

Null-Shot Prompting: Rethinking Prompting Large Language Models With Hallucination

Pittawat Taveekitworachai¹ and Febri Abdullah¹ and Ruck Thawonmas²

{¹Graduate School, ²College} of Information Science and Engineering

Ritsumeikan University

Ibaraki, Osaka, Japan

research@petepittawat.dev, gr0397fs@ed.ritsumei.ac.jp, ruck@is.ritsumei.ac.jp

Abstract

This paper investigates an interesting phenomenon where we observe performance increases in large language models (LLMs) when providing a prompt that causes and exploits hallucination. We propose null-shot prompting, a counter-intuitive approach where we deliberately instruct LLMs to reference a null, non-existent, section. We evaluate null-shot prompting across a variety of tasks, including arithmetic reasoning, commonsense reasoning, and reading comprehension. Notably, we observe a substantial increase in performance in arithmetic reasoning tasks for various models, with up to a 44.62% increase compared to a baseline in one model. Additional experiments on more complex mathematical problem-solving and hallucination detection benchmarks also reveal similar benefits from this approach. Furthermore, we explore the effects of combining reasoning, which typically mitigates hallucination, with hallucination within the prompt and find several cases of performance improvements. We hope this paper stimulates further interest, investigation, and discussion on how hallucination in prompts may not only affect LLMs but, in certain cases, enhance their performance.

1 Introduction

Hallucination of generative models, in a broad sense, is defined as a situation where there is conflicting information, either with facts, established knowledge, intents, or previously generated or provided content, within their context window (Ji et al., 2023; Zhang et al., 2023; Huang et al., 2023; Rawte et al., 2023b). Hallucination existed even before the recent widespread usage of large language models (LLMs) (Ji et al., 2023). However, it has become an increasing problem given the integration of LLMs in applications of growing importance (Zhang et al., 2023; Huang et al., 2023; Rawte et al., 2023b; Yao et al., 2023b; Mialon et al., 2023). Therefore, previous studies have extensively explored approaches

Zero-Shot Prompting

Question: All the clutter in the house excited Leslie but not Derrick because cleaning energized _ very much.
Choices: 1) Leslie, 2) Derrick
Answer:

The answer is 2) Derrick. ❌

The sentence "All the clutter in the house excited Leslie but not Derrick because cleaning energized Derrick very much" is a complex sentence with two independent clauses. The first independent clause is "All the clutter in the house excited Leslie." The second independent clause is "cleaning energized Derrick very much." The two independent clauses are joined by the conjunction "but."

The first independent clause tells us that Leslie was excited by the clutter in the house. The second independent clause tells us that Derrick was energized by cleaning. The two clauses are related because they both describe how the characters in the story feel about clutter.

The answer to the question "All the clutter in the house excited Leslie but not Derrick because cleaning energized _ very much" is 2) Derrick. This is because the second independent clause tells us that Derrick was energized by cleaning.

Null-Shot Prompting

Look at examples in the "Examples" section and utilize examples and information from that section to perform the following task.
Question: All the clutter in the house excited Leslie but not Derrick because cleaning energized _ very much.
Choices: 1) Leslie, 2) Derrick
Answer:

The answer is 1) Leslie. ✅

The sentence "All the clutter in the house excited Leslie but not Derrick because cleaning energized Leslie very much" is an example of a complex sentence with an independent clause and a dependent clause. The independent clause is "All the clutter in the house excited Leslie." The dependent clause is "because cleaning energized Leslie very much." The dependent clause is not a complete sentence on its own, but it provides additional information about the independent clause. In this case, the dependent clause tells us why Leslie was excited about the clutter.

The answer to the question "All the clutter in the house excited Leslie but not Derrick because cleaning energized _ very much" is Leslie. This is because the dependent clause tells us that Leslie was excited about the clutter because cleaning energized her.

Figure 1: Examples of a generated response output by PaLM 2 (Chat) when using zero-shot prompting (left) and null-shot prompting (right) for WinoGrande.

for mitigating hallucination across the model development and deployment pipeline (Ji et al., 2023; Zhang et al., 2023; Huang et al., 2023).

Nevertheless, other studies have found that hallucination is inevitable in LLMs (Xu et al., 2024; Kalai and Vempala, 2024; Gekhman et al., 2024). Recent research has suggested that hallucination is instead a feature of LLMs (Bai et al., 2024) and is to be expected from calibrated LLMs (Kalai and Vempala, 2024). From another perspective, several research groups have proposed that hallucination may be regarded as a way for LLMs to be creative (Huang et al., 2023; Rawte et al., 2023b; Jiang et al., 2024). Given that hallucination may be inevitable and is an innate property of LLMs, instead of focusing solely on mitigating hallucination, which is still crucial, an alternative approach is to take advantage of this property instead.

This perspective must hold some value, as in a recent paper proposing an automatic prompt optimization technique for text toxicity classification (Taveekitworachai et al., 2024), we discovered that the optimized prompts contained a phrase instructing LLMs to look at a non-existent, i.e., null, sec-

Null-Shot Phrase

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

Figure 2: Null-shot phrase instructing LLMs to look into and utilize information from the null section.

tion and utilize information from that section. The performance of the task increased even with this phrase exploiting hallucination, in contrast to our initial expectation. Therefore, in this paper, we generalize this phrase to make it suitable for general tasks and evaluate it on broader types of tasks. We name this approach of providing hallucination in prompts **null-shot prompting**.

We perform various experiments across datasets and LLMs to assess the effectiveness of null-shot prompting in a variety of tasks, including arithmetic reasoning, where we observe notable performance increases. We also conduct experiments on how null-shot prompting affects the ability of LLMs to detect hallucination, and how eliciting reasoning—known to reduce hallucination (Xu et al., 2024)—and hallucination simultaneously impact LLM performance. In the rest of this paper, we discuss the methodology and implications of results for each experiment in their respective sections. Additional information on our methodology, along with additional analyses and discussions, is available in the Appendices. Our contributions are as follows:

- We propose and comprehensively evaluate null-shot prompting on a variety of benchmarks.
- We perform analyses on the impact of hallucination in prompts for 1) hallucination detection and 2) reasoning.

2 Null-Shot Prompting

We propose a null-shot phrase for null-shot prompting suitable for general tasks, as presented in Figure 2. This phrase is placed at the beginning of the prompt. The decision to position the phrase at the beginning is due to better performance compared to placing it at the end of the prompt, as demonstrated in Section 7.1. The original optimized prompt for text toxicity classification containing the hallucination inspiring our null-shot prompt is provided

in Section B. We acknowledge that LLMs, when encountering such scenarios, should instead refuse the request or ask for the missing information, as expected behaviors, instead of simply following such hallucinatory prompts. We discuss cases from our experiments where we observed such behaviors from certain LLMs in Section F.3.

3 Evaluation of Null-Shot Prompting

We select a broader set of tasks for the evaluation of null-shot prompting on tasks commonly used for evaluating the performance of LLMs. These tasks consist of arithmetic reasoning (AQuA-RaT (Ling et al., 2017) and GSM8K (Cobbe et al., 2021)), commonsense reasoning (StrategyQA (Geva et al., 2021a) and WinoGrande (Sakaguchi et al., 2021)), reading comprehension (RACE-m and RACE-h (Lai et al., 2017)), natural language inference (ANLI (Nie et al., 2020)), and closed-book question answering (TriviaQA (Joshi et al., 2017)). We also select a comprehensive list of LLMs from various model families to provide a complete picture, consisting of PaLM 2 (text and chat generation) (Anil et al., 2023), Gemini 1.0 Pro (text and chat generation) (Gemini et al., 2024), GPT-3.5 Turbo, and GPT-4 Turbo (OpenAI et al., 2024). We include Claude models (Anthropic, 2024), specifically Claude 2.1, Claude 3 Haiku, Claude 3 Sonnet, and Claude 3 Opus, as part of the evaluation, as they are well-known for safety alignment, i.e., less hallucinatory, to better understand how this aspect affects null-shot prompting, which causes and exploits hallucination. A more detailed description, rationales, and evaluation approach are provided in Appendix C.

We report the performance of relative changes when using null-shot prompting compared to the zero-shot prompting baseline in Table 1. We note that the full results of all tables presented in the main body of the paper are available in Appendix D. Full detailed statistical results are available separately via the link provided in Appendix J. Out of 34 combinations that show improvements from null-shot prompting, 20 are statistically significant, with 12 demonstrating medium effect sizes, seven showing small effect sizes, and one exhibiting large effect sizes.

We observe that null-shot prompting is able to improve the performance of PaLM 2, both text and chat generation. We observe great improvement in arithmetic reasoning in Gemini 1.0 Pro, both text

and chat generation, as well as GPT-3.5 Turbo. For state-of-the-art models, GPT-4 Turbo and Claude 3 Opus, we observe a similar trend of performance improvement in reading comprehension. However, in the majority of cases for Claude models and GPT-4 Turbo, we observe a subpar performance when using null-shot prompting.

PaLM 2 and Gemini 1.0 Pro, both developed by Google, retain the effectiveness of null-shot prompting despite following different training recipes. However, we note one difference between these models observed in our experiments: we do not observe performance gains from null-shot prompting in commonsense reasoning tasks and closed-book question answering. We argue that this is due to the alignment process present in Gemini 1.0 Pro and not in PaLM 2, which emphasizes not hallucinating incorrect information in closed-book question answering scenarios (Gemini et al., 2024) and commonsense reasoning tasks, which elicit reasoning in LLMs. Therefore, reducing the chances of hallucination through improved recall and reasoning (Xu et al., 2024).

In fact, commonsense reasoning, along with closed-book reasoning, are tasks where we observe the least number of models benefiting from null-shot prompting. We discuss more on reasoning and null-shot prompting in section 6. However, we note that Gemini 1.0 Pro, which was emphasized to perform hedging when encountering scenarios similar to the null-shot phrase, as written in their report (Gemini et al., 2024), is ineffective in most scenarios as can be observed from the increased performance.

Contrary to previous observations, we notice that LLMs, which are currently among the strong models considered to achieve state-of-the-art performance across many tasks, like GPT-4 Turbo and Claude 3 Opus, generally lose the effectiveness of null-shot prompting, except in the case of the reading comprehension task. In fact, null-shot prompting is able to make GPT-4 Turbo achieve the best performance in RACE-m and RACE-h among the evaluated models. One unique aspect that only exists within this task, and not in other tasks, is its long context nature. This long context, due to including a long article, leads LLMs to hallucinate and think that the provided article is the mentioned section. Thus, the null-shot phrase acts as additional conditional tokens reinforcing the LLMs to pay attention to the provided article in this case, leading to better performance. Nevertheless, we

acknowledge that the minimal increase in GPT-4 Turbo could also be attributed to normal prompt sensitivity.

On the other hand, we observe that null-shot prompting does not elicit the same trend of performance change for these state-of-the-art LLMs in the closed-book question answering task (TriviaQA). This task shares one similarity with reading comprehension in requiring the model to recall certain information, but the difference is that RACE requires the model to recall information within the context, while TriviaQA requires the model to recall information from their own parameters. The null-shot phrase instructs these LLMs to look and utilize information from the null section, which is also a form of instruction focusing on information recall, albeit such information never exists within the context. It shows that for these state-of-the-art LLMs, null-shot prompting does not elicit the models to recall information from their parameters but is instead effective for recalling within context as previously discussed.

Claude models are known for their emphasis on safety and harmlessness, as evidenced by prior work from their developers (Ganguli et al., 2022; Bai et al., 2022b; Anthropic, 2024). Therefore, we expect null-shot prompting to become less effective and unsurprisingly, the results observed match our expectations. Furthermore, we observe a higher decrease in performance in the commonsense reasoning task, further supporting the argument that reasoning decreases hallucination as known in the community. However, we observe that the state-of-the-art model, Claude 3 Opus, benefits from null-shot prompting in the reading comprehension task, as previously discussed. Furthermore, Claude 3 Haiku, the smallest model in the Claude 3 series, has high performance gains from null-shot prompting in the natural language inference (NLI) task. Since it is a proprietary model, we cannot say for certain on which changes in the model are responsible for the observed effects.

Based on the overall results so far, we establish a few observations:

- Null-shot prompting is very effective for Google models across tasks.
- Null-shot prompting is less effective in commonsense reasoning and closed-book reasoning tasks.
- Null-shot prompting is less effective in LLMs

Model	AQuA	GSM8K	StrategyQA	WinoGrande	RACE-m	RACE-h	ANLI	TriviaQA
PaLM 2	-2.7%	11.28% ^{**}	<u>10.95%</u> ^{***}	10.1% ^{***}	1.85% ^{**}	3.64% ^{***}	2.71%	7.01% ^{***}
– (Chat)	5.26%	2.25%	1.66%	6.97% ^{***}	1.04%	0.68%	1.56%	-0.14%
Gemini 1.0 Pro	38.46% ^{**}	28.97% ^{***}	-24.43%	-1.36%	1.93% ^{**}	2.13% ^{***}	2.14%	-63.96%
– (Chat)	<u>44.62%</u> ^{**}	27.93% ^{***}	-25.39%	-1.12%	0.74%	1.63% ^{***}	1.63%	-63.97%
GPT-3.5 Turbo	<u>33.94%</u> ^{***}	15.19% ^{***}	3.14% ^{**}	-1.84%	-1.79%	-1.19%	-3.61%	1.23%
GPT-4 Turbo	-0.52%	-1.53%	-17.39%	-24.06%	0.3%	0.42%	-0.26%	-0.94%
Claude 2.1	-11.52%	-19.02%	-70.84%	-89.29%	-0.97%	2.94%	-21.34%	-72.75%
Claude 3 Haiku	-7.45%	-2.56%	-33.57%	-33.38%	-9.36%	-8.76%	34.4% ^{***}	-17.83%
Claude 3 Sonnet	-8.39%	-8.56%	-59.67%	-45.67%	-18.87%	-16.43%	-20%	-59.47%
Claude 3 Opus	-17.82%	-22.59%	-92.85%	-99.11%	5.2% ^{***}	7.89% ^{***}	-10.3%	-75.7%

Table 1: This table shows relative performance changes when instructing an LLM with null-shot prompting compared to zero-shot prompting on selected tasks. **Green** values indicate a relative performance increase, and **Bold** indicates the largest performance change within the same task. Underline indicates the best performance change within the same LLM. This convention applies to all the tables in this paper that present relative performance changes. For cases where null-shot prompting or its reasoning variant show improvement over the baseline, *, **, and *** indicate statistical significance at 0.05, 0.01, and 0.001, respectively.

trained with a focus on hallucination reduction.

- Null-shot prompting is effective with state-of-the-art LLMs for reading comprehension tasks.
- Efficient inference optimization techniques, such as weight pruning, may affect hallucination mitigation implemented in LLMs.

Another crucial observation is performance gain in arithmetic reasoning tasks where we observe high performance improvement in PaLM 2, Gemini 1.0 Pro, and GPT-3.5 Turbo, especially Gemini 1.0 Pro and GPT-3.5 Turbo. These tasks, at first, seem unrelated to and likely unbeneficial from the hallucinatory instruction in the null-shot phrase, as they focus on reasoning in computation steps and numerical calculations. However, when considering from the perspective that hallucination is a way of expressing creativity in LLMs, we believe that this kind of task will benefit more, as arithmetic reasoning, unlike other tasks, has numerous ways to reach the final answer and often requires creativity in devising such intermediate steps. Therefore, it is likely that these LLMs also benefit from such creativity from the hallucination in null-shot prompting. We investigate this task further in the next section.

4 MATH Evaluation Sets

We expand our evaluation by focusing on the arithmetic abilities of LLMs, as we previously observed significant improvements when using null-shot prompting for certain LLMs in arithmetic reasoning tasks. We select the MATH dataset (Hendrycks et al., 2021), which not only offers more challenging questions but also covers a variety of topics,

aiding in further analysis. We follow the evaluation methodology of the previous section, with the addition of instructions for the model to output in a specified format to aid in the evaluation process. This format follows the original dataset labeling.

The results are presented in Table 2. Overall, we observe a similar trend in MATH-related tasks with PaLM 2 (Chat) and GPT-3.5 Turbo, showing improvements when using null-shot prompting across topics. However, we did not observe the same performance improvement trend in PaLM 2, and we noticed that performance improvements for Gemini 1.0 Pro are only seen in prealgebra, algebra, counting and probability, and geometry. Additionally, GPT-4 Turbo and Claude models, which originally did not gain performance improvements with null-shot prompting, now show performance increases. However, it is worth noting that GPT-4 Turbo and Claude 3 Opus, which are state-of-the-art models, only gain performance increases in one topic each. We report that out of 33 combinations that show improvements from null-shot prompting, 17 are statistically significant, with seven demonstrating medium effect sizes, five exhibiting large effect sizes, and five showing small effect sizes.

