 2020), allowing models to adapt to new tasks via prompting. However, recent work shows that ICL is often brittle and relies heavily on surface-level heuristics (Min et al., 2022). Prompt tuning methods (Lester et al., 2021) and instruction-based adaptation aim to make such behavior more robust. Our study builds on the lines of (Park et al., 2025) by asking: how do models behave when the prompt explicitly contradicts their prior knowledge? We explore this as a new axis of generalization in ICL. We distinguish ourselves by providing a *scaling* study for in-context information/instructions as opposed to previous works where only a single novel information/instruction is provided to the LLMs for inference on downstream task.

2.4. Memory, Interference, and Contextual Limits

Recent studies highlight the limits of contextual understanding in long prompts. Liu et al. (2023) show that models often ignore information placed in the middle of long contexts, while Khandelwal et al. (2020) suggest that memorization and nearest-neighbor retrieval influence reasoning. Prompt interference has been studied by Zhao et al. (2021), who found that formatting and context length can significantly affect performance. Our empirical study follow a similar taste and reveals how such interference plays a role in belief updating failure as the number of injected facts grows.

2.5. Dynamic Evaluation and Continual Adaptation

Traditional dynamic evaluation methods (Krause et al., 2017) and newer test-time training approaches (Sun et al., 2020) adapt models to changing inputs, often with gradient updates or additional memory components. However, such methods are not directly applicable in zero-shot or inference-only settings. We consider a constrained version of this challenge: can LLMs adapt their output behavior based purely on contextual input in a single forward pass?

3. Setup

Our goal is to evaluate how well large language models (LLMs) can incorporate and align to *new information* that is **not inherently learned during pretraining** and largely focus on how do they perform when such instances of new information or behaviors increase to a scale of 100s of occurances of new information. Specifically, we aim to assess a model's ability to follow novel rules or perform unseen tasks solely based on contextually injected instructions (in the prompt itself), without any parameter updates. This includes both newly introduced symbolic concepts and behavioral constraints expressed as rules. If a model truly possesses the capabilities required to complete a task based on a large corpus of instructions, rules, or contextual information, then it should be able to follow such instructions—even when they are novel in form or content.

To formalize this setting, consider a pretrained model $\mathcal{M}$ which, when given an input string x, produces an output $y = \mathcal{M}(x)$. This represents the standard inference setup for a LLM. However, we additionally consider a *behavior set* $\mathcal{B}$, which specifies how the model is expected to behave. The set $\mathcal{B}$ can include novel instructions or information required to solve a given task t.

For example, a behavior $b \in \mathcal{B}$ might be an informational statement such as "Toyota has re-released an old car, the Supra, in the market," or a rule such as "Answer in bullet form." The corresponding task t could be a query like "Can I purchase the Supra today?"

It may be the case that solving t requires reasoning over multiple behavioral elements $b \in \mathcal{B}$. However, for the purposes of this analysis, we restrict ourselves to a simplified setting in which exactly one behavior, denoted $b^* \in \mathcal{B}$, is sufficient to solve the task t. This simplification helps isolate and study the impact of individual behavioral constraints.

In real-world scenarios, the behavior set $\mathcal{B}$ may be predefined or updated dynamically in a modular fashion, independent of the model $\mathcal{M}$. Given the model's capacity to handle a wide distribution of tasks, a natural and practical approach (used by many users) is to provide the full behavior set $\mathcal{B}$ within the prompt and allow the model to infer which behavior is relevant to the current task t. This is implemented by constructing a prompt $x = p(\mathcal{B}, t)$ via a function p: which arranges the behavior set,$\mathcal{B}$ and the task t into a single prompt, and performing inference via $y = \mathcal{M}(x)$.

Let $n = |\mathcal{B}|$ denote the number of behavioral elements, rules, or facts provided. When presenting $\mathcal{B}$ within a prompt, some form of sequential indexing must be imposed on the set. Thus, the relevant behavior b^* may appear at any index $i \in 1, 2, \ldots, n$. We define the ***depth*** d of a task t as the relative position of b^* within the list, expressed as a percentage: $d = \frac{i}{n} \cdot 100$.

From this point onward, our focus is on understanding how in-context task-solving strategies are affected by both n (the number of behaviors) and d (the depth of the relevant behavior), across various model sizes and task types. We also aim to identify failure cases and shortcut behaviors (bypassing), which help the community gain intuition for designing effective behavioral prompts—especially in large-scale automation scenarios involving LLMs, such as data generation pipelines.

To conform to the described setup, we utilize two datasets: (i) an open-source dataset, `NewNews` (Park et al., 2025), and (ii) a synthetic dataset constructed by us, referred to as `ADD-THEN-RULES`. The former enables evaluation of models on realistic, contextually injected information, while the latter facilitates controlled experimentation with explicit behavioral constraints and rules. Both datasets are provide novel information that must be reasoned over to answer corresponding questions accurately. We run all experiments using the `Qwen-2.5B-Instruct` model family.

3.1. The `NewNews` Dataset

The `NewNews` dataset consists of novel concepts framed as fictional news items, each paired with five downstream tasks. Each task requires the model to comprehend the news in order to answer a single-correct multiple-choice question with four options. The dataset contains a total of 75 unique news items. An example of a news-question-options triplet is shown below:

> **News:** Mathematicians define ''addiplication'' of x and y as $(x + y) \cdot y$.
> **Ques:** What is the addiplication of 3 and 4?
> **Options:** [28, 7, 12, 0]

These entries are categorized into five thematic domains—`mathematics`, `coding`, `discoveries`, `leaderboards`, and `events`—to enable fine-grained analysis of reasoning behavior across different content

types.

In real-world scenarios, users often provide multiple instructions or informational elements within a single prompt. To emulate this, we inject multiple news items into the context and evaluate whether the model can correctly identify and use the relevant one. We denote this evaluation setting as **MULTI-NewNews**.

To align this with our setup, for each news-question-options triplet (n_i, q_i, o_i) from the NewNews dataset, we construct a behavior set $\mathcal{B}$ such that $b^* = n_i$ and the remaining $N-1$ behaviors are randomly sampled from the other 74 news entries in the dataset. The task t is set as q_i, and inference is performed using the formulation $y = \mathcal{M}(p(\mathcal{B}, q_i))$. [1]

3.1.1. THE ADD-THEN-RULES DATASET

We construct a synthetic, rule-following mathematics dataset named ADD--THEN--RULES. This dataset is designed to evaluate model performance on tasks where the behaviors $b \in \mathcal{B}$ explicitly enforce output constraints . Such rule-following is a highly common use case, both in industry (e.g., for synthetic data generation, downstream LLM pipelining, or LLM evaluation workflows) and among everyday users who often prompt models in an *if-this-then-that* manner.

