Large Language Models are Null-Shot Learners

Anonymous ACL submission

Abstract

This paper presents null-shot (∅-shot) prompting. ∅-shot prompting exploits hallucination in large language models (LLMs) by instructing LLMs to utilize information from the "Examples" section that never exists within the provided context to perform a task. While reducing hallucination is crucial and non-negligible for daily and critical uses of LLMs, we propose that in the current landscape in which these LLMs still hallucinate, it is possible, in fact, to exploit hallucination to increase performance in performing tasks compared to standard zero-shot prompting. Experiments with eight LLMs show improvements in performance across the majority of eight datasets, including reading comprehension, arithmetic reasoning, and closed-book question answering. The observed inconsistency in increased relative performance across the LLMs also potentially indicates a different degree of inherent hallucination in each model. These differences show that it is possible to utilize ∅-shot prompting as a way to detect degrees of hallucination in LLMs using existing benchmarking datasets. We also perform ablation studies, including experimenting with a modified version of ∅-shot prompting that incorporates ideas from zero-shot chain-of-thought prompting, which shows different trends of results.

1 Introduction

Large language models (LLMs) have been adopted across domains and applications due to their versatile capabilities (Zhao et al., 2023). A subset of fine-tuned LLMs that currently dominates is *aligned auto-regressive LLMs* (Ouyang et al., 2022); notable examples include ChatGPT (OpenAI, 2022), PaLM 2 for Chat (Anil et al., 2023), and Llama 2 Chat (Touvron et al., 2023). This type of LLM is fine-tuned to follow user instructions in natural language, i.e., *alignment*. To interact with these models, users provide a prompt, a text input in the case of text-to-text generative models, to LLMs. These prompts greatly dictate the outputs generated by LLMs and directly affect performance on tasks. While prompts can become very complex and consist of many elements (White et al., 2023), typically, a prompt consists of two essential parts for zero-shot prompting (Wei et al., 2022a): (1) task instructions, i.e., what to do and relevant context, and (2) task inputs. However, zero-shot prompting may not bring out the full potential of LLMs, giving rise to more complex instruction composition and prompt engineering (PE).

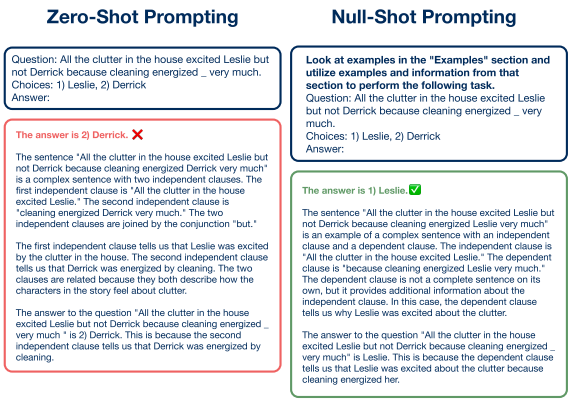

Figure 1: Examples of a generated response outputted by PaLM 2 for Chat when using zero-shot prompting (left) and ∅-shot prompting (right) for WinoGrande.

PE is a field that studies approaches to improve prompts used to interact with LLMs, thereby enhancing the performance of tasks (White et al., 2023). Various PE techniques have been proposed to improve task performance by composing prompts in a specific style, including few-shot prompting (Brown et al., 2020), chain-of-thought (CoT) prompting (Wei et al., 2022c), and zero-shot CoT (0CoT) prompting (Kojima et al., 2022). These PE approaches typically exploit the fact that outputs from LLMs are generated auto-regressively. Therefore, a prompt and the generated output to-

kens so far act as conditions in an output probability space. In other words, we can view these conditions as localizing the trajectory to generate the next output token. Thus, we can consider existing PE approaches as either providing more relevant tokens as an input (prompt) to the model or providing a prompt that elicits the model to generate pretext in the output before providing a final answer, or a combination of both.

For example, in few-shot prompting, this approach involves the inclusion of examples in a prompt, i.e., an input to the model has a longer relevant context. Similarly, in generated knowledge prompting (Liu et al., 2022), prompts include more relevant knowledge. On the other hand, 0CoT prompting instructs LLMs to think step-by-step; hence, the models have been instructed to generate reasoning steps in the output before reaching a final conclusion. CoT prompting exploits both a longer relevant context in a prompt with explicit reasoning steps and elicits the model to follow the provided steps before giving the final answer in its generated outputs. We take an approach similar to 0CoT prompting, encouraging models to provide more preceding tokens before reaching a conclusive answer.