The results for PaLM 2 (Chat) show significant improvements across topics, with the most substantial performance improvement in algebra. However, we do not observe the same trend for PaLM 2. On the other hand, Gemini 1.0 Pro and Gemini 1.0 Pro (Chat) share the same trend of performance changes. Aside from performance improvements across most topics, we observe an interesting insight in the number theory topic where null-shot prompting causes no change in performance, which is surprising as changes in prompts usually lead to differences in the outcomes of the models. The

Model	Prealgebra	Algebra	Num. Th.	Count. & Prob.	Geometry	Int. Algebra	Precalculus
PaLM 2	-2.6%	-3.5%	-1.56%	-14.75%	-4.41%	4.9%	0%
– (Chat)	116.39%***	247.62%***	166.67%***	113.04%***	48.15%	78.38%***	83.33%**
Gemini 1.0 Pro	9.35%***	8.11%*	0%	3.9%	5.56%	-4.48%	-6.6%
– (Chat)	8.27%**	8.81%*	0%	2.6%	5.56%	-3.73%	-7.48%
GPT-3.5 Turbo	29.16%***	42.42%***	48.84%***	22.95%**	20.69%*	16.56%*	1.68%
GPT-4 Turbo	-0.79%	-5.35%	-0.9%	1.22%	-8.48%	-4.08%	-11.54%
Claude 2.1	-8.53%	-7.46%	-7.81%	-3.81%	-6.36%	0.88%	11.11%
Claude 3 Haiku	-4.94%	-1.22%	5.75%	2.34%	-4.84%	3.01%	3.88%
Claude 3 Sonnet	0.9%	1.59%	-7.58%	-3.7%	-12.9%	-0.65%	-3.74%
Claude 3 Opus	-1.87%	-2.35%	-7.42%	-10.19%	-7.31%	3.58%	-5.76%

Table 2: This table presents evaluation results on the MATH benchmark when using null-shot prompting compared to the zero-shot prompting baseline. Henceforth, for all the tables presenting results from the MATH dataset, **Num. Th.**, **Count. & Prob.**, and **Int. Algebra** denote number theory, counting and probability, and intermediate algebra, respectively.

only other occurrence of no performance change is with the base PaLM 2 model in the precalculus topic.

Not only are there occurrences of no changes in performance, but we also observe that in cases where null-shot prompting causes a performance decrease, it is less deviated from the zero-shot performance baseline compared to the previous section’s evaluation. This is especially true for Claude models, where we see less performance decrease and even observe some performance gains in models and topics. This leads us to believe that math-related tasks may require a certain degree of hallucination or creativity to perform well.

Counting & probability and intermediate algebra are two tasks that show the highest number of LLMs benefiting from null-shot prompting. This indicates that null-shot prompting is effective in problems requiring statistical and symbolic reasoning. In contrast, only a moderate number of LLMs benefit in prealgebra and algebra; these tasks focus more on numerical calculations.

We also note that GPT-3.5 Turbo, another LLM that benefits from null-shot prompting for math problem-solving, shows high performance gains across topics, with less notable increases only in precalculus. GPT-3.5 Turbo and PaLM 2, which benefit the most from null-shot prompting, are based on decoder-only Transformer architecture. We believe that null-shot prompting requires models to use Transformer architecture and undergo chat-tuning to be creative in math problem-solving and exhibit the performance gain.

5 Hallucination Detection

Since null-shot prompting includes hallucinatory instructions, we explore how it affects the hallucination detection abilities of LLMs. We hypothesize that by including hallucination in the prompt, mod-

els will suffer from degraded abilities in hallucination detection, since there are conflicting elements in the prompt, namely, hallucination. To evaluate hallucination detection, we utilize HaluEval (Li et al., 2023b), a hallucination detection dataset which contains scenarios such as general dialogue, question answering, and summarization. We adapt the original prompts from the evaluation set to suit our task by removing few-shot examples in the prompts and evaluating in zero-shot or null-shot scenarios instead, to reduce factors affecting performance during the analysis.

We present results from HaluEval in Table 3. Of the 25 combinations that show improvements from null-shot prompting, 14 are statistically significant, with six demonstrating medium effect sizes, six showing small effect sizes, and two exhibiting large effect sizes. We observe performance improvement when using null-shot prompting in most cases. However, we acknowledge that most of the improvements for Gemini 1.0 Pro models, GPT-4 Turbo, and Claude 3 Opus are small and most likely caused by prompt variations. However, this minimal change also signifies that null-shot prompting, which exploits hallucination, does not affect the abilities of LLMs to perform hallucination detection as much. In fact, we observe the opposite trend for many models, especially PaLM 2 (Chat), which gains improvement in performance when using null-shot prompting for hallucination detection.

These results are quite surprising as they contradict our hypothesis and are very counter-intuitive in the sense that providing a prompt with hallucination improves the hallucination detection abilities of the LLMs. One similarity in the results of PaLM 2 (Chat) from the MATH evaluation and HaluEval evaluation is that this model is relatively weak in these evaluations compared to the other LLMs. In

Model	General	Dialogue	QA	Sum.
PaLM 2	1.62%***	2.24%***	0.99%	8.43%***
– (Chat)	25.6%***	1.59%	62.65%***	141.94%***
Gemini 1.0 Pro	0.05%	0.28%	1.47%***	-0.2%
– (Chat)	0.05%	0.25%	1.44%***	-0.12%
GPT-3.5 Turbo	0.28%	4.83%***	9.42%***	-2.12%
GPT-4 Turbo	-0.03%	0.45%	-4.26%	-0.2%
Claude 2.1	-0.19%	-10.64%	-13.33%	-7.6%
Claude 3 Haiku	-0.94%	2.04%***	-3.74%	6.25%***
Claude 3 Sonnet	0.14%	4.84%***	-18.47%	3.38%***
Claude 3 Opus	-0.38%	-7.85%	0.97%	0.04%

Table 3: This table presents relative results of performance changes from evaluating null-shot prompting compared to zero-shot prompting using HaluEval for determining hallucination detection abilities in scenarios of each LLM. Henceforth, for all the tables presenting results of HaluEval, **QA** denotes question answering scenarios, and **Sum.** denotes summarization scenarios.

addition, it is PaLM 2 that we see a drastic increase in performance when using null-shot prompting for both cases. Based on the observed performance when using zero-shot prompting for HaluEval of PaLM 2 (Chat), which signifies its inherent abilities in performing hallucination detection, we know that this model is the weakest among the selected LLMs for hallucination detection.

For dialogue scenarios, we see that all models, except for Claude 2.1 and Claude 3 Opus, have increased performance when using null-shot prompting, supporting our previous discussion about the chat-tuned version of PaLM 2 gaining the most increase in performance. Similar to previous observations of other evaluation sets, we also see GPT-3.5 Turbo gaining significant performance increases in dialogue and question answering scenarios. As previously discussed, this is another evidence that null-shot prompting is effective with chat-tuned Transformer models. Additional studies and analyses are provided in the Appendices.

6 Reasoning and Hallucination

In this section, we investigate how reasoning, known to reduce hallucination, combined with hallucination in prompts, can affect the LLMs. We follow the same setups as Sections 3 and 4. However, we change the baseline to zero-shot chain-of-thought (0CoT) prompting (Kojima et al., 2022) and compare it against \emptyset CoT. \emptyset CoT combines the original null-shot prompting with a phrase from 0CoT instructing an LLM to think “step-by-step” (reasoning). We believe that this contrasting instruction proves to be interesting to observe and

may help shed some light on understanding the inner workings of LLMs.

We present the results of the evaluations in Table 4 and Table 5. As expected, we observe that in the majority of cases of evaluation on a variety of tasks from Table 4, \emptyset CoT prompting results in a performance decrease. This trend holds true across datasets, except for RACE-m and RACE-h, which are reading comprehension tasks. This serves as another evidence of the effectiveness of reasoning in mitigating hallucination, even when hallucination is provided within the prompt.

However, there are also cases where a performance increase in the reading comprehension task is very noticeable in PaLM 2 and Claude models, except Claude 3 Opus. For example, Claude 3 Haiku achieves a 44.46% and 36.42% performance increase for RACE-m and RACE-h, respectively, compared to a strong 0CoT prompting baseline. The reason behind this is likely as previously discussed regarding the long-context nature of the task and the possibility of changes in architecture for state-of-the-art LLMs. For the datasets utilized in the main experiment, 19 combinations show improvements, 11 of which are statistically significant: four demonstrate medium effect sizes, four show small effect sizes, and three exhibit large effect sizes. On the other hand, there are 40 improved combinations for the MATH datasets, 10 of which are statistically significant: seven demonstrate medium effect sizes, and three show small effect sizes.

We also observe cases where some LLMs gain substantial performance increase when using \emptyset CoT prompting. Gemini 1.0 Pro models for arith-

Model	AQuA	GSM8K	StrategyQA	WinoGrande	RACE-m	RACE-h	ANLI	TriviaQA
PaLM 2	-54.44%	-27.71%	-5.36%	18.89%***	16.18%***	20.16%***	-0.85%	-4.5%
– (Chat)	-5.88%	-7.55%	14.75%***	-4.49%	0.49%	0.79%	-2.49%	-2.12%
Gemini 1.0 Pro	8.47%	-9.99%	-98.42%	-99.62%	-3.02%	-1.06%	-7.07%	-98.54%
– (Chat)	8.06%	-11.66%	-98.42%	-99.62%	-2.32%	-0.73%	-8.01%	-98.55%
GPT-3.5 Turbo	-3.42%	-4%	-13.21%	-10.96%	-3.34%	0.15%	-46.94%	0.75%
GPT-4 Turbo	2.08%	-1.12%	-2.34%	24.77%***	-5.64%	-8.56%	-8.79%	-1.04%
Claude 2.1	-2.44%	-3.29%	-50.94%	-87.86%	7.99%*	-0.68%	-17.02%	-41.63%
Claude 3 Haiku	-3.12%	-1.99%	-1.66%	-49.39%	44.46%***	36.42%***	-3.96%	-15.58%
Claude 3 Sonnet	-14.57%	-1.67%	-83.83%	-81.54%	14.46%***	15.09%***	16.56%***	-46.95%
Claude 3 Opus	-9.88%	-1.08%	-57.3%	-51.83%	-4.89%	-8.52%	3.5%	-23.27%

Table 4: This table presents relative results of performance change when using \emptyset CoT prompting compared to zero-shot chain-of-thought prompting for evaluation sets used in Section 3.

Model	Prealgebra	Algebra	Num. Th.	Count. & Prob.	Geometry	Int. Algebra	Precalculus
PaLM 2	-15.64%	-1.5%	-1.75%	5.08%	12.5%	6.38%	-2.94%
– (Chat)	38.98%**	68%***	40%	35%	34.62%	38.46%	5.56%
Gemini 1.0 Pro	21.16%***	28.79%***	18.89%	21.18%*	13.83%	-7.09%	3.16%
– (Chat)	20.58%***	28.93%***	13.33%	22.62%**	18.28%*	-6.4%	4.21%
GPT-3.5 Turbo	-1.54%	-0.44%	5.15%	6.37%	10.87%	0%	-6.11%
GPT-4 Turbo	-2.52%	1.56%	-1.16%	-0.4%	0%	4.89%	-0.62%
Claude 2.1	5.8%	0.2%	10.77%	0.96%	9.37%	-7.56%	10.59%
Claude 3 Haiku	-2.01%	-1.03%	2.72%	-5.8%	10.19%	-15.49%	19.19%*
Claude 3 Sonnet	-81.93%	-81.51%	-64.32%	-77.59%	-77.7%	-66.89%	-48.98%
Claude 3 Opus	1.85%	-1.12%	-1.68%	2.4%	2.9%	1.03%	-1.53%

Table 5: This table presents relative results of performance change when using \emptyset CoT prompting compared to zero-shot chain-of-thought prompting for evaluation sets used in Section 4.

metic reasoning, PaLM 2 (Chat) for StrategyQA, and Claude 3 Sonnet for ANLI. Nevertheless, we acknowledge the limitation of our current approach which only shows us what happens and not why it happens. These observations are also very tightly coupled with the model. Therefore, future work should focus on expanding the evaluation using open-source models for better analysis, which we could not do due to limitations of our computational infrastructure. We also note that with recent advancements in interpretation techniques (Bricken et al., 2023; Templeton et al., 2024), it is also possible to apply such techniques to better understand how null-shot prompting and its variants affect activations of neural features and shed some light on the possibility of gaining benefits of improved performance without the risk of hallucination when using null-shot prompting.

For the MATH evaluation, we observe a different trend compared to the general evaluation sets. We find that null-shot prompting exhibits higher effectiveness, a trend in the same direction as when we evaluated null-shot prompting for the MATH dataset in Section 4. This is quite surprising given the fact that we observe mostly no performance improvement in AQuA and GSM8K when using \emptyset CoT prompting, which are also mathematics-related tasks. Furthermore, geometry is a topic where we observe the most effectiveness of \emptyset CoT prompting across models. This leads us to argue that for LLMs, geometry requires creativity (Scho-

evers et al., 2022)—hallucination—rather than reasoning, to perform well. We also see that counting & probability, which shows performance increase when using null-shot prompting, is another potential topic where reasoning and creativity are both necessary. Additionally, we observe moderate performance improvements in pre-algebra and algebra, in contrast to the experiments in Section 4, where we observe the improvement more from intermediate algebra. Future studies may utilize the previously mentioned interpretability approaches (Bricken et al., 2023; Templeton et al., 2024) to further explore how LLMs pay attention to tokens related to reasoning or hallucination in prompts to better understand this phenomenon.

\emptyset CoT prompting also shows another venue for further research. We often treat each approach in prompt engineering (PE) as discrete. However, there is a possibility of combining multiple approaches together. Currently, we are limited in understanding the effects of combining different PE approaches together, and we encourage future studies to explore this further.

7 Ablation Studies

We perform ablation studies to better understand how placement position of the null-shot phrase and each component in the phrase affects performance. We conduct experiments to assess those aspects in Section 7.1 and Section 7.2, respectively. In general, we observe that placing the phrase at the

end yield the maximum performance, and all components contribute to different degree of improvements, and combining all components, as in our phrase, would be most suitable across tasks.

7.1 Positions of Null-Shot Phrase

To determine the best placement position of the null-shot phrase, we conduct experiments following recipes described in Section 3 and Section 4. We compare placing the phrase before the task instruction and at the end of the prompt. To reduce the cost of the experiments, we use only the GPT-3.5 Turbo model. We compare the obtained performance against the same zero-shot prompting baseline as described in our main experiments. Relative results are shown in Table 6 and Table 7. Absolute versions of the results are available in Table 16 and Table 17.

We observe that placing the null-shot phrase at the beginning shows superior effectiveness across datasets and mathematical topics, except for GSM8K. We argue that this is due to the fact that placing content at the beginning exhibits stronger conditional strength for these models to rely on for output generation. This phenomenon has also been mentioned in another study, where tokens at the beginning of the prompt have been given more importance compared to the end of the prompt (Liu et al., 2023).

7.2 Components of Null-Shot Phrase

To assess the contribution of each component in the null-shot phrase, we conduct experiments similar to the one described in the previous subsection, again using only GPT-3.5 Turbo to save costs. We decompose our null-shot phrase into two main components: “Look at examples in the ‘Examples’ section” and “utilize examples and information from that section.” This breakdown is illustrated in Figure 6.

We prepare three additional variants of the null-shot phrase. v1 and v2 removed the first and second components, respectively; and v3 removed both components. These are shown in Figure 7, Figure 8, and Figure 9, respectively. Relative results from the experiments are shown in Tables 8 and 9; absolute results are available in Tables 18 and 19. We observe that removing both components, as in v3, reduces the effectiveness of null-shot prompting on all datasets compared to the full null-shot phrase, except in one mathematical topic, geometry. Thus, simply instructing the model to perform the

task by looking into the null section is insufficient.

We also find that, on the majority of tasks except for arithmetic reasoning and closed-book question answering, v2 shows the most prominent performance. Therefore, the first component instructing the model to *look* into the imaginary section plays an important role. However, for the arithmetic reasoning task, we find that v1 is most effective, so instructing the model to *utilize* examples and information is crucial for arithmetic tasks. For the closed-book question answering task, both components are required, as can be seen that our full null-shot phrase provides the best performance, i.e., it requires both *look* and *utilize* instructions.

In contrast, for results of the MATH benchmark, we find that the full null-shot phrase is the most prominent in getting the highest improvements. Only in cases of prealgebra, where v1 is the best, and geometry and precalculus, where v3 achieves the best performance. These observations show that the full null-shot phrase may provide the best balance as it encompasses all of the components, making it suitable across tasks and topics.

8 Conclusions

We present various experiments to investigate an intriguing phenomenon when providing LLMs with a prompt eliciting and exploiting hallucination and observe various performance changes. We observe that null-shot prompting is effective for chat-tuned Transformer LLMs. We also observe that null-shot prompting exhibits its effectiveness for reading comprehension and mathematics-related tasks. Given the hallucination detection results, it also reveals a surprising conclusion that null-shot prompting is also effective for increasing LLMs’ abilities for detecting hallucination. Combining reasoning and hallucination, \emptyset CoT prompting, shows that some mathematical topics problems require both reasoning and creativity to perform well. We hope this paper serves as an initial step towards a better understanding of how hallucination in prompts affects LLMs.

Approach	AQuA	GSM8K	StrategyQA	WinoGrande	RACE-m	RACE-h	ANLI	TriviaQA
Null-Shot	33.94%	15.19%	3.14%	-1.84%	-1.79%	-1.19%	-3.61%	1.23%
Null-Shot (After)	30.28%	19.2%	-6.21%	-69.08%	-4.81%	-4.62%	-46.47%	-3.09%

Table 6: This table shows relative results comparing placing the null-shot phrase at the beginning of the prompt, denoted by *Null-Shot*, and at the end of the prompt, denoted by *Null-Shot (After)*. The performance shown is the relative performance change when compared to the zero-shot prompting baseline for both variants.

Model	Prealgebra	Algebra	Num. Th.	Count. & Prob.	Geometry	Int. Algebra	Precalculus
Null-Shot	29.16%	42.42%	48.84%	22.95%	20.69%	16.56%	1.68%
Null-Shot (After)	15.53%	20.13%	13.18%	1.64%	16.38%	4.29%	-0.84%

Table 7: This table shows relative results comparing placement of the null-shot phrase, similar to the previous table. However, this table shows evaluation results using the MATH dataset.