In ADD-THEN-RULES, the core task is to add two numbers a and b, and determine whether their sum falls within a specific range defined by an associated rule. If the condition specified by the rule is satisfied, the output must be modified accordingly, thus testing the model's ability to conditionally transform responses based on behavioral constraints.

For example, if $a = 3$, $b = 7$, and the rule r_1 is:

```
If the answer is in the range [9--19],
then change the answer to "kangaroo",
```

then since $a + b = 10 \in [9, 19]$, the output should be replaced with "kangaroo".

We randomly sample 100 such (a, b) pairs for each pair of digit-lengths of a and b, where the number of digits in a and b vary from 1 to 10. This results in a total of $10 \times 10 \times 100 = 10{,}000$ sample questions.

For each (a, b) pair, we define one *active* rule, denoted as r_a, which directly alters the correct sum. Additionally, we introduce $N-1$ *distractor* rules that do not apply to the sum and thus should not alter the output. Each rule r_i is defined in the following format:

r_i: `If the answer is in the range` $[a_i, b_i]$, `return` c_i `instead of the answer,`

where c_i is a randomly sampled ImageNet class. By design, distractor rules ensure that the sum $a+b$ does *not* fall within the specified range $[a_i, b_i]$.

In this setup, the behavior set is defined as $\mathcal{B} = \{r_a\} \cup \{r_i\}_{i=1}^{N-1}$, where r_a denotes the *active* rule required to solve the task, and $\{r_i\}_{i=1}^{N-1}$ are $N-1$ randomly sampled *distractor* rules. The task is defined as $t =$ `Give me the answer to` $a + b$

Synthetic tasks are particularly valuable due to their scalability and fine-grained control, enabling us to explore edge cases and failure modes that would be difficult to isolate in natural data.

4. Experiments

Our experiments are designed to investigate the following core questions:

- How does model performance change as we increase the number of novel behaviors, i.e., as $N = |\mathcal{B}|$ increases?
- Does the *depth* at which the relevant behavior b^* is injected into the context affect performance?
- What failure modes emerge as contextual load increases—such as shortcutting, pattern matching, or ignoring constraints?
- Are there observable **scaling trends** that inform our understanding of instruction-based prompting and behavioral generalization?

We conduct our evaluation using the Qwen2.5-Instruct family of models, ranging in size from 0.5B to 14B parameters using the vLLM framework (Kwon et al., 2023). This size range reflects a practical subset of models widely used in real-world academic-level automation workflows, as they can be efficiently deployed on a single A100 GPU. By focusing on this range, we ensure that our analysis captures the behavior of models that are accessible to a broad base of researchers and developers.

4.1. NewNews Experiments

We vary the size of the behavior set as $N = |\mathcal{B}| \in \{5, 10, 20, 50, 75\}$ and the depth of the relevant behavior b^* as a percentage $d \in \{0, 20, 40, 60, 80, 100\}$.

This setup allows us to investigate how the number of new beliefs (i.e., conceptual definitions) and the depth d of b^* within $\mathcal{B}$ influence the model's ability to recall and reason

[1] Please find the various prompt functions p used in the appendix

over the appropriate information when answering follow-up questions.

Inference is performed using the following parameters: `temperature = 0.8, top_p = 0.95`. For each prompt, denoted as $x = p(\mathcal{B}, q_i)$, we generate 10 outputs. The prompt p instructs the model to perform Chain-of-Thought (CoT) reasoning (Wei et al., 2023), typically using phrases such as *"think step by step"*. Each generated output $y = \mathcal{M}(x)$ is post-processed to extract the final answer, which is then evaluated against the ground truth option (A, B, C, or D). We report both `pass@10` and the average number of correct generations per sample x, which corresponds to the expected `pass@1`.[2]

Most questions and answer options in the `NewNews` dataset contain keywords that lexically overlap with the news passages required to answer them. This enables shallow matching or lookup-based answering, limiting the expressiveness of the benchmark in evaluating a model's true ability to compose skills and retrieve relevant facts at scale (e.g., $N \sim 50$). In some cases, even the correct answer options share high lexical overlap with the corresponding news snippet.

To disentangle these effects—particularly cases where the model might exploit shortcut patterns—we created several variants in addition to the original `MULTI-NewNews` benchmark (Section 3.1).

To remove assistance from the answer *options*, we introduce two variants. The first, **`+NO-MCQ`**, reformulates the task as free-form generation by eliminating all multiple-choice options. The second, **`+NOTA-MCQ`**, replaces the correct answer with a "None of the above" (**NOTA**) option, which is always the correct answer in this version. To reduce keyword overlap in the *questions*, we introduce the **`+SUBSTITUTE`** transformation. This variant replaces key entities or phrases in the question that also appear in the news passage with substitutions based on common knowledge, thereby reducing lexical cues that models could exploit. Eg:

> **`News:`** `In 2025, the Vatican has ordained women as priests for the first time in history.`
> **`Question:`** `Has the Catholic Church ever ordained women as priests before?`
> **`Substituted Question:`** `Has the religious institution led by the Pope ever granted clerical roles to females before?`

Together, these six configurations form a comprehensive suite for evaluating the robustness and reasoning capabilities of language models without relying on lexical shortcuts: **`MULTI-NewNews`**, **`+NO-MCQ`**, **`+NOTA-MCQ`**, **`+SUBSTITUTE`**, **`+SUBSTITUTE+NO-MCQ`**, **`+SUBSTITUTE+NOTA-MCQ`**.

[2] `pass@k` refers to generating k outputs for a given sample and counting the evaluation as correct if at least one of them matches the ground truth.

4.2. `ADD-THEN-RULES` Experiments

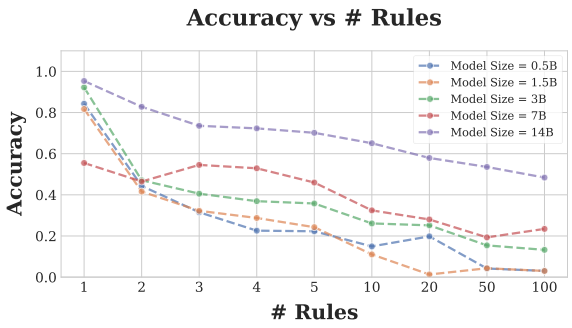


Figure 1: We can observe that as we increase the number of rules for the model to follow, the accuracy of performing the `ADD-THEN-RULES` task goes down drastically. Even at $N = 2$, we see a substantial drop in performance

To evaluate how the number of rules affects performance, we construct multiple datasets with varying numbers of rules:

$$N = |\mathcal{B}| \in \{1, 2, 3, 4, 5, 10, 20, 50, 100\}.$$

For each instance, the actual rule r_a is placed randomly among the *distractor* rules, potentially at any position in the list, thereby varying the depth at which the relevant information appears.