Despite the usefulness of these PE approaches, hallucination in LLMs presents another challenge hindering the widespread use of LLMs, specifically in critical applications. Hallucination, as defined by previous studies (Rawte et al., 2023; Li et al., 2023b; Zhang et al., 2023), is a behavior when LLMs produce outputs that include false or conflicting information. Hallucination is inherent in LLMs and comes in three main types (Zhang et al., 2023): (1) input-conflicting, (2) context-conflicting, and (3) fact-conflicting hallucination. Input-conflicting hallucination occurs when LLMs produce outputs that conflict with user instructions, while context-conflicting hallucination happens when conflicts occur within the generated output itself. The most notable and important type of hallucination is fact-conflicting, where the model produces seemingly possible false information. Various attempts (Zhang et al., 2023) have been made to reduce hallucination at various stages of LLMs, such as pre-training, alignment, and inference. Notably, a related study introduced chain-of-verification (CoVe) prompting (Dhuliawala et al., 2023), a PE approach aimed at reducing hallucination in responses through additional verification steps before returning a conclusive output.

However, we take an alternative approach to CoVe prompting. Instead of using prompting to reduce hallucination, we introduce a counter-intuitive PE technique to exploit the context-conflicting hallucination of LLMs to enhance the performance of LLMs on tasks. We propose null-shot (∅-shot) prompting where we instruct LLMs to perform tasks by looking into and utilizing a non-existent imaginary, i.e., null, "Examples" section, which in existing relevant approaches contain explicit examples of the task. We conduct experiments on eight datasets consisting of arithmetic reasoning, commonsense reasoning, reading comprehension, natural language inference, and closed-book question answering. We select six models for the main experiment: PaLM 2 and PaLM 2 for Chat (Anil et al., 2023), Gemini Pro and Gemini Pro (Chat) (Team et al., 2023). GPT-3.5 Turbo (OpenAI, 2022), and GPT-4 Turbo (OpenAI, 2023). We also include Llama 2 7B and Llama 2 7B Chat (Touvron et al., 2023) for a scaling analysis.

Similar to few-shot prompting, where a previous study (Brown et al., 2020) observed increased performance when providing explicit examples of the task in the prompt, asking LLMs to utilize these null, i.e., implicit, examples shows increased performance. We find an improvement up to 44.62% in one dataset of arithmetic reasoning tasks using Gemini Pro (Chat), compared to zero-shot prompting. We also observe improvements in other combinations of models and tasks. In particular, PaLM 2 is the most notable, as ∅-shot prompting improves the performance of the model on the majority of tasks. Examples of outputs from zero-shot and ∅-shot prompting using PaLM 2 for Chat for the WinoGrande dataset are shown in Figure 1.

In this study, we also discuss the possibility of utilizing ∅-shot prompting for assessing models' hallucination. Additionally, we conduct ablation studies, consisting of a scaling analysis, a variant of ∅-shot prompting inspired by 0CoT prompting, positions of ∅-shot phrase, and effects of each component in the ∅-shot phrase. We find that ∅-shot prompting is effective in Llama 2 7B but not in Llama 2 7B Chat. Also, the reasoning variant of ∅-shot prompting is generally less effective compared to the 0CoT prompting baseline. Finally, placing the phrase before the task instruction shows the most effectiveness in most of the datasets. In summary, our contributions are as follows:

- We propose ∅-shot prompting, exploiting hal-

∅-shot Phrase

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

Figure 2: A ∅-shot phrase instructs LLMs to look into and utilize information from the null section.

lucination in LLMs to improve performance across tasks.

- We perform extensive evaluation across six tasks using eight LLMs and discuss implications of results as well as the potential of using ∅-shot prompting for hallucination detection.
- We conduct various ablation studies, consisting of a scaling analysis, a reasoning variant of ∅-shot prompting, positions of the ∅-shot phrase, and effects of each prompt component in the phrase.