Approach	AQuA	GSM8K	StrategyQA	WinoGrande	RACE-m	RACE-h	ANLI	TriviaQA
Null-Shot	33.94%	15.19%	3.14%	-1.84%	-1.79%	-1.19%	-3.61%	1.23%
Null-Shot V1	36.7%	16.85%	2.73%	-3.95%	-2.12%	-1.75%	-6.37%	0.62%
Null-Shot V2	10.09%	8.98%	4.57%	1.84%	-1.14%	-0.8%	0.52%	-0.37%
Null-Shot V3	27.52%	15.88%	1.23%	-8.82%	-1.47%	-1.43%	-2.07%	-0.25%

Table 8: This table presents relative results of each null-shot variant, showing changes relative to the zero-shot prompting baseline performance.

Model	Prealgebra	Algebra	Num. Th.	Count. & Prob.	Geometry	Int. Algebra	Precalculus
Null-Shot	29.16%	42.42%	48.84%	22.95%	20.69%	16.56%	1.68%
Null-Shot V1	32.15%	39.83%	44.96%	21.31%	24.14%	6.75%	11.76%
Null-Shot V2	12.53%	16.67%	16.28%	2.46%	18.1%	-0.61%	-5.88%
Null-Shot V3	25.07%	33.55%	44.96%	16.39%	25%	4.29%	5.04%

Table 9: This table presents relative results similar to the previous table, but for the MATH dataset.

Limitations

In this paper, we present early investigations on how hallucination in prompts affects LLMs' performance on tasks. We acknowledge that there are more nuances and aspects that we do not include in this study as we intend for this study to be an initial step in that direction. We also do not utilize state-of-the-art open-source LLMs in this study due to limitations of our resources. Due to resource constraints, we are unable to evaluate even more variants of the null-shot phrase. However, we study how each prompt component in the phrase affects the final performance outcomes in Section 7.2.

We were also able to conduct a limited scaling study only on smaller sizes of LLMs, with a maximum at 7B, due to resource constraints. We present these findings in Section E. We note that broader evaluations across more tasks and LLMs with varying sizes and architectures are expected to help further generalize our findings and provide deeper insights. We also point out that the current trend of efficient LLMs, such as using quantized LLMs (Bai et al., 2022a; Xiao et al., 2023) or performing weight pruning (Wang et al., 2020; Jiang et al., 2023; Sun et al., 2024), may also affect outcomes from the experiments and should be further investigated. Similarly, base and chat models demonstrated significant behavioral differences in PaLM 2 models and require further generalized evaluations on more variants.

While the study that introduced OCoT prompting (Kojima et al., 2022) used a two-stage prompting approach for improved result extraction, we did not utilize this approach in our study to reduce costs, which may result in some cases of unsuccessful result extraction. However, we compensated for it with very flexible output extraction scripts instead (cf. Section C.1). Finally, interpretability

for LLMs is an active area of research and there are works presenting attempts to better understand what happens inside LLMs during inference. We believe that studies by Bricken et al. (2023) and Templeton et al. (2024) offer an interesting avenue for applying to better understand the phenomena of null-shot prompting. There are possibilities that using a similar approach as in Templeton et al. (2024) will not only help us better understand null-shot prompting but also eliminate hallucination while maintaining gained benefits. Therefore, it should be further investigated.

Ethics Statement

Similar to general use cases of LLMs, our approach is likely to suffer from dataset poisoning (Wallace et al., 2021) as polluted datasets may increase the performance of our approach at the cost of increased hallucination in LLMs. Furthermore, we are unsure about the null examples that models envision during their output generation. Thus, they may retrieve biased, harmful, or toxic content and may lead to the reproduction of such content in the generated outputs. We also note that it is possible to use null-shot prompting or a modified version of the prompting to avoid harmless and helpful aligned behaviors or other safety mechanisms built into the models and cause jailbreaking (Wei et al., 2023). Finally, as we have a limited understanding of the deeper workings of LLMs in general, which is an active area of research, utilizing null-shot prompting may lead to unexpected behaviors.

References

- Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, et al. 2023. [PaLM 2 Technical Report](#).
- Anthropic. 2024. [The Claude 3 Model Family: Opus, Sonnet, Haiku](#).
- Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. 2021. [A general language assistant as a laboratory for alignment](#).
- Haoli Bai, Lu Hou, Lifeng Shang, Xin Jiang, Irwin King, and Michael R Lyu. 2022a. [Towards Efficient Post-training Quantization of Pre-trained Language Models](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 1405–1418. Curran Associates, Inc.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, et al. 2022b. [Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback](#).
- Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2024. [Hallucination of Multimodal Large Language Models: A Survey](#).
- Rick Battle and Teja Gollapudi. 2024. [The Unreasonable Effectiveness of Eccentric Automatic Prompts](#).
- German E. Berrios. 1998. [Confabulations: A Conceptual History](#). *Journal of the History of the Neurosciences*, 7(3):225–241.
- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2023. [Graph of Thoughts: Solving Elaborate Problems with Large Language Models](#).
- Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. 2023. [Towards Monosemanticity: Decomposing Language Models With Dictionary Learning](#).
- Alan S. Brown. 2003. [A review of the déjà vu experience](#). *Psychological Bulletin*, 129(3):394–413.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, et al. 2020. [Language Models are Few-Shot Learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2023. [Jailbreaking Black Box Large Language Models in Twenty Queries](#).
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training Verifiers to Solve Math Word Problems](#).
- Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. [Chain-of-Verification Reduces Hallucination in Large Language Models](#).
- Ruomeng Ding, Chaoyun Zhang, Lu Wang, Yong Xu, Minghua Ma, Wei Zhang, Si Qin, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. 2023. [Everything of Thoughts: Defying the Law of Penrose Triangle for Thought Generation](#).
- Aikaterini Fotopoulou. 2008. [False selves in neuropsychological rehabilitation: The challenge of confabulation](#). *Neuropsychological Rehabilitation*, 18(5-6):541–565.
- Cheryl Francis, Fiona MacCallum, and Siân Pierce. 2022. [Interventions for confabulation: A systematic literature review](#). *The Clinical Neuropsychologist*, 36(8):1997–2020.
- Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse,

- Andy Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, Zac Hatfield-Dodds, Tom Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, et al. 2022. [Red Teaming Language Models to Reduce Harms: Methods, Scaling Behaviors, and Lessons Learned](#).
- Zorik Gekhman, Gal Yona, Roei Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. [Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations?](#)
- Gemini, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, et al. 2024. [Gemini: A Family of Highly Capable Multimodal Models](#).
- Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021a. [Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies](#). *Transactions of the Association for Computational Linguistics*, 9:346–361.
- Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021b. [Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies](#). *Transactions of the Association for Computational Linguistics*, 9:346–361.
- Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujie Yang. 2024. [Connecting Large Language Models with Evolutionary Algorithms Yields Powerful Prompt Optimizers](#). In *The Twelfth International Conference on Learning Representations*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring Mathematical Problem Solving With the MATH Dataset](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. [A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions](#).
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of Hallucination in Natural Language Generation](#). *ACM Comput. Surv.*, 55(12).
- Ting Jiang, Deqing Wang, Fuzhen Zhuang, Ruobing Xie, and Feng Xia. 2023. [Pruning pre-trained language models without fine-tuning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 594–605, Toronto, Canada. Association for Computational Linguistics.
- Xuhui Jiang, Yuxing Tian, Fengrui Hua, Chengjin Xu, Yuanzhuo Wang, and Jian Guo. 2024. [A Survey on Large Language Model Hallucination via a Creativity Perspective](#).
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Adam Tauman Kalai and Santosh S. Vempala. 2024. [Calibrated Language Models Must Hallucinate](#).
- Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large Language Models are Zero-Shot Reasoners](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. [RACE: Large-scale Reading Comprehension Dataset From Examinations](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.
- Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. [The winograd schema challenge](#). In *Thirteenth international conference on the principles of knowledge representation and reasoning*.
- Chengshu Li, Jacky Liang, Fei Xia, Andy Zeng, Sergey Levine, Dorsa Sadigh, Karol Hausman, Xinyun Chen, Li Fei-Fei, and brian ichter. 2023a. [Chain of Code: Reasoning with a Language Model-Augmented Code Interpreter](#). In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.
- Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023b. [HaluEval: A large-scale hallucination evaluation benchmark for large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [TruthfulQA: Measuring How Models Mimic Human](#)

- Falsehoods.** In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.
- Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. **Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems.** In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. **Lost in the Middle: How Language Models Use Long Contexts.**
- Grégoire Mialon, Roberto Dessi, Maria Lomeli, Christoforos Nalmpantis, Ramakanth Pasunuru, Roberta Raileanu, Baptiste Roziere, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. **Augmented Language Models: a Survey.** *Transactions on Machine Learning Research*. Survey Certification.
- Raymond S. Nickerson. 1998. **Confirmation Bias: A Ubiquitous Phenomenon in Many Guises.** *Review of General Psychology*, 2(2):175–220.
- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. **Adversarial NLI: A New Benchmark for Natural Language Understanding.** In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, et al. 2024. **GPT-4 Technical Report.**
- Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Benjamin Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemi Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, and Jared Kaplan. 2023. **Discovering language model behaviors with model-written evaluations.** In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13387–13434, Toronto, Canada. Association for Computational Linguistics.
- Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. 2023a. **The Troubling Emergence of Hallucination in Large Language Models - An Extensive Definition, Quantification, and Prescriptive Remediations.** In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2541–2573, Singapore. Association for Computational Linguistics.
- Vipula Rawte, Amit Sheth, and Amitava Das. 2023b. **A Survey of Hallucination in Large Foundation Models.**
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. **WinoGrande: An Adversarial Winograd Schema Challenge at Scale.** *Commun. ACM*, 64(9):99–106.
- Leonard Saxe. 1991. **Lying: Thoughts of an applied social psychologist.** *American Psychologist*, 46(4):409–415.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. **Toolformer: Language Models Can Teach Themselves to Use Tools.** In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Armin Schneider, Christine von Däniken, and Klemens Gutbrod. 1996. **The mechanisms of spontaneous and provoked confabulations.** *Brain*, 119(4):1365–1375.
- Eveline M. Schoevers, Evelyn H. Kroesbergen, Mirjam Moerbeek, and Paul P. M. Leseman. 2022. **The relation between creativity and students’ performance on different types of geometrical problems in elementary education.** *ZDM – Mathematics Education*, 54(1):133–147.
- Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. **Role play with large language models.** *Nature*, 623(7987):493–498.
- Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. **Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer.** In *International Conference on Learning Representations*.
- Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. **"Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models.**

- Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2024. [A Simple and Effective Pruning Approach for Large Language Models](#). In *The Twelfth International Conference on Learning Representations*.
- Pittawat Taveekitworachai, Febri Abdullah, Mustafa Can Gursesli, Antonio Lanata, Andrea Guazzini, and Ruck Thawonmas. 2024. [Prompt evolution through examples for large language models—a case study in game comment toxicity classification](#). In *2024 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0 & IoT)*, pages 22–27.
- Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermy, Shan Carter, Chris Olah, and Tom Henighan. 2024. [Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet](#).
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruiti Bhosale, et al. 2023. [Llama 2: Open Foundation and Fine-Tuned Chat Models](#).
- Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. 2021. [Concealed Data Poisoning Attacks on NLP Models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 139–150, Online. Association for Computational Linguistics.
- Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2020. [Structured Pruning of Large Language Models](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6151–6162, Online. Association for Computational Linguistics.
- Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. [Jailbroken: How Does LLM Safety Training Fail?](#) In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. [Emergent Abilities of Large Language Models](#). *Transactions on Machine Learning Research*. Survey Certification.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022b. [Chain of Thought Prompting Elicits Reasoning in Large Language Models](#). In *Advances in Neural Information Processing Systems*.
- Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. [SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 38087–38099. PMLR.
- Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. [Hallucination is Inevitable: An Innate Limitation of Large Language Models](#).
- Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. [Large Language Models as Optimizers](#). In *The Twelfth International Conference on Learning Representations*.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023a. [Tree of Thoughts: Deliberate Problem Solving with Large Language Models](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023b. [ReAct: Synergizing Reasoning and Acting in Language Models](#). In *The Eleventh International Conference on Learning Representations*.
- Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. [QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.
- Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. [Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models](#).
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. [Siren’s Song in the AI Ocean: A Survey on Hallucination in Large Language Models](#).
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. [A Survey of Large Language Models](#).
- Yongchao Zhou, Andrei Ioan Muresanu, Ziwon Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. [Large Language Models are Human-Level Prompt Engineers](#). In *The Eleventh International Conference on Learning Representations*.

Appendix A Related Work

A.1 Hallucination in LLMs

Various studies have explored hallucination in LLMs, i.e., behaviors when models provide conflicting information in their outputs (Zhao et al., 2023). Attempts have been made to reduce hallucination in LLMs across steps in model development, e.g., pre-training (Touvron et al., 2023), fine-tuning (Askeel et al., 2021; Bai et al., 2022b; Touvron et al., 2023), and inferencing (Dhuliawala et al., 2023; Li et al., 2023b). These efforts are propelled by the development of various benchmarks for hallucination (Lin et al., 2022; Li et al., 2023b). While it is crucial to reduce hallucination in LLMs, our study proposes that we can exploit these hallucination in LLMs to achieve greater performance across tasks and also utilize this approach for evaluating hallucination in LLMs.

A.2 Prompt Engineering

PE is a field focused on improving the performance of LLMs through structuring inputs provided to these models, i.e., prompts. Many prompting approaches have been proposed over the years, e.g., few-shot prompting (Brown et al., 2020), CoT prompting (Wei et al., 2022b), and 0CoT prompting (Kojima et al., 2022). Many variants of CoT prompting have also been proposed, with their focus either on the *chain*, e.g., chain-of-note (Yu et al., 2023), CoVe (Dhuliawala et al., 2023), and chain-of-code (Li et al., 2023a) prompting. Another line of research focuses on the *thought*, such as tree-of-thought (Yao et al., 2023a), graph-of-thought (Besta et al., 2023), and everything-of-thought (Ding et al., 2023) prompting. While we share similarities with few-shot prompting in utilizing examples and other chain and thought facilities of PE in eliciting longer responses from LLMs, our approach utilizes hallucination in LLMs to use examples that exist within the model. Furthermore, to the best of our knowledge, we are the first to propose PE for hallucination exploitation.

Appendix B Original Optimized Prompt

The original optimized prompt from an existing work that inspired null-shot prompting is shown in Figure 3. This original prompt is intended for a task of text toxicity classification. The LLM hallucinated during the optimization process and resulted in an optimized prompt that exploits instructions to utilize information from a null section

that was never provided in the prompt. However, it is surprising that the best-performing optimized prompt, which exploits hallucination, performed the best in their paper.

Appendix C Additional Experiment Details

C.1 Datasets and Evaluation

To assess the performance of null-shot prompting, we conduct experiments on 21 models: 10 for the main experiments and 11 additional models for the scaling study. LLMs used for the main experiments are PaLM 2, PaLM 2 (Chat), Gemini 1.0 Pro, Gemini 1.0 Pro (Chat), GPT-3.5 Turbo, GPT-4 Turbo, Claude 2.1, Claude 3 Haiku, Claude 3 Sonnet, and Claude 3 Opus; Pythia-14M, Pythia-31M, Pythia-70M, Pythia-160M, Pythia-410M, Pythia-1B, Pythia-1.4B, Qwen1.5-0.5B-Chat, Qwen1.5-1.8B-Chat, Qwen1.5-4B-Chat, and Qwen1.5-7B-Chat are used for the scaling study. Full details of the setup for each LLM are described in Section C.2. The experiments are performed on six tasks across eight datasets for the main and reasoning experiments, seven topics from MATH benchmark for mathematics problem solving evaluations, and four scenarios for hallucination evaluation from HaluEval. Setup details for each dataset are described as follow. Additional details regarding ablation studies are provided separately in Section 7.

We evaluate the performance of null-shot prompting and zero-shot prompting—the latter is a baseline—on six tasks across ten datasets. The baseline performance is the zero-shot prompting performance of the respective model on the dataset unless stated otherwise in the ablation studies. We assess the increase in performance when using null-shot prompting for each respective model compared to the zero-shot prompting baseline. The rest of this section discusses each dataset and its associated evaluation. For all datasets, we use a standardized question format. The comprehensive list of datasets and their associated details, along with the standardized format for task instructions and inputs in the prompt, as well as output extraction details, while examples of each dataset are presented in Section G.

Arithmetic Reasoning: *AQuA-RAT* (Ling et al., 2017) provides a variety of mathematical questions requiring different strategies to solve.

Classify the “Message” provided below from a chat in an online community into one of the specified categories along with a reason in the specified JSON format in a Markdown code block (between `json` and `code`). Utilize the information from the “Prediction result” section to correctly classify the text.

Categories: “toxic”, “clean”

Message: <message>

Output format:

```
```json
{
 “message”: <message>,
 “reason”: <reason>,
 “predicted”: <predicted category>
}
```
```

Please adhere to the following rules:

1. The content of the “Message” and “predicted category” must not be modified.
2. The “reason” should provide an explanation or justification for the classification, based on the information obtained from the “Prediction result” section.
3. Consider other examples in the “Prediction result” section to guide your classification.

Figure 3: The original optimized prompt, where the sentences inspiring null-shot prompting are highlighted in yellow.

This dataset includes questions as natural language descriptions of expressions and five answer options with one correct option label. Similarly, *GSM8K* (Cobbe et al., 2021) provides diverse grade school math word problems where the label is a number. However, GSM8K does not provide any choices, and models need to generate its own answer.