We conduct a fine-grained analysis to assess how much of the model's accuracy stems from cases where the correct answer is explicitly present in the numerical range specified by a rule—that is, when the answer matches one of the bounds a_j or b_j in some rule. We refer to these instances as `On-Edge` cases, which account for approximately 40% of the dataset. The remaining 60% of examples, where the correct answer does not appear in any of the rule ranges, are referred to as `Not On-Edge` cases.

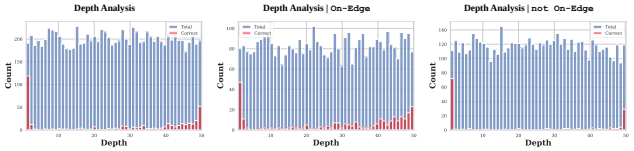
All inferences made here are in a greedy decoding fashion `temperature = 0.0` so that the results are reproducible.

5. Results

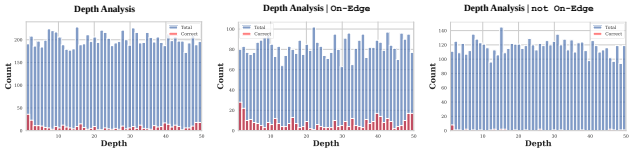
5.1. Effect of $N = |\mathcal{B}|$

We observe in both the `NewNews` and `ADD-THEN-RULES` settings that as the number of news items or rules increases, model performance consistently degrades. While the performance drop in the `NewNews` task (Figure 3) is relatively modest, the degradation in `ADD-THEN-RULES` (Figure 1) is much more severe. For instance, increasing from $N = 1$ to $N = 2$ leads to a nearly 50% drop in accuracy for even the decently large models ranging from 1.5B to 7B parameters.

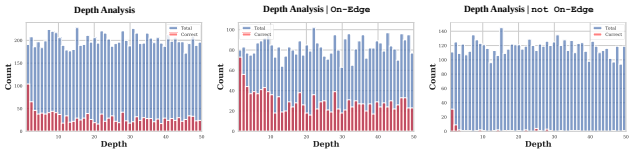
Interestingly, the **`+SUBSTITUTE`** variant shows significant degradation as N increases. This suggests that when pattern matching is insufficient and reasoning over transformed



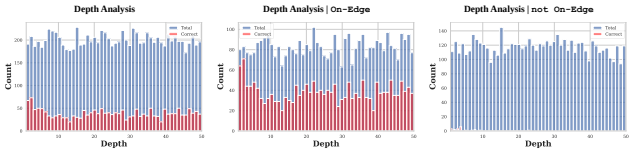
(a) 0.5B Qwen2.5 model



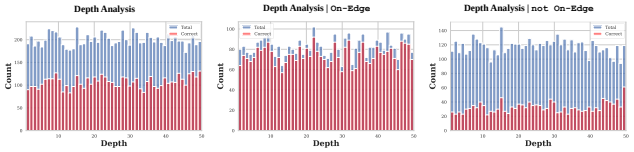
(b) 1.5B Qwen2.5 model



(c) 3B Qwen2.5 model



(d) 7B Qwen2.5 model



(e) 14B Qwen2.5 model

Figure 2: Model performance comparisons across depths of *actual* rule, r_a for model different sizes at $N = 50$

content is required, increased contextual load substantially hurts performance. However, larger models like the 14B variant remain relatively stable under such conditions.

It is interesting to note that current benchmarks may overestimate the model's true generalization and compositional capabilities when evaluated under single-behavior ($N = 1$) settings. Robust assessment should account for performance under increasing behavioral or informational complexity.

5.2. Effect of Depth on performance

For the `MULTI-NewNews` dataset (Figure 1), the effects of the depth of placement of b^* are non-trivial, depending on the task at hand. While smaller models such as 0.5B exhibit greater sensitivity to depth, they are inherently less powerful and struggle even with following the structured output format. In contrast, larger models like 7B and 14B demonstrate relatively stable performance across all depths. Mid-sized models, such as 1.5B and 3B, show stronger trends.

Looking at the NewNews variations, we observe that for the original task design (`MULTI-NewNews` and `+SUBSTITUTE`), model performance decreases as we increase the depth. This contrasts with the trends reported in the long-context literature, where performance typically peaks at the beginning and the end of the depth range.

When we remove the assistance provided by the MCQ options and introduce a distractor rule, we observe a somewhat opposite trend. In this case, the placement of b^* at the end consistently yields better performance across all model sizes, although the effect is not extremely large. However, removing the MCQ options results in a substantial drop in performance—approximately 2 or more additional incorrect predictions per 10 generations.

For free-text generation tasks without any MCQ options, we again observe a different trend. However, this aligns with the long-context literature, where placing b^* at the beginning and the end leads to improved performance.

These observations allow us to draw the following conclusions:

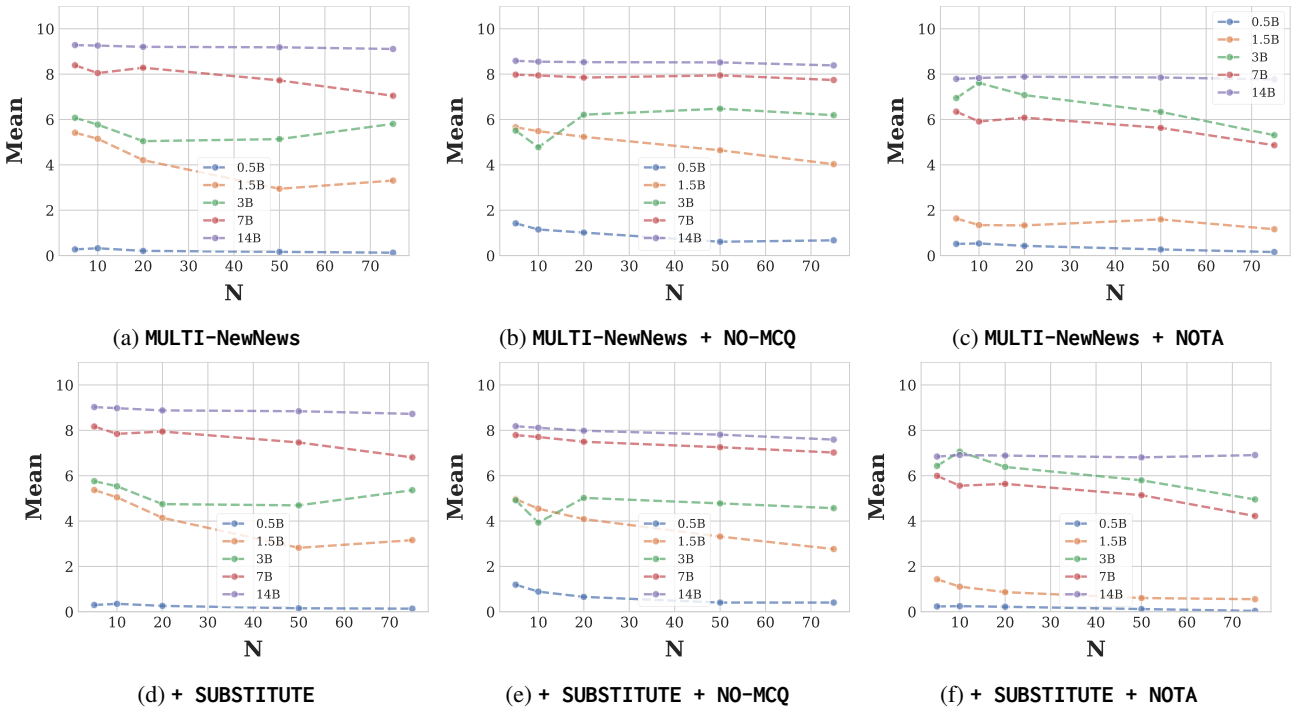
- When there is significant cognitive load in answering the question—such as determining the `NOTA` (None of the Above) option rather than relying on simple pattern matching—models tend to perform better if b^* is placed nearby (i.e. towards the end).
- Free-form text generation, which is a better proxy for the model's true capabilities, follows the traditional trends observed in the long-context literature.