2 Null-Shot Prompting

We propose ∅-shot prompting by placing the ∅-shot phrase, as shown in Figure 2, at the beginning of the prompt, i.e., before task instructions and task inputs, in contrast to 0CoT prompting where they place the phrase at the end of the prompt. The reason for this positioning is the observed superiority compared to placing it at the end of the prompt, as demonstrated in Section 5.3. We can view the phrase as instructing LLMs to retrieve the "Examples" section from their internal memory, i.e., trained weights, to accomplish this task.

We note that one of expected behaviors of LLMs when encountering this kind of instruction should be to notify users that there is an issue in the instruction, where we instruct the models to utilize information that we did not provide. We observe several instances, as shown in Appendix G, of this expected behavior from GPT-4 Turbo and Gemini Pro. The instances demonstrate behavior in informing the user about the unavailability of the "Examples" section.

3 Experiments

To assess the performance of ∅-shot prompting, we conduct experiments on eight models: six for the main experiment and two additional models for the scaling analysis. LLMs used for the main experiment are PaLM 2, PaLM 2 for Chat, Gemini Pro, Gemini Pro (Chat), GPT-3.5 Turbo, and GPT-4 Turbo; Llama 2 7B and Llama 2 7B Chat are used for the scaling analysis. Full details of the setup for each LLM are described in Appendix C. The experiments are performed on six tasks across eight datasets. Setup details for each dataset are described as follow. Additional details regarding ablation studies are provided separately in Section 5.

We evaluate the performance of ∅-shot prompting and zero-shot prompting–the latter is a baseline–on six tasks across eight 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 ∅-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, are shown in Appendix B, while qualitative examples of each dataset are presented in Appendix J.

Arithmetic Reasoning: *AQuA-RAT* (Ling et al., 2017) provides a variety of mathematical questions requiring different strategies to solve. 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., 2021) 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 chal-

lenge for models. We utilize both versions of the dataset in the experiment.

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 ∅-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.

4 Results and Discussions

We observe an overall increase in performance when comparing zero-shot and ∅-shot prompting on each combination of dataset and model. We utilized approximately 1340 GPU hours on NVIDIA A100 80GB and L40S 48GB for all Llama 2 models' evaluations and about 290 hours of accumulated inference time from interacting with each model via APIs. Relative results are shown in Table 1, absolute results are presented in Appendix I, and error analysis is discussed in Appendix H. Additional discussion about hallucination in LLMs compared to phenomena in humans is provided in Appendix F. Henceforth, in all the tables, including Table 1, in cases where performance is increased compared to the baseline, the percentage values are colored green; **bold** denotes the maximum increase (in **green**) or minimum decrease (in **black**) ratio in performance.

Overall, we find that the Gemini Pro models exhibit the most significant increase in performance compared to the other models, within the same dataset, when using ∅-shot prompting for the arithmetic reasoning task, i.e., 44.62% (chat) and 28.97% increase in AQuA-RAT and GSM8K, respectively. We also note that PaLM 2 shows increased performance across all datasets except for AQuA-RAT, with a slight decrease in performance. Additionally, we observe that our ∅-shot prompting diminishes the performance across tasks for GPT-4 Turbo, especially in StrategyQA and WinoGrande, both of which are commonsense reasoning task. Similarly, we observe noticeable reduced performance when using ∅-shot prompting with the Gemini Pro models in StrategyQA and TriviaQA.

4.1 PaLM 2 vs PaLM 2 for Chat

PaLM 2 exhibits increased performance across all datasets except for AQuA-RAT, which shows a slight decrease in performance. Similarly, we observe improved performance in the majority of cases for PaLM 2 for Chat. However, the increased performance of PaLM 2 for Chat using ∅-shot prompting can not match that of PaLM 2. This difference is likely attributed to the possibility that PaLM 2 for Chat underwent further fine-tuning specifically for chat conversations, i.e., multi-turn conversations. Throughout this fine-tuning process, additional guidelines and knowledge may have been incorporated into the model, leading PaLM 2 for Chat to exhibit potentially less hallucination compared to PaLM 2. The relatively lower increase in performance may serve as an indicator of this characteristic.