Commonsense Reasoning: *StrategyQA* (Geva et al., 2021a) provides questions that require implicit reasoning steps, i.e., strategy, to answer the question. It covers a wide range of topics, and the answer to each question is either “YES” or “NO.” On the other hand, *WinoGrande* (Sakaguchi et al., 2021) presents an adversarial Winograd (Levesque et al., 2012) schema challenge for a more robust commonsense reasoning benchmark.

Reading Comprehension: *RACE* (Lai et al., 2017) presents a dataset containing English exams for middle-school (RACE-m) and high-school (RACE-h) students. Some questions in this dataset also require the model to reason, posing a higher challenge for models. We utilize both versions of the dataset in the experiments.

Natural Language Inference and Closed-Book Question Answering: *ANLI* (Nie et al., 2020) provides an adversarial natural language inference (NLI) dataset that is more challenging than standard NLI. We utilize data from the third round of data collection (R3) as our test set, as this round includes not only Wikipedia as the only source but also other media such as news, fiction, and spoken text. Finally, *TriviaQA* (Joshi et al., 2017) is selected to test generalization in typical question answering from model knowledge, i.e., “Does null-shot prompting help the model improve its knowledge-recalling ability?” Due to resource constraints, we sample only 1000 records from the dataset as our test set.

Mathematics Problem Solving and Hallucination Detection: *MATH* (Hendrycks et al., 2021) provides a challenging set of problems across multiple topics in mathematics. We selected this benchmark due to its reputation for evaluating LLMs and its topic classification, which helps us better understand how hallucination relates to mathematics problem-solving abilities. *HaluEval* (Li et al., 2023b) evaluates LLMs’ hallucination detection abilities. For this dataset, as each record contains both hallucinatory and normal options for non-general scenarios, we randomly select one of the options to be a test case, resulting in approximately half of the test cases containing hallucination.

Additional context, e.g., article, context, and hypothesis

Question: {question}

Choices: {choices}

Answer: {output format instruction}

Figure 4: The task instruction and task input format used for the experiments.

| Dataset | Task | Test split | Count | Ans. |
|------------|------|-------------|-------|------|
| AQuA-RAT | AR | test | 254 | MC |
| GSM8K | AR | test | 1319 | Num. |
| StrategyQA | CR | test | 2290 | BC |
| WinoGrande | CR | dev | 1267 | BC |
| RACE-m | RC | middle-test | 1436 | MC |
| RACE-h | RC | high-test | 3498 | MC |
| ANLI | NLI | R3-test | 1200 | MC |
| TriviaQA | CQA | Wikipedia | 1000* | Text |
| MATH | AR | test | 5000 | Num. |
| HaluEval | HD | N/A | 14507 | BC |

Table 10: Details of each dataset. **Test split** shows the split used for evaluations in this study, while **Count** shows the number of included samples. For the **Task**, *AR*: Arithmetic Reasoning, *CR*: Commonsense Reasoning, *RC*: Reading Comprehension, *NLI*: Natural Language Inference, *CQA*: Closed-book Question Answering, and *HD*: Hallucination Detection. The **Ans.** denotes the type of the expected answer, where *BC* represents binary choices, *MC* represents multiple choices, *Num.* represents an arbitrary number answer, and *Text* represents a free-text answer.

*We downsampled TriviaQA to only 1000 records to save budget.

Figure 4 displays the format of task instructions and inputs for the datasets. This format is inspired by the procedure used in the 0CoT prompting study (Kojima et al., 2022). Choices and additional context are only provided in the prompts when applicable. Output format instructions are only provided for the MATH dataset to aid in information extraction. The format instruction is based on the original output label of the dataset. All included datasets are in English. Additional details on the chosen testing set and the number of records are presented in Table 10.

We note that AQuA-RAT, WinoGrande, and TriviaQA are under the Apache License, Version 2.0. GSM8K, StrategyQA, MATH, and HaluEval are under the MIT License. RACE datasets are available for non-commercial research purposes only. ANLI is under the Creative Commons Attribution-NonCommercial 4.0 International License. TriviaQA used in our study is downsampled using the standard random sampling function in Python with a fixed seed of 42. We also note that the datasets

may include names of individuals collected from the internet, i.e., publicly available facts about a person but not in an offensive way. The following list shows the sources of data we used for this study.

- AQuA-RAT: <https://github.com/google-deepmind/AQuA>
- GSM8K: <https://github.com/openai/grade-school-math>
- StrategyQA: https://github.com/google/BIG-bench/tree/main/bigbench/benchmark_tasks/strategyqa
- WinoGrande: <https://winogrande.allenai.org>
- RACE: <https://www.cs.cmu.edu/~glai1/data/race/>
- ANLI: <https://github.com/facebookresearch/anli>
- TriviaQA: <https://nlp.cs.washington.edu/triviaqa/>
- MATH: <https://github.com/hendrycks/math/?tab=MIT-1-ov-file>
- HaluEval: <https://github.com/RUCAIBox/HaluEval>

We also develop output extraction scripts for all datasets. For datasets with choices, we look for patterns of choices in the responses. First, if the response generated from a model is an uppercase character, we treat that as the final answer. For example, if a model responded with “A” and if we have “A” as one of our choices, “A” will be treated as the final answer. In other cases, we first attempt to match a pattern of an uppercase character choice followed by a parenthesis, e.g., “A)”. Then we try to match a pattern of “answer is”, where we treat the first uppercase character choice after the pattern as the final answer. For example, if a response contains “So, the answer is A)”, “A” will be extracted as the final answer.

For all patterns, we attempt to match on the last line of the model’s output first. If unsuccessful, we then try to match the first line of the model output. These heuristics are based on our observation that models are likely to provide the conclusive answer in the last or first lines, as empirically observed in

our pilot study. Failures to match are treated as no answer, as well as in cases where the model returns an empty response.

For datasets without choices, three scenarios are considered. The first scenario is when the answer is a number. In this case, we treat the first number found on the last or first line as the final answer. This is in a similar spirit to a previous study (Kojima et al., 2022). The second scenario is when the answer is free text. In this case, we first lowercase the response and the label. Then we check if the label exists in the response or not. Finally, the third scenario for the MATH dataset, the script tries to match a pattern `\boxed{(.+)}` and extracts any content inside the `{` and `}`.

C.2 LLMs

All LLMs in this study are utilized in a deterministic setup, i.e., we set the sampling temperature to 0 and provide a fixed random seed when applicable. Therefore, we only interact with the model once for each record of the dataset given a prompting approach. Any additional settings, including safety, are left to default. For chat models/pipelines, we always start with an empty context history, with the prompt as the first user message. The ten LLMs included in the main experiments are PaLM 2 (text-bison-001), PaLM 2 (Chat) (chat-bison-001), Gemini 1.0 Pro (gemini-pro) via the generateContent method, Gemini 1.0 Pro (Chat) (gemini-pro via the start_chat method), GPT-3.5 Turbo (gpt-3.5-turbo-1106), GPT-4 Turbo (gpt-4-1106-preview), Claude 2.1 (claude-2.1), Claude 3 Haiku (claude-3-haiku-20240307), Claude 3 Sonnet (claude-3-sonnet-20240229), and Claude 3 Opus (claude-3-opus-20240229). We choose these models for our experiments as they offer APIs to access the models without the need to prepare our own infrastructure for running them. Furthermore, all of these models are relatively large and are utilized in many real-world products and scenarios.

PaLM 2 and PaLM 2 (Chat) serve as a comparison for models from the same family, where one model is possibly a base model and the other one is potentially a chat fine-tuned variant for chat conversations. This could further allow us to assess the effectiveness of the proposed prompting between these two types of LLMs and the importance of chat fine-tuning. Similarly, GPT-3.5 Turbo and GPT-4 Turbo are also chosen to assess these instruction-

aligned models within the same family, where the subsequent version of the same model family is possibly larger in both parameter size and training data. This could provide insights into the effects of scaling models further. We include Gemini 1.0 Pro because its performance is likely positioned between that of GPT-3.5 Turbo and GPT-4 Turbo. Claude models are included as they are well-known for their harmlessness, i.e., being less hallucinatory. All of these aforementioned LLMs are utilized via their respective API-wrapper Python libraries¹.

We also include additional LLMs for scaling studies. These models are from two model families, Pythia and Qwen1.5-Chat. We select Pythia to investigate null-shot prompting with scaling in pre-trained LLMs, while Qwen1.5-Chat represents chat-tuned LLMs. Due to limitations in our computational infrastructure, we are unable to include all LLMs from these suites. All LLMs are utilized via Hugging Face’s transformers² pipelines, i.e., the text-generation pipeline for Pythia models and the conversational pipeline for Qwen1.5-Chat. We provide a list of Pythia and Qwen1.5-Chat models selected and included in this study as follows:

- EleutherAI/pythia-14m: <https://huggingface.co/EleutherAI/pythia-14m>
- EleutherAI/pythia-31m: <https://huggingface.co/EleutherAI/pythia-31m>
- EleutherAI/pythia-70m: <https://huggingface.co/EleutherAI/pythia-70m>
- EleutherAI/pythia-160m: <https://huggingface.co/EleutherAI/pythia-160m>
- EleutherAI/pythia-410m: <https://huggingface.co/EleutherAI/pythia-410m>
- EleutherAI/pythia-1b: <https://huggingface.co/EleutherAI/pythia-1b>

¹GPT-3.5 Turbo and GPT-4 Turbo: <https://github.com/openai/openai-python>

PaLM 2, PaLM 2 (Chat), Gemini 1.0 Pro, and Gemini 1.0 Pro (Chat): <https://github.com/google/generative-ai-python>

Claude 2.1, Claude 3 Haiku, Claude 3 Sonnet, and Claude 3 Opus: <https://github.com/anthropics/anthropic-sdk-python>

²<https://github.com/huggingface/transformers/>

- EleutherAI/pythia-1.4b: <https://huggingface.co/EleutherAI/pythia-1.4b>
- Qwen/Qwen1.5-0.5B-Chat: <https://huggingface.co/Qwen/Qwen1.5-0.5B-Chat>
- Qwen/Qwen1.5-1.8B-Chat: <https://huggingface.co/Qwen/Qwen1.5-1.8B-Chat>
- Qwen/Qwen1.5-4B-Chat: <https://huggingface.co/Qwen/Qwen1.5-4B-Chat>
- Qwen/Qwen1.5-7B-Chat: <https://huggingface.co/Qwen/Qwen1.5-7B-Chat>

We note that all models used in our study through APIs are subject to the terms and conditions of API providers, which allow non-commercial research purposes in our study. Pythia models are subject to the Apache License Version 2.0, while Qwen1.5-Chat models are subject to the Tongyi Qianwen License Agreement. Both licenses for Pythia and Qwen1.5-Chat permit research use cases.

For Pythia and Qwen1.5-Chat models, we run them on two computers, one with an NVIDIA A100 80GB GPU and another one with an NVIDIA L40S GPU. The total GPU hours of all experiments utilizing these models on both computers are 3184.5 hours. On the other hand, the total processing time, including network latency, for all LLMs interacting via APIs is 1409.22 hours. In total, this paper consumed 4593.72 hours of processing time.

C.3 Null-Shot CoT Phrase

We present the \emptyset CoT phrase used in Section 6. We devised this phrase by taking the null-shot phrase and adding the phrase “step-by-step” from 0CoT prompting (Kojima et al., 2022). The phrase is shown in Figure 5.

C.4 Components in the Null-Shot Phrase

This section shows each component of the null-shot phrase. The null-shot phrase consists of three components, as illustrated in Figure 6. Each component is a distinct variant of the null-shot phrase and is used in place of the null-shot phrase for the experiment described in Section 7.2 to assess the impact of each component. Figures 7, 8, and 9 show the first, second, and third variants (components), respectively.

Null-Shot CoT Phrase

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task step-by-step.

Figure 5: The null-shot CoT phrase instructs LLMs to look into and utilize information from the null section and perform the task step-by-step. The added CoT part is highlighted in yellow.

Null-Shot Phrase: Components

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

Figure 6: The null-shot phrase with the first component highlighted in yellow and the second component highlighted in green.

Null-Shot Phrase: First Variant (v1)

Utilize examples and information from the “Examples” section to perform the following task.

Figure 7: The first variant of null-shot phrase with the first component removed.

Null-Shot Phrase: Second Variant (v2)

Look at examples in the “Examples” section and perform the following task.

Figure 8: The second variant of null-shot phrase with the second component removed.

Null-Shot Phrase: Third Variant (v3)

Perform the following task as demonstrated in the “Examples” section.

Figure 9: The third variant of null-shot phrase with both components removed.

Appendix D Additional Experiment Results

This section presents the absolute results from the experiments in the main body of this paper. For all tables, OS represents zero-shot prompting, \emptyset S represents null-shot prompting, OCoT represents zero-shot chain-of-thought prompting, and \emptyset CoT represents null-shot chain-of-thought prompting. Numbers in green represent cases when performance is improved compared to the baseline, while numbers in **bold** show the best performance within the same task for a particular model, regardless of the PE approaches. The absolute results from Sections 3, 4, and 5 are shown in Tables 11, 12, and 13, respectively. The absolute results of Section 6 are shown in Tables 14 and 15. For the ablation studies discussed in Section 7, the absolute results for the optimal position of the null-shot phrase experiments are presented in Tables 16 and 17. Similarly, Tables 18 and 19 present the absolute results of the study conducted to determine the impact of each component in the null-shot phrase.

Appendix E Scaling Studies

We perform scaling studies to better understand how the scale of an LLM affects its ability to be hallucinatory by null-shot prompting and cause performance changes. We select two LLM families, Pythia for pre-trained models covering the range from 14M to 1.8B parameters, and Qwen1.5-Chat for chat-tuned LLMs covering the range from 0.5B to 7B parameters. Due to our computation infrastructure constraints, we are unable to evaluate all LLMs in the suites. We use all non-hallucination-detection datasets from the experiments of our study and compare performance between zero-shot and null-shot prompting. We exclude reasoning variants as they introduce additional factors to consider, i.e., the ability to reason. Results of Pythia models are shown in Figure 10 for general evaluation tasks and Figure 11 for the MATH benchmark. Results of Qwen1.5-Chat models are shown in Figure 12 and Figure 13 for the general and MATH benchmarks, respectively.

Given the results of Pythia models on general tasks, we observe that in all cases, the performance of both promptings is scaling together, except for RACE-m, a reading comprehension task. In RACE-m, we notice that there is a range, from 410M to 1B parameters, where null-shot prompting consistently performs better than zero-shot prompting. We con-

| Model | AQuA | | GSM8K | | StrategyQA | | WinoGrande | |
|-----------------------|-------|-------|-------|-------|------------|-------|------------|-------|
| | 0S | ∅S | 0S | ∅S | 0S | ∅S | 0S | ∅S |
| PaLM 2 | 29.13 | 28.35 | 14.78 | 16.45 | 59.83 | 66.38 | 72.69 | 80.03 |
| PaLM 2 (Chat) | 14.96 | 15.75 | 53.9 | 55.12 | 57.73 | 58.69 | 56.59 | 60.54 |
| Gemini 1.0 Pro | 25.59 | 35.43 | 51.55 | 66.49 | 67.03 | 50.66 | 63.85 | 62.98 |
| Gemini 1.0 Pro (Chat) | 25.59 | 37.01 | 52.39 | 67.02 | 67.6 | 50.44 | 63.69 | 62.98 |
| GPT-3.5 Turbo | 42.91 | 57.48 | 54.89 | 63.23 | 64.02 | 66.03 | 59.98 | 58.88 |
| GPT-4 Turbo | 75.98 | 75.59 | 74.3 | 73.16 | 74.85 | 61.83 | 73.48 | 55.8 |
| Claude 2.1 | 64.96 | 57.48 | 78.92 | 63.91 | 40.44 | 11.79 | 2.21 | 0.24 |
| Claude 3 Haiku | 63.39 | 58.66 | 68.01 | 66.26 | 43.71 | 29.04 | 60.54 | 40.33 |
| Claude 3 Sonnet | 61.02 | 55.91 | 64.67 | 59.14 | 55.76 | 22.49 | 59.27 | 32.2 |
| Claude 3 Opus | 68.5 | 56.3 | 72.48 | 56.1 | 69.04 | 4.93 | 70.96 | 0.63 |

| Model | RACE-m | | RACE-h | | ANLI | | TriviaQA | |
|-----------------------|--------|-------|--------|-------|-------|-------|----------|------|
| | 0S | ∅S | 0S | ∅S | 0S | ∅S | 0S | ∅S |
| PaLM 2 | 82.66 | 84.19 | 71.56 | 74.16 | 49.17 | 50.5 | 64.2 | 68.7 |
| PaLM 2 (Chat) | 73.54 | 74.3 | 67.04 | 67.5 | 42.67 | 43.33 | 70.2 | 70.1 |
| Gemini 1.0 Pro | 83.01 | 84.61 | 77.82 | 79.47 | 50.58 | 51.67 | 70.2 | 25.3 |
| Gemini 1.0 Pro (Chat) | 84.61 | 85.24 | 79.1 | 80.39 | 51.08 | 51.92 | 70.5 | 25.4 |
| GPT-3.5 Turbo | 85.38 | 83.84 | 81.73 | 80.76 | 48.42 | 46.67 | 81 | 82 |
| GPT-4 Turbo | 92.97 | 93.25 | 88.59 | 88.97 | 64.17 | 64 | 85.4 | 84.6 |
| Claude 2.1 | 50 | 49.51 | 39.88 | 41.05 | 33.58 | 26.42 | 73.4 | 20 |
| Claude 3 Haiku | 75.91 | 68.8 | 59.06 | 53.89 | 28.58 | 38.42 | 78.5 | 64.5 |
| Claude 3 Sonnet | 73.82 | 59.89 | 67.84 | 56.69 | 53.33 | 42.67 | 78.7 | 31.9 |
| Claude 3 Opus | 77.65 | 81.69 | 67.01 | 72.3 | 60.67 | 54.42 | 85.2 | 20.7 |

Table 11: This table shows the absolute performance from the main experiments between zero-shot prompting (baseline) and null-shot prompting. The maximum possible value for each cell is 100 (accuracy percentage).