For the `ADD-THEN-RULES` dataset (first column of Figure 2), we observe that smaller models (0.5B and 1.5B) perform better when b^* is placed at the initial or final depths, whereas the 3B and 7B models tend to perform better when b^* is placed at the start. Only the 14B model consistently performs well across all depths, with a slight inclination towards optimal performance at the 100% depth position. .

5.3. Cheating Behaviors

Often, we correlate a model's ability to solve problems with its performance on a dataset, and then generalize this ability to other tasks. However, it is important to note that if experiments are not carefully designed, they can enable the model to "game" answers, providing a false sense of capability. We observe such behavior in the `MULTI-NewNews` and `ADD-THEN-RULES` datasets. This analysis highlights how the presentation of data to the model can affect its performance on downstream tasks.

From Table 1, we see that if we remove the cheating signals by augmenting the dataset `+NOTA` or `+SUBSTITUTE`



(a) MULTI-NewNews (b) MULTI-NewNews + NO-MCQ (c) MULTI-NewNews + NOTA

(d) + SUBSTITUTE (e) + SUBSTITUTE + NO-MCQ (f) + SUBSTITUTE + NOTA

Figure 3: Mean (average correct per 10 generations) per sample vs N for various models sizes across the six augmentations created from the NewNews dataset

then not only do we observe the performance go down for even N=1 tasks, but also the sensitivity of degradation increases as N increases. See the drop in N=50 or N=75 for MULTI-NewNews vs MULTI-NewNews+NOTA+SUBSTITUTE. As expected, smaller models are more prone to this behavior.

In the ADD-THEN-RULES dataset (Figure 2), this behavior is particularly prominent. Models with fewer than 14B parameters are unable to correctly handle any of the Not On-Edge cases. This suggests that the models are not truly performing the task as intended but are instead leveraging cues from the rules to guess the answers. Alternatively, this might indicate that the combination of skills required—first to compute the correct answer, and then to verify its membership within a set—is not additive or straightforward for these models.

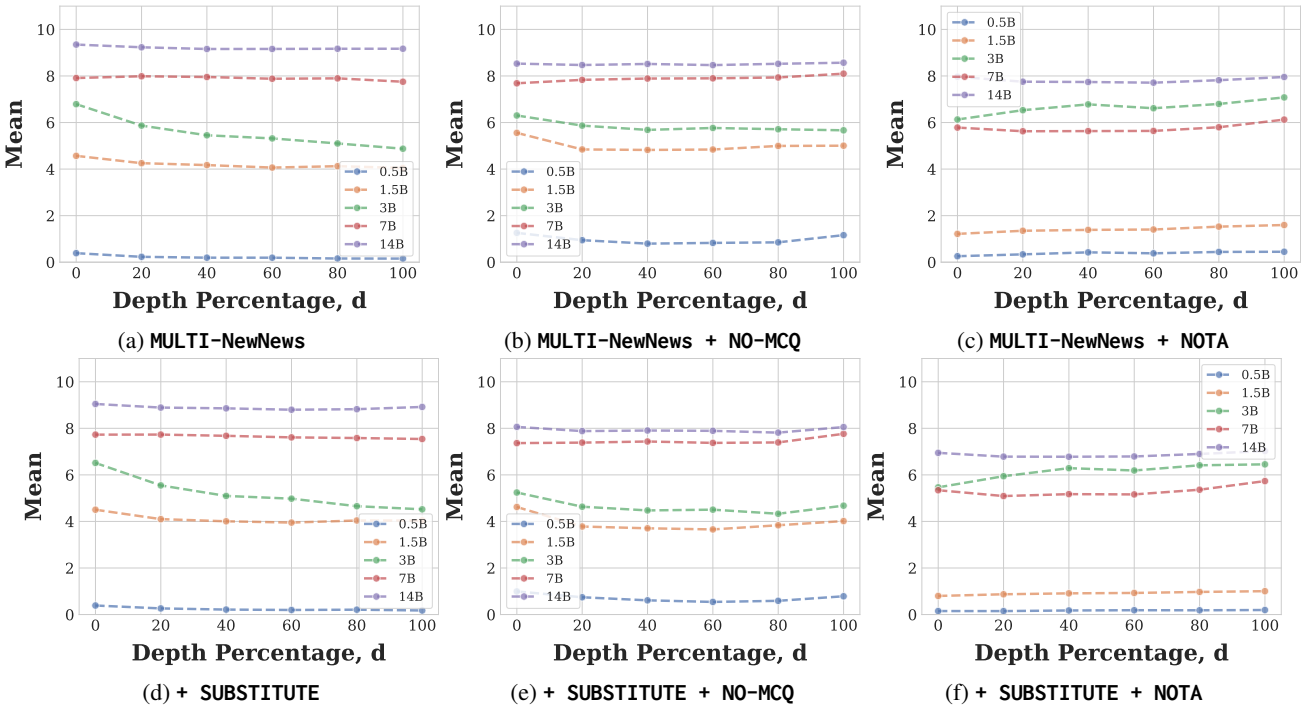
Overall, the accuracy of the addition task is fairly impressive, indicating that the primary performance drop comes from the subsequent application of the membership-checking.