Another noteworthy observation is that, as PaLM 2 is a text-to-text generative LLM focused on natural language tasks, it tends to provide an answer with only one character in cases where the dataset offers multiple choices. Even if ∅-shot prompting does not result in an increase in the output token count (i.e., output tokens play a less significant role as a condition for output generation), the ∅-shot phrase still significantly improves performance compared to the baseline in each dataset given the fact that they also tend to yield only one character as the final output. Therefore, this finding could also serve as evidence of the effectiveness of ∅-shot prompting as a sufficiently strong condition to guide the model in generating the correct answer.

4.2 GPT-3.5 Turbo vs GPT-4 Turbo

GPT-3.5 Turbo, in general, outperforms GPT-4 Turbo when using ∅-shot prompting. GPT-3.5 Turbo demonstrates a significant performance improvement in the arithmetic reasoning task. However, when considering other datasets, its performance is subpar compared to PaLM 2 and PaLM 2 for Chat. GPT-3.5 Turbo performs relatively suboptimal on WinoGrande, reading comprehension, and NLI, with performance decreases ranging from -1.19% to -3.61%. On the other hand, GPT-4 Turbo has no notable performance improvement across tasks, except for a slight increase in reading comprehension.

Based on observations, a few assumptions can

Model	AQuA	GSM8K	StrategyQA	WinoGrande	RACE-m	RACE-h	ANLI	TriviaQA
PaLM 2	-2.70%	11.28%	**10.95%**	**10.10%**	1.85%	**3.64%**	**2.71%**	**7.01%**
PaLM 2 for Chat	5.26%	2.25%	1.66%	6.97%	1.04%	0.68%	1.56%	-0.14%
GPT-3.5 Turbo	33.94%	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%	0.30%	0.42%	-0.26%	-0.94%
Gemini Pro	38.46%	**28.97%**	-24.43%	-1.36%	**1.93%**	2.13%	2.14%	-63.96%
Gemini Pro (Chat)	**44.62%**	27.93%	-25.39%	-1.12%	0.74%	1.63%	1.63%	-63.97%

Table 1: This table presents relative results indicating the performance changes compared to the zero-shot prompting baseline.

be drawn. First, GPT-3.5 Turbo exhibits strong hallucination in the arithmetic reasoning task, as evidenced by a substantial performance increase on both AQuA-RAT and GSM8K datasets. Second, GPT-4 Turbo experiences less hallucination induced by ∅-shot prompting across tasks. This suggests that GPT-4 Turbo is more adept at handling hallucination, and a non-factual phrase like the one introduced in ∅-shot prompting is less effective for this model. This aligns with a prior report indicating that GPT-4 models are less prone to hallucination compared to GPT-3.5 models (OpenAI et al., 2023) and observations from qualitative examples in Appendix G. Finally, both GPT-3.5 Turbo and GPT-4 Turbo exhibit less hallucination than PaLM 2 and PaLM 2 for Chat in general, as evidenced by the subtle performance changes.

4.3 Gemini Pro

Similar to GPT-3.5 Turbo in arithmetic reasoning, we find drastic increases when using ∅-shot prompting with Gemini Pro. Thus, the Gemini Pro models are potentially more prone to hallucination in arithmetic reasoning tasks. As all of the models, except GPT-4 Turbo, show increased performance in at least one dataset of the arithmetic reasoning task, we argue that this is potentially due to the fact that most hallucination detection benchmarks and approaches in mitigating hallucination focus more on text-based tasks (Zhang et al., 2023; Rawte et al., 2023). Arithmetic reasoning tasks are one area where there is less emphasis on mitigating hallucination. Thus, we observe this trend of effectiveness of ∅-shot prompting across all models. We encourage future studies to put more emphasis on mitigating hallucination in arithmetic-related tasks as well.

We also observe the opposite trend in commonsense reasoning and closed-book question answering tasks, particularly in StrategyQA and TriviaQA. Performances in StrategyQA and TriviaQA show a noticeable drop up to -64.97%. We posit that this is due to the models refusing to perform the tasks as the instruction involves the use of "Examples", which does not appear in the context. Qualitative examples of this behavior of Gemini Pro are shown in Appendix G. Results observed on StrategyQA and TriviaQA are in line with the prior report by Team et al. (2023) regarding the factuality aspect of the model, showing less hallucination in situations where the models are elicited to produce false information.