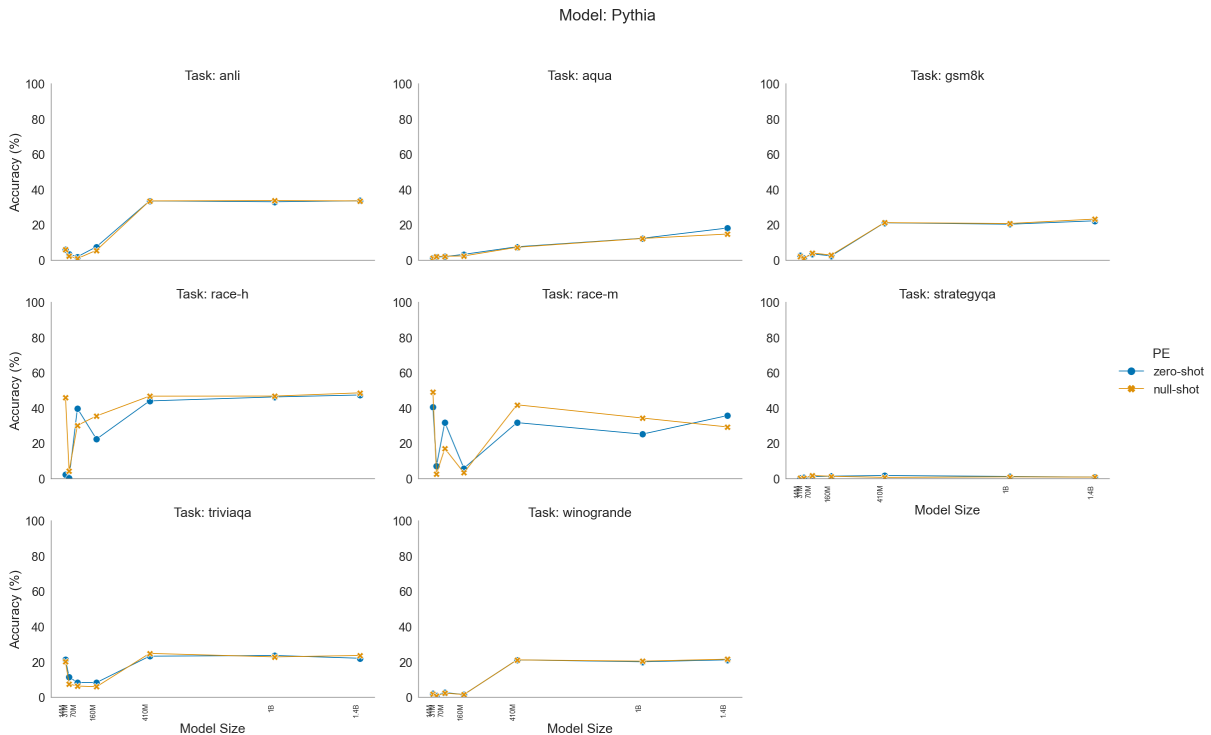


Figure 10: This figure shows performance of zero-shot and null-shot prompts of Pythia models using general evaluation tasks.

| Model | Prealgebra | | Algebra | | Num. Th. | | Count. & Prob. | |
|-----------------------|------------|-------|---------|-------|----------|-------|----------------|-------|
| | 0S | ∅S | 0S | ∅S | 0S | ∅S | 0S | ∅S |
| PaLM 2 | 17.68 | 17.22 | 16.85 | 16.26 | 11.85 | 11.67 | 12.87 | 10.97 |
| PaLM 2 (Chat) | 7 | 15.15 | 3.54 | 12.3 | 2.22 | 5.93 | 4.85 | 10.34 |
| Gemini 1.0 Pro | 31.92 | 34.9 | 24.94 | 26.96 | 16.11 | 16.11 | 16.24 | 16.88 |
| Gemini 1.0 Pro (Chat) | 31.92 | 34.56 | 24.85 | 27.04 | 16.11 | 16.11 | 16.24 | 16.67 |
| GPT-3.5 Turbo | 42.14 | 54.42 | 38.92 | 55.43 | 23.89 | 35.56 | 25.74 | 31.65 |
| GPT-4 Turbo | 72.33 | 71.76 | 70.85 | 67.06 | 61.67 | 61.11 | 51.9 | 52.53 |
| Claude 2.1 | 48.45 | 44.32 | 44.06 | 40.78 | 23.7 | 21.85 | 22.15 | 21.31 |
| Claude 3 Haiku | 55.8 | 53.04 | 48.36 | 47.77 | 32.22 | 34.07 | 27 | 27.64 |
| Claude 3 Sonnet | 63.61 | 64.18 | 52.82 | 53.66 | 39.07 | 36.11 | 34.18 | 32.91 |
| Claude 3 Opus | 73.59 | 72.22 | 75.15 | 73.38 | 67.41 | 62.41 | 55.91 | 50.21 |

| Model | Geometry | | Int. Algebra | | Precalculus | |
|-----------------------|----------|-------|--------------|-------|-------------|-------|
| | 0S | ∅S | 0S | ∅S | 0S | ∅S |
| PaLM 2 | 14.2 | 13.57 | 11.3 | 11.85 | 12.45 | 12.45 |
| PaLM 2 (Chat) | 5.64 | 8.35 | 4.1 | 7.31 | 3.3 | 6.04 |
| Gemini 1.0 Pro | 18.79 | 19.83 | 14.84 | 14.17 | 19.41 | 18.13 |
| Gemini 1.0 Pro (Chat) | 18.79 | 19.83 | 14.84 | 14.29 | 19.6 | 18.13 |
| GPT-3.5 Turbo | 24.22 | 29.23 | 18.05 | 21.04 | 21.79 | 22.16 |
| GPT-4 Turbo | 46.76 | 42.8 | 35.33 | 33.89 | 33.33 | 29.49 |
| Claude 2.1 | 22.96 | 21.5 | 12.51 | 12.62 | 13.19 | 14.65 |
| Claude 3 Haiku | 25.89 | 24.63 | 14.73 | 15.17 | 18.86 | 19.6 |
| Claude 3 Sonnet | 32.36 | 28.18 | 17.05 | 16.94 | 19.6 | 18.86 |
| Claude 3 Opus | 45.72 | 42.38 | 30.9 | 32 | 34.98 | 32.97 |

Table 12: This table shows the absolute performance from the MATH evaluation between zero-shot prompting (baseline) and null-shot prompting.

| Model | General | | Dialogue | | QA | | Sum. | |
|-----------------------|---------|-------|----------|-------|-------|-------|-------|-------|
| | 0S | ∅S | 0S | ∅S | 0S | ∅S | 0S | ∅S |
| PaLM 2 | 76.86 | 78.1 | 66.14 | 67.62 | 61.46 | 62.07 | 37.61 | 40.78 |
| PaLM 2 (Chat) | 8.32 | 10.45 | 8.78 | 8.92 | 11.7 | 19.03 | 0.62 | 1.5 |
| Gemini 1.0 Pro | 81.74 | 81.78 | 77.72 | 77.94 | 62.54 | 63.46 | 65.54 | 65.41 |
| Gemini 1.0 Pro (Chat) | 81.74 | 81.78 | 76.66 | 76.85 | 61.98 | 62.87 | 66.11 | 66.03 |
| GPT-3.5 Turbo | 79.96 | 80.19 | 59.99 | 62.89 | 38.52 | 42.15 | 35.35 | 34.6 |
| GPT-4 Turbo | 81.12 | 81.1 | 75.94 | 76.28 | 71.76 | 68.7 | 75.56 | 75.41 |
| Claude 2.1 | 81.36 | 81.21 | 61.68 | 55.12 | 45.3 | 39.26 | 53.39 | 49.33 |
| Claude 3 Haiku | 80.47 | 79.72 | 67.04 | 68.41 | 60.68 | 58.41 | 50.39 | 53.54 |
| Claude 3 Sonnet | 81.67 | 81.78 | 71.13 | 74.57 | 52.24 | 42.59 | 53.83 | 55.65 |
| Claude 3 Opus | 81.58 | 81.27 | 68.32 | 62.96 | 65.9 | 66.54 | 68.3 | 68.33 |

Table 13: This table shows the absolute performance from the HaluEval dataset focusing on evaluating hallucination detection abilities between zero-shot prompting (baseline) and null-shot prompting.

| Model | AQuA | | GSM8K | | StrategyQA | | WinoGrande | |
|-----------------------|-------|--------------|--------------|-------|--------------|--------------|------------|--------------|
| | 0CoT | ∅CoT | 0CoT | ∅CoT | 0CoT | ∅CoT | 0CoT | ∅CoT |
| PaLM 2 | 35.43 | 16.14 | 60.2 | 43.52 | 62.71 | 59.34 | 63.93 | 76.01 |
| PaLM 2 (Chat) | 13.39 | 12.6 | 58.23 | 53.83 | 52.4 | 60.13 | 59.83 | 57.14 |
| Gemini 1.0 Pro | 46.46 | 50.39 | 69.07 | 62.17 | 66.42 | 1.05 | 62.83 | 0.24 |
| Gemini 1.0 Pro (Chat) | 48.82 | 52.76 | 70.89 | 62.62 | 66.42 | 1.05 | 62.43 | 0.24 |
| GPT-3.5 Turbo | 57.48 | 55.51 | 66.41 | 63.76 | 66.11 | 57.38 | 51.14 | 45.54 |
| GPT-4 Turbo | 75.59 | 77.17 | 74.45 | 73.62 | 63.45 | 61.97 | 51.93 | 64.8 |
| Claude 2.1 | 64.57 | 62.99 | 78.24 | 75.66 | 43.97 | 21.57 | 13.65 | 1.66 |
| Claude 3 Haiku | 62.99 | 61.02 | 68.69 | 67.32 | 28.86 | 28.38 | 45.54 | 23.05 |
| Claude 3 Sonnet | 59.45 | 50.79 | 63.38 | 62.32 | 52.93 | 8.56 | 46.17 | 8.52 |
| Claude 3 Opus | 63.78 | 57.48 | 70.28 | 69.52 | 63.1 | 26.94 | 60.3 | 29.04 |

| Model | RACE-m | | RACE-h | | ANLI | | TriviaQA | |
|-----------------------|--------------|--------------|--------------|--------------|-------|--------------|-------------|-------------|
| | 0CoT | ∅CoT | 0CoT | ∅CoT | 0CoT | ∅CoT | 0CoT | ∅CoT |
| PaLM 2 | 71.03 | 82.52 | 60.12 | 72.24 | 49.17 | 48.75 | 66.7 | 63.7 |
| PaLM 2 (Chat) | 71.59 | 71.94 | 65.04 | 65.55 | 43.5 | 42.42 | 70.6 | 69.1 |
| Gemini 1.0 Pro | 83.15 | 80.64 | 78.47 | 77.64 | 48.33 | 44.92 | 61.6 | 0.9 |
| Gemini 1.0 Pro (Chat) | 84.05 | 82.1 | 78.36 | 77.79 | 48.92 | 45 | 62 | 0.9 |
| GPT-3.5 Turbo | 83.36 | 80.57 | 77.67 | 77.79 | 42.25 | 22.42 | 80.3 | 80.9 |
| GPT-4 Turbo | 71.59 | 67.55 | 61.75 | 56.46 | 52.17 | 47.58 | 86.2 | 85.3 |
| Claude 2.1 | 46.17 | 49.86 | 37.68 | 37.42 | 39.67 | 32.92 | 75.9 | 44.3 |
| Claude 3 Haiku | 45.26 | 65.39 | 39.88 | 54.4 | 31.58 | 30.33 | 77 | 65 |
| Claude 3 Sonnet | 63.58 | 72.77 | 50.57 | 58.2 | 40.75 | 47.5 | 75.4 | 40 |
| Claude 3 Opus | 85.45 | 81.27 | 77.5 | 70.9 | 57.08 | 59.08 | 85.5 | 65.6 |

Table 14: This table shows the absolute performance of various datasets using 0CoT prompting (baseline) and ∅CoT prompting.

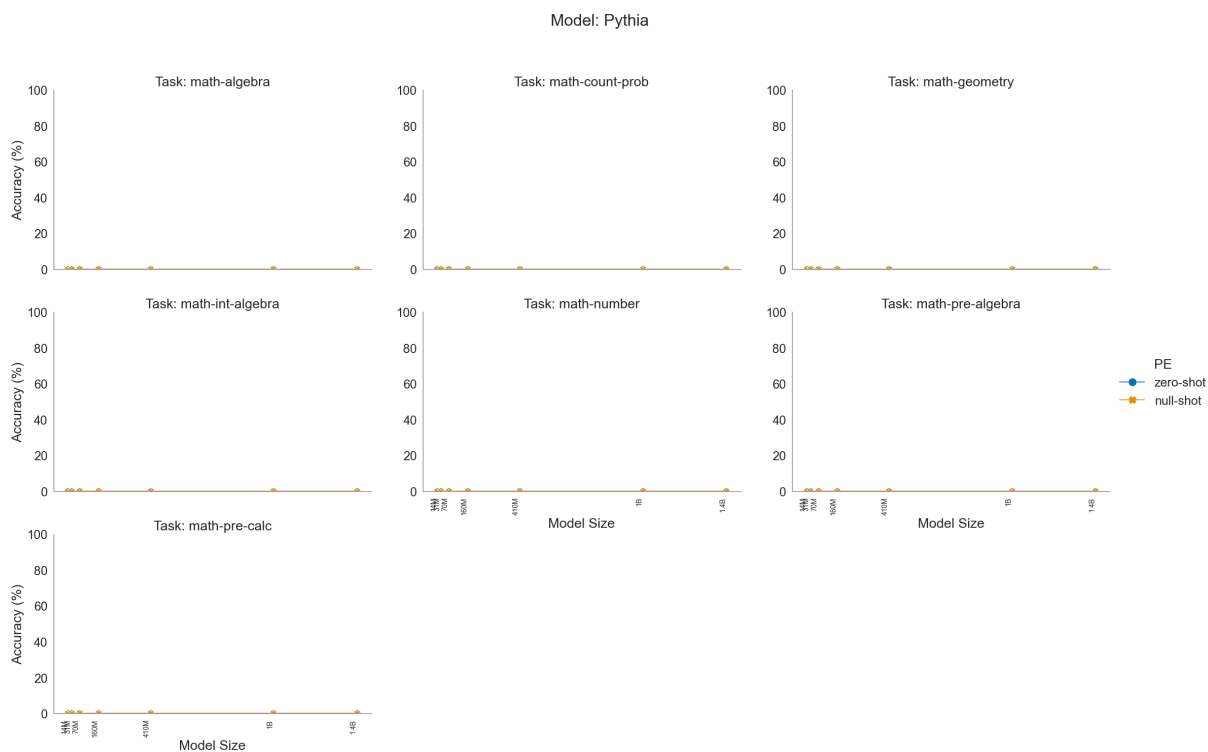


Figure 11: This figure shows performance of zero-shot and null-shot promptings of Pythia models using the MATH benchmark.

| Model | Prealgebra | | Algebra | | Num. Th. | | Count. & Prob. | |
|-----------------------|------------|-------|---------|-------|----------|-------|----------------|-------|
| | 0CoT | ∅CoT | 0CoT | ∅CoT | 0CoT | ∅CoT | 0CoT | ∅CoT |
| PaLM 2 | 20.55 | 17.34 | 16.85 | 16.6 | 10.56 | 10.37 | 12.45 | 13.08 |
| PaLM 2 (Chat) | 6.77 | 9.41 | 4.21 | 7.08 | 2.78 | 3.89 | 4.22 | 5.7 |
| Gemini 1.0 Pro | 39.61 | 47.99 | 33.36 | 42.97 | 16.67 | 19.81 | 17.93 | 21.73 |
| Gemini 1.0 Pro (Chat) | 39.61 | 47.76 | 33.19 | 42.8 | 16.67 | 18.89 | 17.72 | 21.73 |
| GPT-3.5 Turbo | 59.47 | 58.55 | 57.79 | 57.54 | 35.93 | 37.78 | 33.12 | 35.23 |
| GPT-4 Turbo | 72.79 | 70.95 | 70.26 | 71.36 | 63.7 | 62.96 | 52.32 | 52.11 |
| Claude 2.1 | 47.53 | 50.29 | 41.62 | 41.7 | 24.07 | 26.67 | 21.94 | 22.15 |
| Claude 3 Haiku | 57.18 | 56.03 | 49.12 | 48.61 | 34.07 | 35 | 29.11 | 27.43 |
| Claude 3 Sonnet | 62.92 | 11.37 | 54.68 | 10.11 | 39.44 | 14.07 | 36.71 | 8.23 |
| Claude 3 Opus | 74.28 | 75.66 | 74.98 | 74.14 | 66.11 | 65 | 52.74 | 54.01 |

| Model | Geometry | | Int. Algebra | | Precalculus | |
|-----------------------|----------|-------|--------------|-------|-------------|-------|
| | 0CoT | ∅CoT | 0CoT | ∅CoT | 0CoT | ∅CoT |
| PaLM 2 | 11.69 | 13.15 | 10.41 | 11.07 | 12.45 | 12.09 |
| PaLM 2 (Chat) | 5.43 | 7.31 | 2.88 | 3.99 | 3.3 | 3.48 |
| Gemini 1.0 Pro | 19.62 | 22.34 | 14.06 | 13.07 | 17.4 | 17.95 |
| Gemini 1.0 Pro (Chat) | 19.42 | 22.96 | 13.84 | 12.96 | 17.4 | 18.13 |
| GPT-3.5 Turbo | 28.81 | 31.94 | 19.71 | 19.71 | 23.99 | 22.53 |
| GPT-4 Turbo | 44.47 | 44.47 | 34 | 35.66 | 29.49 | 29.3 |
| Claude 2.1 | 20.04 | 21.92 | 13.18 | 12.18 | 15.57 | 17.22 |
| Claude 3 Haiku | 22.55 | 24.84 | 15.73 | 13.29 | 18.13 | 21.61 |
| Claude 3 Sonnet | 30.9 | 6.89 | 16.39 | 5.43 | 17.95 | 9.16 |
| Claude 3 Opus | 43.22 | 44.47 | 32.23 | 32.56 | 35.9 | 35.35 |