5.4. Requirement of Additional Skills

We also investigate how the model's behavior changes when it is tasked with solving more challenging problems that require multiple reasoning steps. The underlying assumption is that if a problem necessitates additional skills, it is inherently more difficult than questions that merely require

N	Model				
	0.5B	**1.5B**	**3B**	**7B**	**14B**
	MULTI-NewNews				
N=1	**0.1**	**6.4**	**6.9**	**7.8**	**9.2**
N=50	9.2	52.6	22.1	1.3	-1.1
N=75	36.4	46.1	11.9	11.3	-0.6
	MULTI-NewNews + SUBSTITUTE				
N=1	**0.2**	**6.2**	**6.7**	**8.0**	**9.3**
N=50	20.0	53.3	28.3	4.5	2.0
N=75	32.4	47.9	17.2	14.6	3.5
	MULTI-NewNews + NO-MCQ				
N=1	**1.4**	**7.0**	**5.3**	**8.5**	**8.6**
N=50	57.5	29.4	-29.3	3.5	-2.2
N=75	54.5	38.8	-21.5	6.0	-0.2
	MULTI-NewNews + NO-MCQ + SUBSTITUTE				
N=1	**1.3**	**6.4**	**5.0**	**8.4**	**8.5**
N=50	72.2	46.4	3.8	11.0	5.0
N=75	72.4	55.9	8.7	13.8	8.2
	MULTI-NewNews + NOTA				
N=1	**0.1**	**1.5**	**5.9**	**6.5**	**7.4**
N=50	-51.4	-7.8	-8.9	10.5	-4.8
N=75	9.3	23.3	12.1	26.1	-3.1
	MULTI-NewNews + NOTA + SUBSTITUTE				
N=1	**0.1**	**1.7**	**6.0**	**6.3**	**6.8**
N=50	36.4	65.4	3.3	18.1	0.0
N=75	79.8	67.8	18.7	34.1	-0.2

Table 1: % relative decrease in the mean number of correct prediction per 10 generations for $N = 50, 75$ as compared to $N = 1$ for the NewNews dataset and it's variations



(a) MULTI-NewNews (b) MULTI-NewNews + NO-MCQ (c) MULTI-NewNews + NOTA

(d) + SUBSTITUTE (e) + SUBSTITUTE + NO-MCQ (f) + SUBSTITUTE + NOTA

Figure 4: Mean (average numer of correct predictions per 10 generations) comparisons across depths of *actual* rule, b^*, for different model sizes for the various augmentations created from the NewNews dataset

simple lookup inference from the News.

To evaluate this, we manually annotated examples that require additional skills and assessed the model's performance specifically on these examples. We observe a significant performance gap for questions that demand these additional reasoning abilities (see Table 2).

This finding is particularly interesting, as the difficulty of solving a problem should ideally be independent of retrieving the correct b^* from the corpus. However, we observe that the degradation in performance when increasing N from 1 to 50 or 75 is significantly greater in scenarios where additional cognitive load is required—that is, when the problem demands additional skills. This suggests that current benchmarks that evaluate a skill based on a single piece of information do not scale well with an increase in the number of alignment behaviors.

6. Conclusion

With the help of this paper, we aim to present our extensive findings on how large language models (LLMs) handle a wide range of behaviors, particularly novel ones. We examine how LLMs behave under these conditions so that the community can be more aware of the potential pitfalls of current inference strategies—especially when multiple instructions are given to LLMs simply because they support a larger context length.

N	Model Size				
	0.5B	**1.5B**	**3B**	**7B**	**14B**
Additional Skills Not Required (240 samples)					
N=1	22.9 (0.2)	99.1 (7.0)	99.1 (7.5)	96.6 (8.1)	99.5 (9.5)
N=5	25.9 (0.3)	99.3 (6.3)	97.0 (7.0)	98.4 (8.7)	99.6 (9.6)
N=10	28.6 (0.3)	99.5 (6.1)	96.0 (6.6)	98.1 (8.3)	99.9 (9.6)
N=20	19.8 (0.2)	98.8 (5.0)	95.8 (5.8)	98.2 (8.6)	99.6 (9.6)
N=50	16.3 (0.1)	94.6 (3.5)	96.3 (5.8)	97.5 (7.9)	99.6 (9.6)
N=75	13.1 (0.1)	98.1 (3.9)	97.1 (6.6)	97.7 (6.9)	99.7 (9.6)
Additional Skills Required (135 samples)					
N=1	16.2 (0.1)	94.0 (3.9)	94.8 (5.1)	91.8 (7.8)	97.0 (8.7)
N=5	17.2 (0.2)	93.5 (3.7)	89.1 (4.4)	92.5 (7.7)	95.5 (8.6)
N=10	19.9 (0.2)	93.2 (3.3)	84.8 (4.1)	92.4 (7.5)	94.8 (8.4)
N=20	14.4 (0.1)	89.0 (2.7)	84.3 (3.5)	93.7 (7.5)	95.9 (8.4)
N=50	11.4 (0.1)	79.8 (1.9)	88.6 (3.8)	93.9 (7.3)	94.8 (8.3)
N=75	08.6 (0.0)	82.7 (2.1)	87.6 (4.2)	94.3 (7.2)	94.9 (8.1)

Table 2: % pass@10 (average correct predictions per 10 generations) for the MULTI-NewNews dataset. The degradation in performance as N increases is notably more severe for cases requiring additional skills. Note that in this setting, we have not filtered any samples, and therefore the model benefits from helpful signals present in both the questions and the options. While previous results indicated that large models (14B) remain relatively stable as N increases, in this setting—where additional skills are required—even the large model exhibits a significant performance degradation, with a relative decrease exceeding 10%.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning, specifically in understanding and evaluating the ability of language models to handle multi-step reasoning and alignment behavior under complex settings and scaled up behaviour provided in context. While the techniques and findings reported here are intended to improve the robustness and interpretability of machine learning models, we recognize potential risks associated with their misuse or unintended consequences, such as reinforcing biases or misinterpretation of model capabilities in real-world applications. However, we believe that the broader implications align with advancing safe and transparent AI systems, and there are no immediate or specific societal or ethical concerns that require further emphasis at this time.

References

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., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B., and Wei, F. Knowledge neurons in pretrained transformers, 2022. URL https://arxiv.org/abs/2104.08696.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., and Lewis, M. Generalization through memorization: Nearest neighbor language models, 2020. URL https://arxiv.org/abs/1911.00172.

Krause, B., Kahembwe, E., Murray, I., and Renals, S. Dynamic evaluation of neural sequence models, 2017. URL https://arxiv.org/abs/1709.07432.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning, 2021. URL https://arxiv.org/abs/2104.08691.

Lin, S., Hilton, J., and Evans, O. Truthfulqa: Measuring how models mimic human falsehoods, 2022. URL https://arxiv.org/abs/2109.07958.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. Lost in the middle: How language models use long contexts, 2023. URL https://arxiv.org/abs/2307.03172.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in gpt, 2023. URL https://arxiv.org/abs/2202.05262.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work?, 2022. URL https://arxiv.org/abs/2202.12837.

Mitchell, E., Lin, C., Bosselut, A., Finn, C., and Manning, C. D. Fast model editing at scale, 2022. URL https://arxiv.org/abs/2110.11309.

Park, C. F., Zhang, Z., and Tanaka, H. *New News*: System-2 fine-tuning for robust integration of new knowledge, 2025. URL https://arxiv.org/abs/2505.01812.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A. A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts, 2020. URL https://arxiv.org/abs/1909.13231.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Zhao, T. Z., Wallace, E., Feng, S., Klein, D., and Singh, S. Calibrate before use: Improving few-shot performance of language models, 2021. URL https://arxiv.org/abs/2102.09690.

A. Prompts used

A.1. MULTI-NewNews

Prompt

```
<|im_start|>system
You are a helpful assistant. You are given a set of news and you have to give the answer to the
question given in the <question></question> tags and give the final answer in between the
<answer>....</answers> tags taking the relevant news in consideration. You will be given 4 options
A, B, C, D in the <options></option> tags. You have to select the correct option and give the answer
option (either A, or B or C or D) in the <answer>....</answer> tags.<|im_end|>
<|im_start|>user
Here are the news
News 1: <News1>
News 2: <News2>

Please answer the question
<question>
<Question>
</question>

<options>
A: <Option A>
B: <Option B>
C: <Option C>
D: <Option D>
</options>

<|im_end|>
<|im_start|>assistant
Let me solve this step by step.
<think>
```

A.2. MULTI-NewNews + NOTA

Prompt

```
<|im_start|>system
You are a helpful assistant. You are given a set of news and you have to give the answer to the
question given in the <question></question> tags and give the final answer in between the
<answer>....</answers> tags taking the relevant news in consideration. You will be given 4 options
A, B, C, D in the <options></option> tags. You have to select the correct option and give the answer
option (either A or B or C or D) in the <answer>....</answer> tags.<|im_end|>
<|im_start|>user
Here are the news
News 1: <News1>
News 2: <News2>

Please answer the question
<question>
<Question>
</question>

<options>
A: <Option A>
B: <Option B>
C: <Option C>
D: None of the above
</options>

<|im_end|>
<|im_start|>assistant
Let me solve this step by step.
<think>
```

A.3. MULTI-NewNews + NO-MCQ

Prompt

```
<|im_start|>system
You are a helpful assistant. You are given a set of news and you have to give the answer to the
question given in the <question></question> tags and give the final answer in between the
<answer>....</answers> tags taking the relevant news in consideration.
Give the answer in the <answer>....</answer> tags.<|im_end|>
<|im_start|>user
Here are the news
News 1: <News1>
News 2: <News2>

Please answer the question
<question>
<Question>
</question>

<|im_end|>
<|im_start|>assistant
Let me solve this step by step.
<think>
```

B. Additional Analysis on ADD-THEN-RULES Dataset

An even more intriguing observation arises when analyzing how different models perform on the overall task versus its constituent sub-tasks. We evaluate model accuracy on both the full ADD--THEN--RULES task and the individual sub-tasks—addition and rule application—across various model sizes and different numbers of rules.

As shown in Figure 5, a particularly interesting trend emerges for the case when $|R_d| = 1$. In this setting, models sometimes perform better on the full task than on the plain addition sub-task alone. This suggests that the model may be exploiting artifacts in the task or prompt—effectively "cheating" by inferring the correct answer based on superficial patterns rather than genuine rule-following or arithmetic reasoning.

B.1. Do LLMs cheat on this task?

As previously noted, there is a possibility that the language model may be "cheating"—that is, leveraging unintended shortcuts—to achieve higher performance on the overall task.

While the $|R_d| = 1$ setting provides macro-level evidence of this behavior, it is plausible that a significant portion of predictions in other settings may also rely on similar shortcut strategies. To investigate this, we explore an hypotheses that could plausibly explain such behavior: The range specified in the rule matches the correct sum exactly, making it easy for the model to associate the rule with the required output.

1. Answer-in-the-Rule Cheating

We conduct a fine-grained analysis to assess how much of the model's accuracy stems from cases where the correct answer is explicitly present in the numerical range specified by a rule—that is, when the answer matches one of the bounds a_j or b_j in some rule. We refer to these instances as On-Edge cases, which account for approximately 40% of the dataset. The remaining 60% of examples, where the correct answer does not appear in any of the rule ranges, are referred to as Not On-Edge cases.

From Figure 6, we observe that as the number of rules $|R_d|$ increases, the contribution to overall accuracy from Not On-Edge cases declines substantially—particularly for smaller models (i.e., those with fewer than 14B parameters). Interestingly, the smallest model (0.5B) still shows non-trivial contributions from Not On-Edge cases. This may suggest that the model