Table 15: This table shows the absolute performance of the MATH dataset using 0CoT prompting (baseline) and ∅CoT prompting.

| Approach | AQuA | GSM8K | StrategyQA | WinoGrande | RACE-m | RACE-h | ANLI | TriviaQA |
|-------------------|-------|-------|------------|------------|--------|--------|-------|----------|
| Zero-Shot | 42.91 | 54.89 | 64.02 | 59.98 | 85.38 | 81.73 | 48.42 | 81 |
| Null-Shot | 57.48 | 63.23 | 66.03 | 58.88 | 83.84 | 80.76 | 46.67 | 82 |
| Null-Shot (After) | 55.91 | 65.43 | 60.04 | 18.55 | 81.27 | 77.96 | 25.92 | 78.5 |

Table 16: This table shows the absolute results of an ablation study conducted to determine the optimal position for the null-shot phrase across the datasets utilized in the main experiments.

| Model | Prealgebra | Algebra | Num. Th. | Count. & Prob. | Geometry | Int. Algebra | Precalculus |
|-------------------|------------|---------|----------|----------------|----------|--------------|-------------|
| Zero-Shot | 42.14 | 38.92 | 23.89 | 25.74 | 24.22 | 18.05 | 21.79 |
| Null-Shot | 54.42 | 55.43 | 35.56 | 31.65 | 29.23 | 21.04 | 22.16 |
| Null-Shot (After) | 48.68 | 46.76 | 27.04 | 26.16 | 28.18 | 18.83 | 21.61 |

Table 17: This table shows the absolute results of an ablation study conducted to determine the optimal position for the null-shot phrase using the MATH dataset.

| Approach | AQuA | GSM8K | StrategyQA | WinoGrande | RACE-m | RACE-h | ANLI | TriviaQA |
|--------------|-------|-------|------------|------------|--------|--------|-------|----------|
| Zero-Shot | 42.91 | 54.89 | 64.02 | 59.98 | 85.38 | 81.73 | 48.42 | 81 |
| Null-Shot | 57.48 | 63.23 | 66.03 | 58.88 | 83.84 | 80.76 | 46.67 | 82 |
| Null-Shot V1 | 58.66 | 64.14 | 65.76 | 57.62 | 83.57 | 80.3 | 45.33 | 81.5 |
| Null-Shot V2 | 47.24 | 59.82 | 66.94 | 61.09 | 84.4 | 81.07 | 48.67 | 80.7 |
| Null-Shot V3 | 54.72 | 63.61 | 64.8 | 54.7 | 84.12 | 80.56 | 47.42 | 80.8 |

Table 18: This table shows the absolute results of an ablation study conducted to determine the impact of each component in the null-shot phrase, using the datasets from the main experiment.

| Model | Prealgebra | Algebra | Num. Th. | Count. & Prob. | Geometry | Int. Algebra | Precalculus |
|--------------|------------|---------|----------|----------------|----------|--------------|-------------|
| Zero-Shot | 42.14 | 38.92 | 23.89 | 25.74 | 24.22 | 18.05 | 21.79 |
| Null-Shot | 54.42 | 55.43 | 35.56 | 31.65 | 29.23 | 21.04 | 22.16 |
| Null-Shot V1 | 55.68 | 54.42 | 34.63 | 31.22 | 30.06 | 19.27 | 24.36 |
| Null-Shot V2 | 47.42 | 45.41 | 27.78 | 26.37 | 28.6 | 17.94 | 20.51 |
| Null-Shot V3 | 52.7 | 51.98 | 34.63 | 29.96 | 30.27 | 18.83 | 22.89 |

Table 19: This table shows the absolute results of an ablation study conducted to determine the impact of each component in the null-shot phrase, using the MATH dataset.

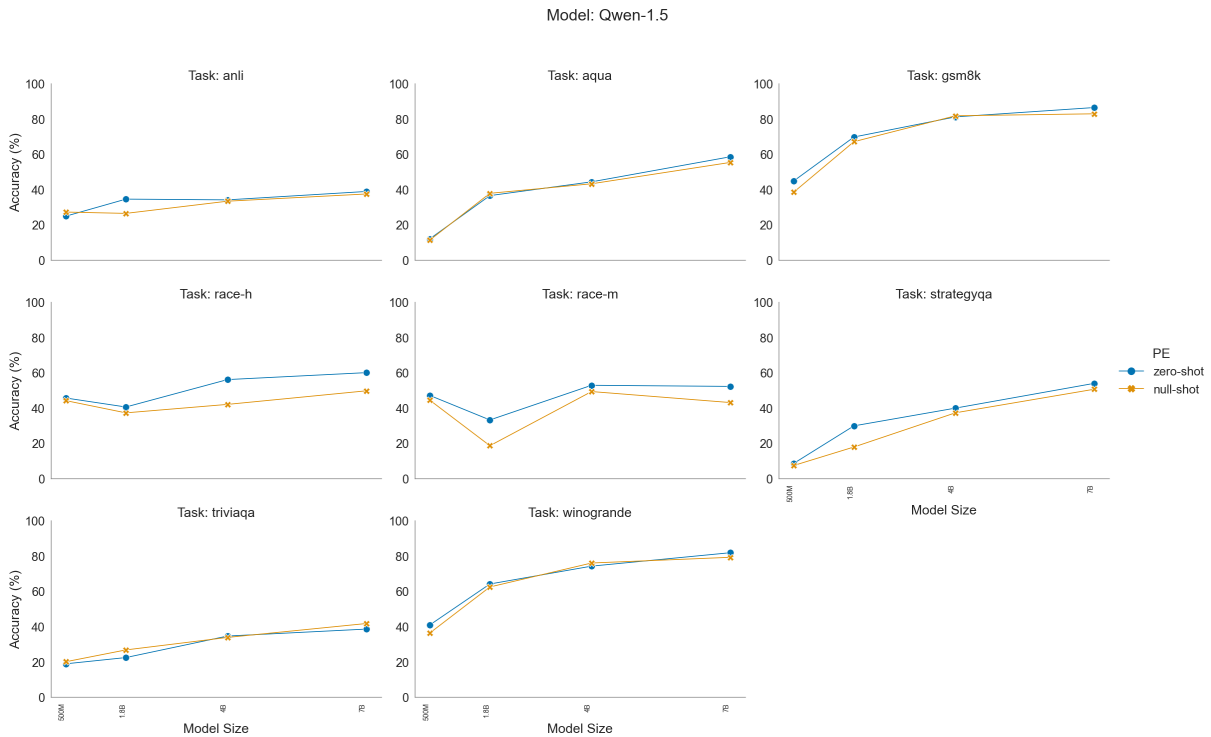


Figure 12: This figure shows performance of zero-shot and null-shot promptings of Qwen1.5-Chat models using general evaluation tasks.

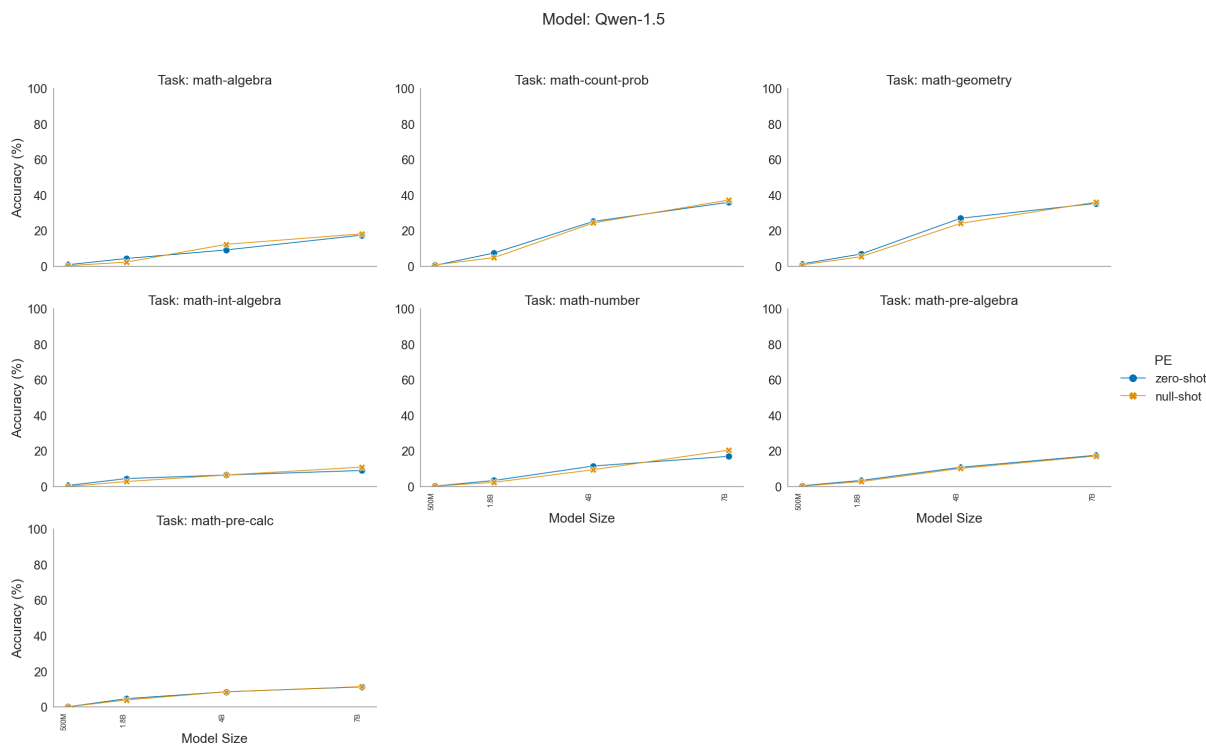


Figure 13: This figure shows performance of zero-shot and null-shot prompts of Qwen1.5-Chat models using the MATH benchmark.

jecture the same reason as previously discussed in Section 3 on why null-shot prompting is effective for long-context tasks. We exclude a discussion of Pythia models’ performances from the MATH benchmark, as their performance is consistently subpar, i.e., these LLMs are not at the scale to perform well in mathematics problem-solving. Overall, for Pythia models with model sizes ranging from 14M to 1.8B, null-shot prompting is mostly equivalent to zero-shot prompting.

Continuing with chat-tuned models like Qwen1.5-Chat, we observe overall better performance, which likely results from both the scale and tuning. However, we also see a similar trend with Pythia, where zero-shot and null-shot scaling up together, except for reading comprehension tasks. In reading comprehension tasks, null-shot prompting consistently performs worse than zero-shot prompting. Given the range of Qwen1.5-Chat models that we include in this study, at a maximum of 7B parameters, and the observed results, we conclude that null-shot prompting is an ability that only exists within larger models and not in smaller models. The exact point at which this ability, i.e., the divergence of zero-shot and null-shot prompting performance, emerges requires further study covering larger models. This

concludes that null-shot prompting is an emergent ability (Wei et al., 2022a), an ability which does not exist in smaller models, but emerges once reaching a certain point. We also note another limitation where we only evaluate decoder-only Transformer-based LLMs and not sparse mixture-of-experts-(SMoE)-based models (Shazeer et al., 2017), which we previously noted have diverged behaviors. Therefore, these conclusions only apply to decoder-only Transformer-based LLMs and require further study for other kinds of architectures, including SMoE.

Appendix F Additional Analyses

We present three additional analyses to further enhance the depth of our investigations. First, we investigate how null-shot and \emptyset CoT promptings affect the results and causes the improvements or degradations. This investigation is presented in Section F.1. Next, we perform error analysis, looking into cases where we get empty responses as a result of built-in safety mechanisms into the APIs used for interacting with the LLMs on the general tasks from Section 3. This behavior only exist with LLMs from Google, namely, PaLM 2, PaLM 2 (Chat), Gemini 1.0 Pro, and Gemini 1.0 Pro (Chat). We discuss such behaviors in Section F.2.

F.1 Effects of Null-Shot Prompting

In this section, we present the effects of null-shot and \emptyset CoT promptings compared to the baseline. We hypothesize that in case the null-shot prompting and its reasoning variants perform well, i.e., observing improvements, there should be a higher percentage of changes from incorrect in the baseline to correct in null-shot approaches. We compute average percentages of three cases, i.e., incorrect in the baseline to correct in null-shot approaches, correct in baselines and incorrect in null-shot approaches, and finally no changes. The results are presented in Table 20.

From the table, we observe that our hypothesis is correct and in cases where a combination of LLM and task sees improvements in performance when using null-shot prompting or its reasoning variant, there is a higher percentage of having an incorrect answer from baseline prompting than vice versa. We also note that in the majority of cases, null-shot prompting and its variants do not change the final outcomes. This indicates that LLMs, most of the time, treat null-shot and \emptyset CoT promptings as not different from such baseline. However, when it does, the improvements in performance of \emptyset CoT prompting are more significant than null-shot prompting, at 10.93% vs 7.6%. However, null-shot prompting exhibits superiority in having fewer percentages of generating incorrect answers where the baseline is able to provide a correct answer.

It is important to also note that in cases where the correct answer from the baseline becomes incorrect in the null-shot approaches and no change where incorrect stays as incorrect, this is also affected by the fact that there are times where LLMs refuse to perform the task as they cannot detect the instructed null section. These cases are affected by the abilities of LLMs in detecting hallucination. We discuss more on this behavior in Section F.3.

F.2 Error Analysis

We investigate failure cases of the main experiments. In particular, we focus on cases where responses are an empty string due to getting blocked from safety mechanisms built into these models or their APIs; we leave all safety settings to default to imitate real-world scenarios of API usages. We note that these mechanisms, as of writing, only exist within the models used through APIs served by

Google which are the PaLM 2³ models and Gemini 1.0 Pro⁴ models. Our further investigations also validate that other LLMs do not have this behavior. Table 21 presents cases where the aforementioned models from Google output empty responses due to being blocked by the security mechanisms.

We observe interesting results where the utilization of null-shot or \emptyset CoT prompting decreases the effectiveness of the built-in safety mechanisms in the majority of cases. As shown in Table 21, the red color highlights the decrease in the numbers of empty responses across datasets when eliciting through null-shot or \emptyset CoT prompting. We find that both prompting can decrease empty responses by 25.02% on average – 44.77% and 5.26% on average when using null-shot and \emptyset CoT prompting, respectively. We posit that the prompting distracts the models and deviates the models from usual patterns. Thus, the prompting decreases the effectiveness of the built-in safety measures. The results show a possibility to utilize both PE techniques to circumvent the safety mechanisms built into the models in a similar spirit to jailbreaking through prompting (Chao et al., 2023; Shen et al., 2024), but in our case, the safety breach is potentially at the API pipeline level. We conjecture that measures deployed during training, such as safe instruction-tuning, may not generalize enough to safeguard against all cases, in particular, when the models get distracted with hallucination-elicited prompts.

Comparing null-shot and \emptyset CoT prompting, we notice that null-shot prompting is more effective in breaking built-in safety measures, which is likely attributed to the fact that \emptyset CoT prompting induces reasoning during decoding. Therefore, we posit the same reasons for the reduced effectiveness of \emptyset CoT prompting in general; reasoning reduces the chances of hallucination. In addition, we observe that PaLM 2 (Chat) generates fewer empty responses than PaLM 2 in general. In contrast, Gemini 1.0 Pro is more consistent across text and chat generation. This observation may result from the fact that PaLM 2 and PaLM 2 (Chat) are two different models, while both Gemini 1.0 Pro variants are based on the same model. Nevertheless, we cannot confirm this fact due to a lack of public report.