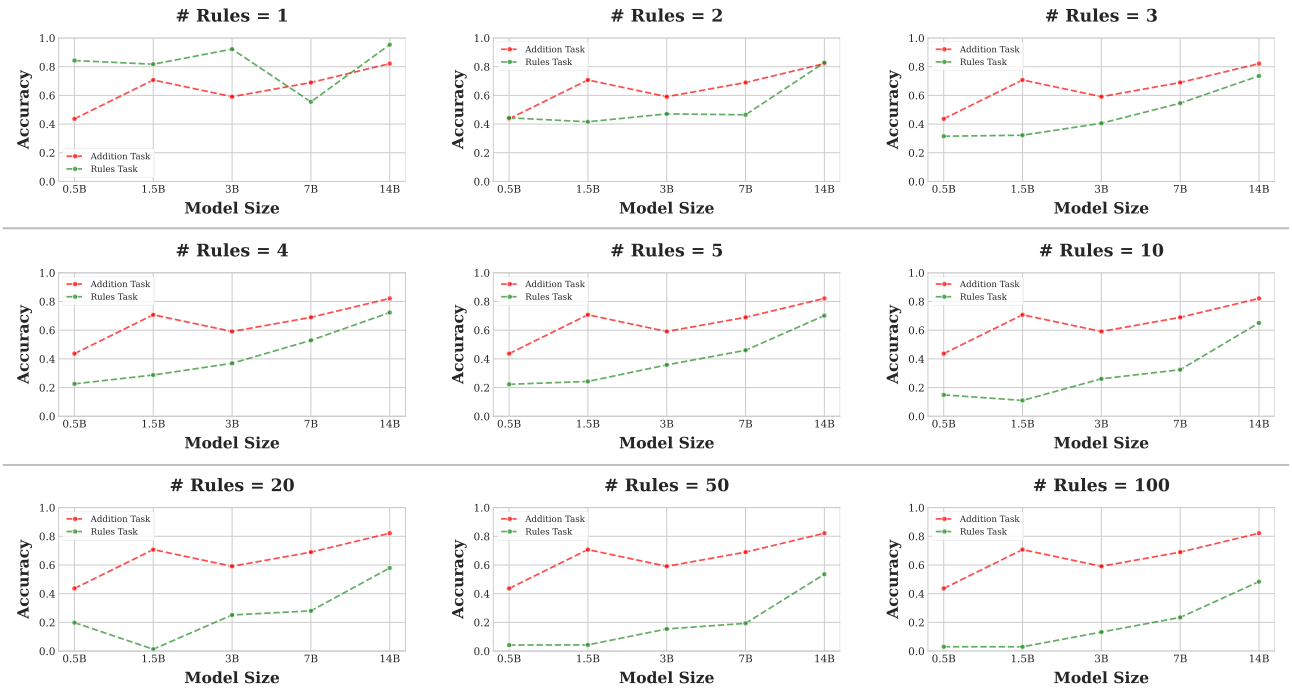


Figure 5: Fine grained into model performance as we change the number of rules. Also comparing it to the 2 subtasks of addition and comparison.

benefits simply from the presence of a rule to follow, regardless of whether the rule is actually applicable—potentially indicating superficial pattern-matching behavior rather than true rule grounding.

We also perform a recall-style analysis to evaluate how many of the total `On-Edge` and `Not On-Edge` cases are correctly answered across different models and rule set sizes.

From Figure 7, we observe that for $|R_d| = 1$, all models are able to correctly predict every `On-Edge` example. However, even a slight increase in the number of rules (e.g., from 1 to 2) leads to a noticeable drop in recall for smaller models. Interestingly, the largest model (14B) maintains high accuracy on `On-Edge` samples even as $|R_d|$ increases, suggesting greater robustness to in-context distractors.

To further evaluate the cheating hypothesis—i.e., whether models rely on the answer being explicitly present in a rule—Figure 8 offers deeper insight. If a model performs well on a majority of the `Not On-Edge` cases (i.e., cases without the answer present in any rule range), it implies genuine problem-solving capability without relying on superficial cues. Notably, at $|R_d| = 1$, nearly all models correctly answer a large fraction of `Not On-Edge` examples. However, as $|R_d|$ increases, this percentage drops drastically. In fact, some models are unable to correctly answer *any* `Not On-Edge` examples at higher rule counts.

These findings suggest three key points:

1. At $|R_d| = 1$, some models may be blindly applying a rule simply because one exists, even if it is not applicable—see our distractor experiment for supporting evidence.

2. The performance drop is largely driven by poor handling of `Not On-Edge` cases, while `On-Edge` examples disproportionately boost model accuracy.

3. The 14B model retains the ability to correctly answer a subset of `Not On-Edge` samples even at large $|R_d|$, indicating that larger models may exhibit emergent behavior: they begin to search over the rule space and apply rules more selectively.

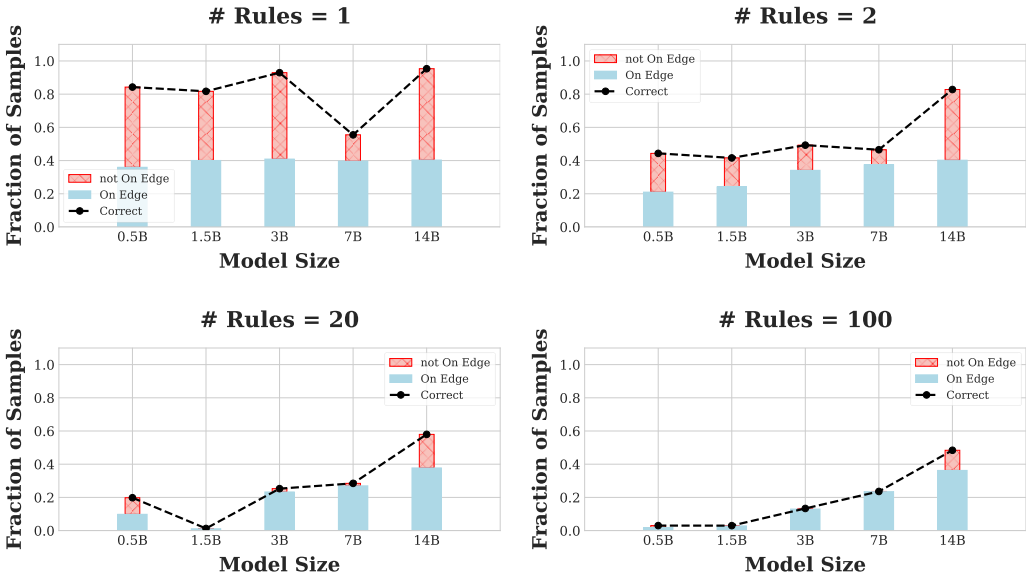


Figure 6: Graphs with fine-grained contributions of examples for which the sum $a + b$ is either a_j or b_j for the *actual* rule r_a

Refer to the depth analysis in the section for additional insights on how model accuracy varies with the depth at which the actionable rule appears in context.

B.2. Distractors with Negative Rules

To better understand model capacity for correctly identifying and ignoring irrelevant rules, we conduct a negative control experiment in which all rules provided are distractors—i.e., none of the rules trigger a change to the result of the addition task. Ideally, model accuracy under such conditions should match the original addition task accuracy, as reported in the "Add Acc." column of Table 3.

However, we observe that multiple models—particularly the smaller ones—suffer a relative accuracy drop of 40–90% in this setting. This indicates that smaller models fail to distinguish between actionable and non-actionable rules and instead attempt to follow a rule regardless of its applicability.

We hypothesize two possible explanations for this behavior:

1. There may be insufficient in-context evidence demonstrating that rules can be safely ignored. Although we do include at least one such example in the context, it may not be enough for generalization.

2. The format of the prompt for the rule-following task differs from that of the plain addition task. This prompt shift might influence the model's behavior. A potential follow-up experiment could involve providing a rule-following style prompt with zero rules and comparing the resulting accuracy to isolate the effect of prompt formatting.