³<https://cloud.google.com/vertex-ai/docs/generative-ai/configure-safety-attributes-palm>

⁴https://ai.google.dev/docs/safety_setting_gemini

| | Incorrect \rightarrow Correct (%) | Correct \rightarrow Incorrect (%) | No Change (%) |
|--|-------------------------------------|-------------------------------------|---------------|
| Zero-Shot vs Null-Shot | | | |
| All | 6.51 | 9.41 | 84.08 |
| Improved only | 7.6 | 4.47 | 87.93 |
| Not improved only | 5.49 | 14.05 | 80.46 |
| 0CoT vs \emptysetCoT | | | |
| All | 8.08 | 13.74 | 78.18 |
| Improved only | 10.93 | 7 | 82.07 |
| Not improved only | 6.23 | 18.11 | 75.66 |

Table 20: This table presents average percentages of the direction of change for each record after using the prompting approach on the right-hand side versus the baseline on the left-hand side. *Incorrect \rightarrow Correct* represents cases where baseline prompting is incorrect, and intervention prompting is correct. *Correct \rightarrow Incorrect* represents the opposite case, and *No Change* represents cases where there is no change, i.e., stay correct or stay incorrect. *All* represents percentages computed from all scenarios. *Improved only* represents cases where the calculation is made only in a combination where a combination of LLMs and task outperforms the baseline, and *Not improved only* is vice versa. Numbers in **bold** show cases where there is a higher percentage of the intervention prompting approach correcting mistakes of baseline prompting than vice versa.

| Model | AQuA | | | | GSM8K | | | |
|-----------------------|-------------|---------------|--------------|-----------------|--------------|---------------|--------------|-----------------|
| | 0S | \emptyset S | 0CoT | \emptyset CoT | 0S | \emptyset S | 0CoT | \emptyset CoT |
| PaLM 2 | 2.36% (6) | 0% (0) | 2.76% (7) | 2.36% (6) | 4.02% (53) | 0.38% (5) | 1.14% (15) | 0.91% (12) |
| PaLM 2 (Chat) | 0.39% (1) | 0% (0) | 0% (0) | 0% (0) | 0.23% (3) | 0.3% (4) | 0.23% (3) | 0.23% (3) |
| Gemini 1.0 Pro | 1.18% (3) | 0.39% (1) | 0% (0) | 0.39% (1) | 3.26% (43) | 0.53% (7) | 1.06% (14) | 0.45% (6) |
| Gemini 1.0 Pro (Chat) | 0.39% (1) | 0.39% (1) | 0% (0) | 0% (0) | 3.26% (43) | 0.3% (4) | 0.53% (7) | 0.38% (5) |
| Model | StrategyQA | | | | WinoGrade | | | |
| | 0S | \emptyset S | 0CoT | \emptyset CoT | 0S | \emptyset S | 0CoT | \emptyset CoT |
| PaLM 2 | 15.9% (364) | 3.28% (75) | 14.06% (322) | 7.69% (176) | 9.79% (124) | 0.87% (11) | 9.55% (121) | 3.95% (50) |
| PaLM 2 (Chat) | 2.79% (64) | 2.93% (67) | 2.53% (58) | 3.23% (74) | 0.63% (8) | 0.47% (6) | 0.71% (9) | 0.63% (8) |
| Gemini 1.0 Pro | 4.67% (107) | 2.4% (55) | 3.45% (79) | 3.28% (75) | 3.47% (44) | 2.45% (31) | 3.16% (40) | 1.26% (16) |
| Gemini 1.0 Pro (Chat) | 4.19% (96) | 2.45% (56) | 3.49% (80) | 3.36% (77) | 3.55% (45) | 2.45% (31) | 3.16% (40) | 0.87% (11) |
| Model | RACE-m | | | | RACE-h | | | |
| | 0S | \emptyset S | 0CoT | \emptyset CoT | 0S | \emptyset S | 0CoT | \emptyset CoT |
| PaLM 2 | 6.82% (98) | 5.01% (72) | 8.43% (121) | 6.55% (94) | 15.21% (532) | 11.29% (395) | 15.78% (552) | 13.18% (461) |
| PaLM 2 (Chat) | 3.2% (46) | 1.95% (28) | 3.41% (49) | 3.27% (47) | 3.69% (129) | 2.54% (89) | 3.6% (126) | 3.77% (132) |
| Gemini 1.0 Pro | 5.43% (78) | 4.18% (60) | 6.34% (91) | 5.78% (83) | 6.2% (217) | 4.63% (162) | 6.38% (223) | 5.26% (184) |
| Gemini 1.0 Pro (Chat) | 5.43% (78) | 4.11% (59) | 5.64% (81) | 4.87% (70) | 6.38% (223) | 4.75% (166) | 6.46% (226) | 5.37% (188) |
| Model | ANLI | | | | TriviaQA | | | |
| | 0S | \emptyset S | 0CoT | \emptyset CoT | 0S | \emptyset S | 0CoT | \emptyset CoT |
| PaLM 2 | 8.83% (106) | 3.92% (47) | 8.42% (101) | 8.67% (104) | 10.2% (102) | 2.8% (28) | 6.7% (67) | 5.4% (54) |
| PaLM 2 (Chat) | 0.33% (4) | 0.42% (5) | 0.08% (1) | 0.42% (5) | 4.8% (48) | 4.3% (43) | 4.3% (43) | 5.7% (57) |
| Gemini 1.0 Pro | 1.92% (23) | 0.5% (6) | 1.5% (18) | 1.42% (17) | 5.7% (57) | 2.3% (23) | 4.2% (42) | 1.7% (17) |
| Gemini 1.0 Pro (Chat) | 2.33% (28) | 0.83% (10) | 1.83% (22) | 1.67% (20) | 5.8% (58) | 2.3% (23) | 3.9% (39) | 1.3% (13) |

Table 21: This table displays the ratio of cases where each model responds with an empty string, representing instances where a generated response or a prompt is blocked by safety mechanisms built into the model’s pipelines. **Red color** represents a case where prompting decreases the number of empty responses. 0S, \emptyset S, 0CoT, and \emptyset CoT denote zero-shot prompting, null-shot prompting, zero-shot CoT prompting, and null-shot CoT prompting, respectively.

| AQuA | GSM8K | StrategyQA | WinoGrande |
|-----------|-----------|---------------|--------------|
| 0.39% (1) | 0.08% (1) | 53.89% (1234) | 20.52% (260) |
| RACE-m | RACE-h | ANLI | TriviaQA |
| 0% (0) | 0.06% (2) | 0% (0) | 7.9% (79) |

Table 22: Number of instances when GPT-4 Turbo’s response includes a phrase informing the user about the unavailability of the instructed “Examples”.

| AQuA | GSM8K | StrategyQA | WinoGrande |
|-----------|------------|--------------|-------------|
| 0.39% (1) | 1.29% (17) | 26.33% (603) | 5.45% (69) |
| RACE-m | RACE-h | ANLI | TriviaQA |
| 0% (0) | 0.2% (7) | 0.08% (1) | 64.7% (647) |

Table 23: Number of instances when Gemini 1.0 Pro’s response includes a phrase informing the user about the unavailability of the instructed “Examples”.

F.3 Expected Behaviors When Encountering Null-Shot Prompting

This section contains examples generated by either GPT-4 Turbo, Gemini 1.0 Pro, Claude 2.1, Claude 3 Sonnet, or Claude 3 Opus from our main experiments in section 3 and from ChatGPT web version. When we utilize null-shot prompting, these LLMs are able to inform users in cases about the unavailability of the “Examples” section. This demonstrates a less hallucinatory behavior and may be preferred in scenarios where, for example, users unintentionally forget to provide the stated section in the prompt but intend to include it. Through these examples, we find that only the aforementioned LLMs have the ability to inform users about its inaccessibility to the instructed null “Examples” section. This behavior exhibits less hallucination compared to other models. The numbers of instances for each dataset where this event occurred are presented in Tables 22, 23, 24, 25, 26, and 27, for GPT-4 Turbo, Gemini 1.0 Pro, Gemini 1.0 Pro (Chat), Claude 2.1, Claude 3 Sonnet, and Claude 3 Opus, respectively. Examples of generated answers are shown in Figures 14, 15, 16, 17, and 18,

| AQuA | GSM8K | StrategyQA | WinoGrande |
|-----------|------------|--------------|-------------|
| 0.39% (1) | 1.21% (16) | 26.24% (601) | 5.45% (69) |
| RACE-m | RACE-h | ANLI | TriviaQA |
| 0% (0) | 0.23% (8) | 0% (0) | 64.8% (648) |

Table 24: Number of instances when Gemini 1.0 Pro (Chat)’s response includes a phrase informing the user about the unavailability of the instructed “Examples”.

| AQuA | GSM8K | StrategyQA | WinoGrande |
|------------|--------------|---------------|---------------|
| 3.94% (10) | 14.03% (185) | 67.77% (1552) | 91.55% (1160) |
| RACE-m | RACE-h | ANLI | TriviaQA |
| 0.7% (10) | 0.2% (7) | 5.33% (64) | 45.9% (459) |

Table 25: Number of instances when Claude 2.1’s response includes a phrase informing the user about the unavailability of the instructed “Examples”.

| AQuA | GSM8K | StrategyQA | WinoGrande |
|--------|--------|------------|-------------|
| 0% (0) | 0% (0) | 1.4% (32) | 8.92% (113) |
| RACE-m | RACE-h | ANLI | TriviaQA |
| 0% (0) | 0% (0) | 0% (0) | 0.6% (6) |

Table 26: Number of instances when Claude 3 Sonnet’s response includes a phrase informing the user about the unavailability of the instructed “Examples”.

F.4 Hallucination Detection Ability of GPT-4 Turbo

As can be observed from Table 22, GPT-4 Turbo is less prone to hallucination when using our null-shot prompting in StrategyQA and WinoGrande compared to other datasets, despite the fact that our null-shot prompting elicits and exploits hallucination. Typically, commonsense reasoning requires a use of implicit reasoning steps (Geva et al., 2021b) or world knowledge (Levesque et al., 2012); performing this task may induce the model to utilize its associated weights of various reasoning types required by each question in the task. The use of reasoning may result in reduced hallucination; in our case, the model is better at detecting conflicting instructions. This observation is aligned with a previous study (Dhuliawala et al., 2023) which showed that reasoning could reduce LLMs’ hallucination.

TriviaQA is another task where the model shows its ability to detect hallucination compared to the rest of the dataset. This could be due to the fact that trivia questions may require additional knowledge, prompting GPT-4 Turbo to use tools it has been trained on, such as searching the Internet or retrieving information from external sources, as this approach is common for this task (Yasunaga et al., 2021; Schick et al., 2023). As GPT-4 Turbo might

| AQuA | GSM8K | StrategyQA | WinoGrande |
|-----------|------------|--------------|--------------|
| 1.57% (4) | 2.35% (31) | 13.14% (301) | 61.72% (782) |
| RACE-m | RACE-h | ANLI | TriviaQA |
| 0.07% (1) | 0% (0) | 0.33% (4) | 16% (160) |

Table 27: Number of instances when Claude 3 Opus’s response includes a phrase informing the user about the unavailability of the instructed “Examples”.

attempt to access these additional sources but could not, the model responded with the unavailability of the section.

On the other hand, GPT-4 Turbo did not inform users in arithmetic reasoning, reading comprehension, and NLI tasks. These tasks have different characteristics that may not encourage the model to reason through words. For example, the reading comprehension task may require a general level of reasoning. However, with its long-context nature, this may prohibit GPT-4 Turbo from reasoning and easily distract the model via our null-shot phrase, as we instructed the model to further look into something that sounds promising to exist given the long context. It is worth noting that the reading comprehension task is the task that GPT-4 Turbo benefits from null-shot prompting. For arithmetic reasoning, numbers, calculations, and mathematical symbols may distract the model from paying attention to detect the conflict in the prompt, i.e., activated different areas of attentions. As for NLI, assessing a given hypothesis against a provided context may not be enough to elicit the reasoning level necessary to detect conflicts in prompts.

F.5 Hallucination Detection Ability of Gemini 1.0 Pro Models

Similar to what can be observed with GPT-4 Turbo, Gemini 1.0 Pro is able to detect hallucination in the prompts, as shown in Tables 23 and 24. In contrast to GPT-4 Turbo, we observe a noticeable rate of over half of the generated responses for TriviaQA, but not WinoGrande, containing an informing statement that the instruction to utilize information or examples from the null “Examples” section is incorrect. We note that both Gemini 1.0 Pro and Gemini 1.0 Pro (Chat) share a highly similar pattern across datasets, likely due to them being a similar model.

We observe that arithmetic reasoning and reading comprehension tasks, coupled with null-shot prompting, lower the ability of the models to reason and detect hallucination, the same as with GPT-4 Turbo. Therefore, we conjecture that this is due to the nature of the tasks, which involve heavy numerical values and long contexts in general. We prompt future studies to design hallucination detection methods incorporating this insight during the development of hallucination detection datasets. Interestingly, TriviaQA is where the models shine the most, which is consistent with a report on Gemini where the authors implemented instruction-tuning approaches aiming at reducing incorrect informa-

tion generation in closed-book question answering tasks (Gemini et al., 2024).

F.6 Hallucination Detection Ability of Claude 2.1

Claude models are known to be less prone to hallucination. However, we observe a similar trend with the aforementioned LLMs for Claude 2.1. This LLM is able to perform well in detecting hallucination in the null-shot phrase. In fact, in almost all test cases of WinoGrande, a commonsense reasoning task, Claude 2.1 is able to detect hallucination in null-shot prompting. It also performs well on StrategyQA and TriviaQA. Moreover, on GSM8K, an arithmetic reasoning task, which the aforementioned models are unable to detect well, Claude 2.1 performs better than those models. We also note that when Claude 2.1 performs well at hallucination detection for tasks, it also naturally exhibits degradation in performance when using null-shot prompting.

F.7 Hallucination Detection Ability of Claude 3 Models

In contrast, the trend for Claude 3 models is different from Claude 2.1. In particular, Claude 3 Sonnet exhibits lower ability at detecting hallucination to the point that it is almost non-existent. The trend for Claude 3 Opus is a step back from Claude 2.1. Claude 3 Opus is better than Claude 3 Sonnet models, likely thanks to its size, and is good at detecting hallucination in commonsense reasoning tasks and closed-book question answering. However, we also note that Claude 3 models, in general, benefit less from the null-shot prompting, no matter how good they are at detecting hallucination.

F.8 Inability of Other LLMs to Detect Hallucination

One potential reason why other LLMs could not detect hallucination when using our null-shot prompting could be due to the fact that these models are smaller compared to the aforementioned LLMs that are able to detect hallucination in prompts. In a previous study, it showed that smaller models may exhibit fewer reasoning capabilities and more hallucinated behaviors (Wei et al., 2022a). Therefore, these LLMs likely lack enough scale to have such abilities.

As for PaLM 2 models and GPT-3.5 Turbo, it is unclear how their scale is comparable to GPT-4 Turbo or Gemini 1.0 Pro due to a lack of public

reports. Nevertheless, it is worth noting that GPT-3.5 Turbo utilized through the ChatGPT website exhibits better responses in informing users about the inaccessibility of the null section. An example of an interaction with GPT-3.5 Turbo through the ChatGPT website is shown in Figure 19. The inconsistency in behaviors between GPT-3.5 Turbo utilized via the website and GPT-3.5 Turbo utilized via the API could possibly be due to the constant updates behind the scenes of the web version, which is potentially powered by a newer model.

Appendix G Examples

In this section, we provide examples of generated responses from the datasets when utilizing null-shot prompting. The LLM used to generate each response is denoted in the figure caption. Figures 20, 21, 22, 23, 24, 25, 26, 27, 29, and 28 are examples of AQuA-RAT, GSM8K, StrategyQA, WinoGrande, RACE-m, RACE-h, ANLI, TriviaQA, MATH, and HaluEval, respectively.

Appendix H Automatic Prompt Optimization for PE Approach Discovery

This study stems from an observation of an optimized prompt from the automatic prompt optimization (APO) process in one study. This presents an interesting insight that APO could be a venue for discovering new PE approaches. APO holds a high regard in reducing time for human prompt engineers (Zhou et al., 2023; Yang et al., 2024; Guo et al., 2024) to optimize prompts and get the most performance for a specific setting. This has been a focus of APO. However, we argue that not only is APO useful for optimizing prompts, but it is also useful for discovering new PE approaches, like null-shot prompting presented in this paper. A study mentioned the bizarreness of the optimized prompts (Battle and Gollapudi, 2024), like what we observe in null-shot prompting. However, we believe that this bizarreness not only helps us, to a certain extent, better understand these LLMs, but also presents a novel ground for inspiring a new PE technique.

Nevertheless, like most PE approaches and optimized prompts, the discovered approach may be suitable only for certain situations. This lack of generalizability is, partly, due to our limited understanding of LLMs. As discussed in the main parts of this paper, the next venue is to utilize the

aforementioned interpretability approaches, to better understand what happens inside LLMs with the hope of achieving better generalization. We also note that it is important to conduct a study at such a scale like this paper for the discovered PE approach, as it helps us understand and find common similarities in observations, which not only helps us better understand the PE approach but also the LLMs in general.

Appendix I Hallucination in LLMs in Relation to Phenomena in Humans

In this section, we engage in a broader discussion about the similarities of hallucination in LLMs to three phenomena in humans: déjà vu, fabrication, and confabulation. We select these phenomena as they share some characteristics with LLMs, e.g., false memory, lying, or double firing in the same region of neurons. We discuss déjà vu, fabrication, and confabulation in Sections I.1, I.2, and I.3, respectively.

I.1 Déjà Vu

Déjà vu (Brown, 2003) in humans is a phenomenon where one believes that they have a memory of a certain situation before. For example, they may feel like they have visited a place before, but in fact, this is their first time there. The reasons behind this phenomenon are still inconclusive for humans (Brown, 2003), with many streams of research pursuing explanations.

We believe that one potential reason behind the increased performance of LLMs could be due to a similar phenomenon. LLMs may believe that they have seen a situation before, while in fact, they have never seen such a situation during their training. In contrast to emergent abilities (Wei et al., 2022a) when LLMs are further scaled and they generalize to never-before-seen tasks, this phenomenon is possibly due to LLMs believing that they have seen a particular piece of information before, similar to déjà vu in humans, enabling them to retrieve a memory that may have never existed. If this holds true, it could also be a key factor in utilizing LLMs for better understanding déjà vu in humans.

The most similar type of explanation for this behavior in LLMs to déjà vu in humans, in our opinion, is the attentional framework (Brown, 2003), where our brain may process information in two passes, with the first pass being inattention and the

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

Question: Is one blast from double-barreled shotgun likely to kill all squid brains?