	$\lvert R_d \rvert$	1	2	3	4	5	10	20	50	100	Add Acc.
Model Size	0.5B	94.39	99.29	98.92	96.61	95.46	97.94	99.82	96.61	98.26	43.64
	1.5B	48.18	45.34	52.50	70.38	63.13	66.61	84.24	82.41	55.99	70.73
	3B	58.15	24.43	-7.58	-8.26	-1.59	41.37	57.08	41.92	7.28	59.07
	7B	-17.61	-18.77	-8.98	-4.48	-9.08	20.10	88.15	74.05	43.36	68.94
	14B	11.26	15.31	13.91	16.68	17.00	22.76	21.47	25.11	32.57	82.12

Table 3: Relative performance drop (%) after adding $\lvert R_d \rvert$ distracting rules

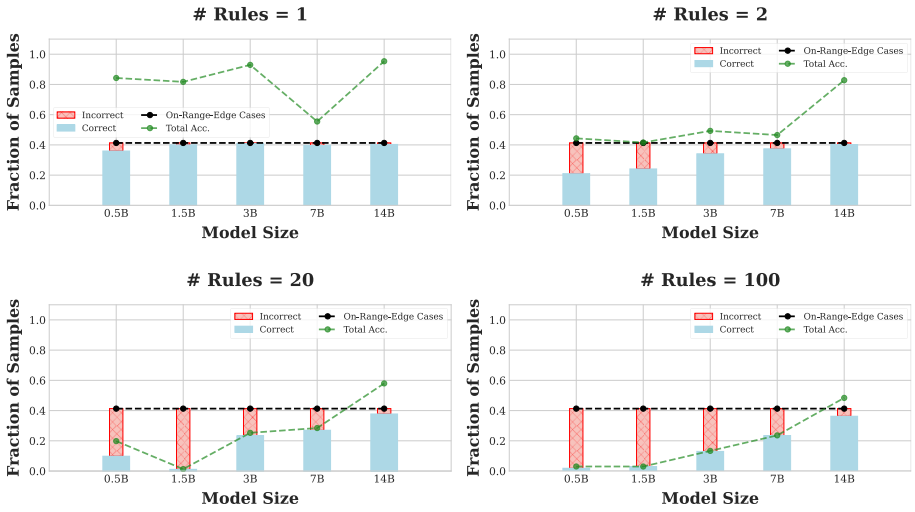


Figure 7: Distribution of correct and incorrect answers for the `On-Edge` cases ($\sim 40\%$)

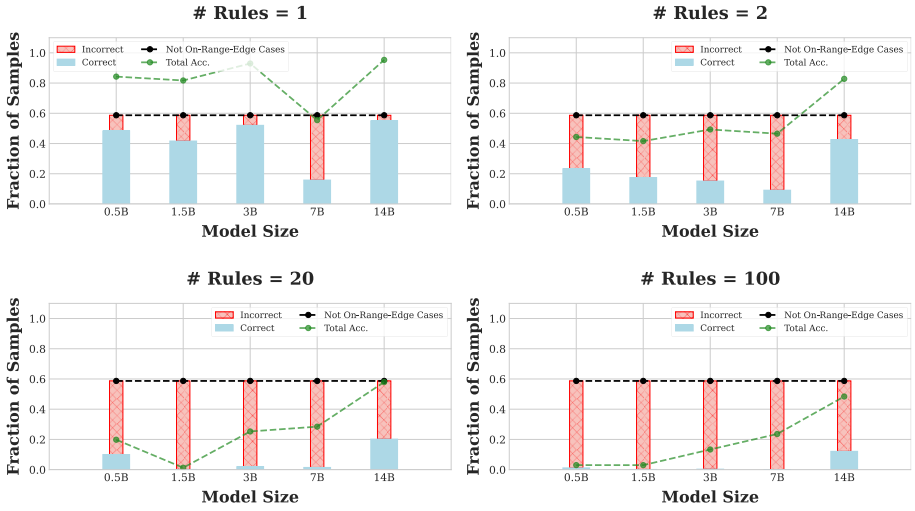


Figure 8: Distribution of correct and incorrect answers for the `not On-Edge` cases ($\sim 60\%$)

Interestingly, some green data points in Table 3 indicate an *improvement* over the original addition task accuracy. This non-trivial gain warrants further investigation. One possible reason could be formatting-related: models may avoid certain tokenization or prediction errors when operating under the rule-following prompt format. Another factor could be the consistent use of temperature $= 0$, which minimizes variability in outputs. This determinism can amplify systematic errors—if a model adopts a flawed strategy, all samples may be affected in the same way. For instance, if the model defaults to a verbose or token-intensive addition method and hits the token limit, it may fail uniformly across many inputs.

These results further emphasize the fragility of small models in settings requiring conditional logic and underscore the importance of robust prompt design and task structure.

C. Additional Information

C.1. Free Form Text Generation

For checking the answers to the free form text generation datasets like the **+NO-MCQ** and **+NO-MCQ + SUBSTITUTE**, we made use of the Qwen2.5-32B-Instruct Model

Prompt

```
<|im_start|>system
You are a helpful assistant. You are given a new news in the <news></news> tag. You are also given a
question the <question></question> tags and the ground truth answer to the question given in
<gt_answer> </gt_answer> tags. You are also given a students response in <response></response>
tags. The answer has to take into account the news in order to answer the quesion correctly. Your
job is to judge if the student has given the correct answer to this question. If you think the
sutdents answer to the question matches the ground truth answer given the news then return YES like
this <answer>YES</answer> tags else return NO like <answer>NO</answer> tags. Give reasoning before
giving the answer in the <think></think> tags.<|im_end|>
<|im_start|>user
Here are the
<news>
{news}
</news>

Here is the question
<question>
{question}
</question>

Here is the ground truth answer
<gt_answer>{gt_answer}</gt_answer>

And here is the students response
<response>{response}</response>

Please tell me if the students response is correct or not as per the instructions provided above.

<|im_end|>
<|im_start|>assistant
Let me answer this step by step.
<think>
```

C.2. +SUBSTITUTE Generations

For creating the substitute questions, we use the Qwen2.5-72B-Instruct Model to generate a question that does not contain a keyword from the news and check if the generation is valid. We generate 10 samples and select only those cases where all 10 generations are valid according to the model's self-evaluation.

The prompt used is as follows

Prompt

```
|im_start|>system
You are Qwen, created by Alibaba Cloud. You are a helpful assistant.<|im_end|>
<|im_start|>user
I have a paragraph on the basis of which we have a question. I want you to rewrite the question so
that none of the major words in the question appear in the paragraph yet the meaning of the question
remains the same. This could include substituting nouns for other common ways of representing them,
eg 'sun' can be replaced with 'The star in our solar system' and other substitutions so that
if 'sun' was appearing in the paragraph it does not appear in the question yet the meaning of the
question remains the same and hence its answer as well. In case you have to make a substitution using
the info in the paragraph itself, then return NO between the <valid></valid> tags after the
revised question generated in the <question></question> tags and the substitutions (A changed to B)
given in <substitute>A::B</substitute>format.

Here is the paragraph:
{News}

Here is the question:
{question}

Now give me the revised question, check if you
made a substitution using the paragraph itself. Think step by step:<|im_end|>
<|im_start|>assistant
Let me solve this step by step.
<think>
```

C.3. Code

We have made the code available in a github repo: https://github.com/llmrules/llm_rules.