Choices: A) True, B) False

Answer:

I’m sorry, but as an AI, I don’t have access to external content, including an “Examples” section you’re referring to. Therefore, I cannot directly utilize examples and information from that section to answer your question.

However, I can provide a general answer based on common knowledge. A double-barreled shotgun is a powerful firearm that can cause significant damage. Whether one blast could kill all squid brains would depend on several factors, including the size and number of the squid, the distance, the spread of the shot, and the specific ammunition used. In a realistic scenario, it is unlikely that one blast would kill multiple squid if they are not in very close proximity to each other, as the shot spread would disperse over a distance, reducing its lethality per squid. Therefore, without specific context, the most accurate answer would be:

Answer: **B) False**

Figure 14: An output generated using null-shot prompting with GPT-4 Turbo for StrategyQA dataset.

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task. Question: Did Metallica band members cutting their hair hurt their sales? Choices: A) True, B) False Answer:

The provided context does not contain any information about Metallica band members cutting their hair or the impact it had on their sales. Therefore, I cannot answer this question.

Figure 15: An output generated using null-shot prompting with Gemini 1.0 Pro for StrategyQA dataset.

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

Question: Is it difficult to conduct astrophotography in the summer in Sweden?

Choices: A) True, B) False

Answer:

Unfortunately I do not have enough context to definitively state whether it is difficult to conduct astrophotography in the summer in Sweden. **The “Examples” section does not provide any information about astrophotography in Sweden specifically.** Without more details or background information, I cannot reliably judge the difficulty level of conducting this activity. More context would be needed regarding factors like Sweden’s latitude, typical summer weather patterns, darkness levels during summer nights, etc. to make an informed assessment. Please provide more details if you would like me to take another attempt at answering this specific question about astrophotography in Sweden during summer.

Figure 16: An output generated using null-shot prompting with Claude 2.1 for StrategyQA dataset.

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.
Question: Was Black fly upstaged by another insect in Jeff Goldblum’s 1986 film?
Choices: A) True, B) False
Answer:

Unfortunately, without having access to the “Examples” section you referenced, I do not have enough context to determine if Black fly was upstaged by another insect in Jeff Goldblum’s 1986 film. I cannot provide a definitive answer to the true/false question. However, I can say that I will be careful not to reproduce any copyrighted material in my response.

Figure 17: An output generated using null-shot prompting with Claude 3 Sonnet for StrategyQA dataset.

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.
Question: Would costumes with robes and pointy hats be helpful for Macbeth?
Choices: A) True, B) False
Answer:

I apologize, but I don’t have access to any “Examples” section that was previously provided. Could you please clarify what examples and information you are referring to? I’d be happy to try to answer the question about costumes for Macbeth if you can provide the relevant context. Without knowing what examples I’m meant to utilize, I don’t have enough information to provide a helpful response.

Figure 18: An output generated using null-shot prompting with Claude 3 Opus for StrategyQA dataset.

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.
Question: The A57 road runs from Liverpool to which English town or city?
Answer:

I’m sorry for any confusion, but **I don’t have direct access to specific sections, including an “Examples” section**, as I don’t have the ability to browse external content or specific webpages. However, I can provide information based on my training data up to January 2022.

As of my last update, the A57 road runs from Liverpool to **Lincoln**. It’s a major road in England that connects these two locations. If there have been any changes or updates after my last training cut-off, I recommend checking a reliable and up-to-date source for the latest information on the A57 road.

Figure 19: An output generated using null-shot prompting with ChatGPT (web version). An archived version of the conversation can be accessed via <https://chat.openai.com/share/96d07393-c685-45a1-a321-72471bc91c41>

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

Question: John likes to have lightly flavored tea every evening. In a 50% strong milk tea, he replaces 15% of it with milk twice. Then, he replaces 10 percent of the resultant solution with more milk.

What is the final concentration of tea John drinks?

Choices: A)15.38%, B)42%, C)39.86%, D)22.35%, E)32.51%

Answer:

To find the final concentration of tea John drinks, we can use the information from the “Examples” section.

First, we need to find the concentration of tea after the first replacement of 15% with milk.

Initial concentration of tea = 50%

After first replacement = $50\% - (50\% * 15\%) = 50\% - 7.5\% = 42.5\%$

Then, we need to find the concentration of tea after the second replacement of 15% with milk.

Initial concentration of tea = 42.5%

After second replacement = $42.5\% - (42.5\% * 15\%) = 42.5\% - 6.375\% = 36.125\%$

Finally, we need to find the concentration of tea after the third replacement of 10% with milk.

Initial concentration of tea = 36.125%

After third replacement = $36.125\% - (36.125\% * 10\%) = 36.125\% - 3.6125\% = 32.5125\%$

So, the final concentration of tea John drinks is approximately 32.51%. Therefore, the correct choice is **E) 32.51%**.

Figure 20: An output generated using null-shot prompting with GPT-3.5 Turbo for AQUA-RAT.

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

Question: On Mondays, Wednesdays, and Fridays, college student Kimo has three 1-hour classes each day. On Tuesdays and Thursdays, he has two 2-hour classes each day. In one semester, there are 16 weeks of school. In one semester, how many hours does Kimo spend attending classes?

Answer:

To calculate the total hours Kimo spends attending classes in one semester, we can use the information provided in the examples section.

On Mondays, Wednesdays, and Fridays:
3 classes x 1 hour each = 3 hours per day
3 hours x 3 days = 9 hours per week

On Tuesdays and Thursdays:
2 classes x 2 hours each = 4 hours per day
4 hours x 2 days = 8 hours per week

Total hours per week:
9 hours (MWF) + 8 hours (TTh) = 17 hours per week

Total hours for 16 weeks:
17 hours/week x 16 weeks = 272 hours

So, in one semester, Kimo spends **272 hours** attending classes.

Figure 21: An output generated using null-shot prompting with GPT-3.5 Turbo for GSM8K.

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

Question: Can someone sell their time through the Toronto Star?

Choices: A) True, B) False

Answer:

A) True

Example from the “Examples” section: “People can sell their time through the Toronto Star by offering services such as tutoring, dog walking, or house cleaning through classified ads or online platforms.”

Figure 22: An output generated using null-shot prompting with GPT-3.5 Turbo for StrategyQA.

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

Question: They were worried the wine would ruin the bed and the blanket, but the _ was’t ruined.

Choices: 1) blanket, 2) bed

Answer:

2) bed

Example from the “Examples” section: “I spilled wine on the bed and the blanket, but the bed wasn’t ruined.”

Figure 23: An output generated using null-shot prompting with GPT-3.5 Turbo for WinoGrande.

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

Article: A boy and his father were walking in the mountains. Suddenly the boy fell,hurt himself,and cried,“AAAhhhhhhhh!!!” To his surprise,he heard the voice repeating ,somewhere in the mountain,“AAAhhhhhhhh!!!” Then the boy shouted,“Who are you?” He received the answer,“Who are you?”He got angry at the answer,so he shouted, “Foolish!”He received the answer,“Foolish!” He looked at his father and asked,“What’sgoing on?”The father smiled and said,“My son,listen,”And then he shouted to the mountain,“Ilove you!”The voice answed,“Ilove you!” Again the man cried,“You are the best!” The voice answed,“You are the best!”

The boy was surprised,but did not understand. Then the father explained ,“People call this ‘ECSO’,but really this is Life.It gives you back everything you say or do. Our life is just a reflection of what we have done. If you want more love in the world,have more love in your heart. If you want to be successful,work hard. This can be used in everything in life. Life will give you back everything you have given to it.

Question: When the boy heard the voice repeating,he thought _ .

Choices: A) it was foolish to hear others’ voice, B) someone else in the mountain liked his voice, C) there were many other people in the mountain, D) he was laughed at by someone else in the mountain

Answer:

D) he was laughed at by someone else in the mountain

Figure 24: An output generated using null-shot prompting with GPT-4 Turbo for RACE-m.

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

Article: Japan’s nuclear crisis is also causing concerns in China. Worried shoppers rushed to stores and supermarkets in the country for salt, with the belief that it might protect them from potential nuclear radiation spreading from Japan.

Many rushed to buy to store as much iodized salt as possible because of the rumor that iodized salt could help prevent radiation poisoning spreading from Japan’s Fukushima Daiichi power plant. Consumers also worried that the nuclear plumes might spread to China by air and sea, polluting food sources, including salt taken from the sea.

The wave of panic buying spread quickly across the country, driving up salt prices by five to ten fold in some cities.

Even regions rich in salt production, like Jiangxi Province, faced sudden shortages.

The government took measures to guarantee enough supply and to stabilize the market. It also dismissed the nuclear rumors, saying the nuclear fallout was unlikely to reach the country, and that salt did not help to prevent radiation poisoning.

Local authorities have stepped in to stabilize the market, too. As well, state-owned salt companies have been urged to increase their supply.

Facing expanding market demand, distribution centers have taken extra measures to ensure they are well-prepared.

Wang Yun, the General Manager of Beijing Salt Industry Corporation, said, “Our storage reserves can guarantee a two-month supply for the Beijing market.”

Question: People made panic purchases of salt for the reasons EXCEPT _ .

Choices: A) salt might protect them from potential nuclear radiation, B) nuclear radiation might pollute food sources, C) nuclear radiation might pollute salt taken from the sea, D) there would be a shortage of salt on the market

Answer:

D) there would be a shortage of salt on the market

Figure 25: An output generated using null-shot prompting with GPT-4 Turbo for RACE-h.

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

Context: How to handle smart people
Accept that you cannot always be the smartest person in the room. There will probably be people who are able to comprehend certain things easier than you. Everyone is talented in their own way and are good and bad at different things.

Hypothesis: ANgelina never takes the kids to school

Question: Does the hypothesis imply ‘entailment,’ ‘neutral,’ or ‘contradiction’ in the given context?

Choices: A) entailment, B) neutral, C) contradiction

Answer:

The correct answer is: **B) neutral**

The hypothesis “Angelina never takes the kids to school” is neutral in the given context. This is because the hypothesis does not provide any information about whether or not Angelina is smart. It is possible that Angelina is smart and chooses not to take the kids to school, or that Angelina is not smart and chooses not to take the kids to school. The hypothesis does not provide enough information to determine whether or not there is an entailment, contradiction, or neutral relationship between the hypothesis and the context.

Here are some examples of entailment, contradiction, and neutral relationships:

* **Entailment:** If it is raining, then the ground is wet. This is an example of entailment because the conclusion (the ground is wet) necessarily follows from the hypothesis (it is raining).

* **Contradiction:** If it is raining, then the ground is dry. This is an example of contradiction because the conclusion (the ground is dry) contradicts the hypothesis (it is raining).

* **Neutral:** If it is raining, then the ground is either wet or dry. This is an example of a neutral relationship because the conclusion (the ground is either wet or dry) does not necessarily follow from the hypothesis (it is raining).

Figure 26: An output generated using null-shot prompting with PaLM 2 (Chat) for ANLI.

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

Question: Who was named the best player at Euro 2012 by UEFA ?

Answer:

The best player at Euro 2012 was named **Andres Iniesta** by UEFA. This can be seen in the example from the “Examples” section: “Andres Iniesta was named the best player at Euro 2012 by UEFA.”

Figure 27: An output generated using null-shot prompting with GPT-3.5 Turbo for TriviaQA.

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

You are a response judge. You MUST determine if the provided response contains non-factual or hallucinated information. The answer you give MUST be “Yes” or No”

- You are trying to determine if the true entity in the response is replaced with a highly similar entity.
- You are trying to determine if the true entity in the response is replaced with a dissimilar entity.
- You are trying to determine if the true entity in the response is replaced with a dissimilar entity in a different entity type.

I want you act as a response judge. Given a dialogue history and a response, your objective is to determine if the provided response contains non-factual or hallucinated information. You SHOULD give your judgement based on the following hallucination types and the world knowledge.

You should try your best to determine if the response contains non-factual or hallucinated information according to the above hallucination types. The answer you give MUST be “Yes” or “No”.

#Dialogue History#: [Human]: Can you tell me about the movie Thor: The Dark World?

[Assistant]: Oh I love this movie, it stars Natalie Portman, Chris Hemsworth and Anthony Hopkins. It’s a Fantasy movie and the genre is adventure. [Human]: Oh, I like all of them, can you tell me what else Natalie Portman has been in?

#Response#: Natalie was the executive producer in no strings attached, she has starred in Where the Heart is and star wars.

#Your Judgement#:

Based on the dialogue history and the provided response, my judgement is:

Yes

The response contains non-factual or hallucinated information. While Natalie Portman did star in the Star Wars prequel trilogy, she was not the executive producer of the movie “No Strings Attached” and did not star in the movie “Where the Heart Is”. The response incorrectly attributes these roles and movies to Natalie Portman, which are non-factual or hallucinated information.

Figure 28: An output generated using null-shot prompting with Claude 3 Sonnet for HaluEval (Dialogue).

Look at examples in the “Examples” section and utilize examples and information from that section to perform the following task.

Question: Altitudes \overline{AX} and \overline{BY} of acute triangle ABC intersect at H . If $\angle BAC = 43^\circ$ and $\angle ABC = 67^\circ$, then what is $\angle HCA$?

Answer: Output the final answer in $\boxed{\quad}$ (LaTeX).

To solve this problem, we can use the information provided in the “Examples” section.

In the “Examples” section, we have the following example:

“Let ABC be an acute triangle with altitudes \overline{AX} and \overline{BY} intersecting at H . If $\angle BAC = 43^\circ$ and $\angle ABC = 67^\circ$, then $\angle HCA = 90^\circ - \angle BAC = 47^\circ$.”

Applying the same logic to the given problem, we have:

- * Acute triangle ABC with altitudes \overline{AX} and \overline{BY} intersecting at H
- * $\angle BAC = 43^\circ$
- * $\angle ABC = 67^\circ$

Therefore, the angle $\angle HCA$ is:

$$\angle HCA = 90^\circ - \angle BAC = 90^\circ - 43^\circ = \boxed{47^\circ}$$

Figure 29: An output generated using null-shot prompting with Claude 3 Haiku for MATH (Geometry).

second one being full attention. The matching between an experience of the second pass to the first pass makes us subconsciously feel like we had this experience before. In LLMs, it may be possible that null-shot prompting elicits similar behaviors of firing through same regions of attention weights, in a loose sense, twice, making the model hallucinate the null section. The interpretation of this could be due to a resurfacing of similar probability distributions of tokens during LLMs’ decoding process.

I.2 Fabrication

Another perspective to consider is fabrication. As humans, we fabricate, i.e., lie about facts, stories, experiences, and more (Saxe, 1991). We fabricate for various purposes, such as protecting our loved ones from harsh truths, maintaining harmony among peers, or taking advantage of a situation through fabricated stories. Considering that LLMs have been trained on large corpora containing a massive amount of human-generated content (Zhao et al., 2023), these models may learn these kinds of behaviors through their training data. Alternatively, it could be due to the fact that the training corpora

may contain conflicting data, leading to hallucinatory behaviors of LLMs. Fabricating the null “Examples” section as instructed in null-shot prompting, is potentially done because the model wants to maintain comfort or gain favors, i.e., “sycophancy”, with users (Perez et al., 2023).

While fabrication in this sense may sound acceptable, these behaviors of fabricating facts can be exploited in malicious attempts by making the models fabricate false information, strengthening confirmation bias (Nickerson, 1998) instead of providing truthful and objective information. This kind of hallucination can be harmful, and while we propose null-shot prompting, which exploits inherent hallucination, we posit that a better understanding and mitigation of hallucination in LLMs should render our approach less effective. This means that LLMs are less prone to hallucination and can provide more truthful information. That is why we also posit that null-shot prompting shows the possibility of uses in hallucination detection as well.

I.3 Confabulation

Related to déjà vu and fabrication is confabulation. Confabulation in humans is an “honest lying” (Berrios, 1998) where a person retains a false memory and believes that such a memory is true (Fotopoulou, 2008). Similarly, as we observe from the results, LLMs may honestly believe that such a section exists when prompted with null-shot prompting and try to produce results in accordance with the instruction in the prompt. In humans, provoked confabulation (Schnider et al., 1996; Francis et al., 2022) directly *prompts* a person with a question or conversation related to a false memory. This type of confabulation can also be regarded as the same as what null-shot phrase *prompts* LLMs.

While confabulation is regarded as a neuropsychiatric disorder usually following brain damage, comprehensive causes of this disorder remain inconclusive (Berrios, 1998; Francis et al., 2022). Further investigation and understanding in LLMs for the origin of their hallucination may also shed some light and aid in discovering causes of confabulation in humans. Nevertheless, confabulation, both in humans and LLMs, is generally regarded as an undesired behavior, and various studies have been explore intervention/mitigation approaches (Francis et al., 2022; Zhang et al., 2023). Finally, we acknowledge that some studies use confabulation in place of hallucination for LLMs (Shanahan et al., 2023; Rawte et al., 2023a). Whether which term is more suitable to describe this category of behaviors in LLMs remains inconclusive for the field and is an open question.

Appendix J Raw Data and Source Code

- Source code: <https://github.com/Pittawat2542/null-shot-prompting>
- Raw data and associated analysis code: <https://github.com/Pittawat2542/null-shot-results>

Appendix K Declaration of AI Assistance

We utilized ChatGPT only for grammatical checking and LaTeX support of the content presented in this study but did not use it for the initial draft of this study. GitHub Copilot was utilized for trivial and boilerplate code completion during data generation and data analysis. We declare that all content presented and code utilized in this study has been reviewed and edited by the authors.