We also want to point out that the reading comprehension task, which has a longer context compared to other tasks, generally elicits more hallucination from Gemini Pro, similar to other models except GPT-3.5 Turbo. We conjecture that this shows such models are less effective in detecting non-factual content in the prompt and more prone to hallucination with longer context. This may prompt future studies on how the length of the input context can affect LLMs' hallucination.

4.4 Null-Shot Prompting for Hallucination Detection

Building on the previous aforementioned report (OpenAI et al., 2023) and empirical results observed in this study, the general conclusion is that GPT-4 Turbo is the least hallucinated model when instructed with the ∅-shot phrase, while GPT-3.5 Turbo is the next least hallucinated model in all tasks except for the arithmetic reasoning task. PaLM 2 models are the most hallucinated, while further fine-tuning for chat conversation in PaLM 2 for Chat demonstrates the effectiveness of the approach in reducing hallucination due to ∅-shot prompting.

For Gemini Pro, we find the results of the arithmetic reasoning datasets quite surprising, with a very high increase in performance compared to the results of StrategyQA and TriviaQA, where ∅-shot prompting significantly degrades the performance due to the models being able to detect hallucination and stop performing the task. This observation

shows another notable aspect: that our approach may be able to provide finer detailed information on what aspects a model of interest tends to hallucinate. Furthermore, arithmetic reasoning, in general, seem to be an area where LLMs' hallucination is easily elicited through ∅-shot prompting.

This also exemplifies that not only can ∅-shot prompting be utilized to increase performance in hallucinated LLMs, but it can also serve as a simple proxy for determining the degree of hallucination in LLMs. In other words, the higher the performance increase stemming from ∅-shot prompting compared to the baseline, the more likely the model exhibits higher hallucinated responses. Furthermore, utilizing ∅-shot prompting does not require any specialized hallucination detection dataset, which are expensive to construct, task-specific, and scarce as of the current state, contrary to our prompting approach that can be applied to any existing benchmarking datasets for various tasks.

4.5 "Examples"

As can be observed from the increased performance across the datasets and models, as well as qualitative examples shown in Appendix J, it is possible to draw an assumption that the null examples that these LLMs envisioned and utilized to increase their performance may come from their own internal knowledge, i.e., trained weight parameters. Many studies have shown that LLMs may have internal mental model demonstrated through space and time world models (Gurnee and Tegmark, 2023), reasoning capabilities (Wei et al., 2022c; Kojima et al., 2022), and abilities to use tools (actions) (Yao et al., 2023b). Nevertheless, studies (Valmeekam et al., 2023a,b) argued that this is limited to only some simpler tasks, and the model fails in more complex scenarios or fails to perform tasks autonomously. We conjecture that results shown through our experiments demonstrate that LLMs, at least to a certain degree, have their own internal world model. This trait enables them to look into the null "Examples" section and utilize it to increase performance in tasks.

It may also be possible that the examples that these LLMs envisioned and referred to might be the best possible examples that will help increase the performance of a particular task. If this holds true, it may be possible to eliminate the need to supply explicit examples, as in few-shot prompting, and rely on LLMs themselves to retrieve these null examples while still achieving higher performance. Future studies could also explore whether changing the instruction to that of providing an explicit number of shots to retrieve, for example, *Look at the first three examples in the "Examples" section*, or looking into more sections or examine how section name affects the performance of LLMs on datasets.

One potential area where the performance of ∅-shot prompting could be further improved might be to provide the ∅-shot phrase along with actual explicit examples. A hybrid-shot approach like this may further improve the performance of the model, i.e., null examples could augment the explicit examples. Besides, this hybrid approach might be a key to achieving increased performance while reducing hallucinatory behaviors of the model for a particular task. Another key area that could be worth investigating is to utilize interpretability techniques similar to a study by Cunningham et al. (2023) to understand how ∅-shot prompting triggers attention areas of LLMs and verify if the models really utilize the internal knowledge when instructed to.

On the other hand, we posit that this approach also demonstrates potential in better understanding hallucination in LLMs. Similar to asking LLMs to look into the null "Examples" section in our approach, providing LLMs with other kinds of instructions to utilize the null context in the tasks may hold another key to investigate hallucination in specific areas. As previously discussed, this may serve as a simpler way to assess hallucination in LLMs.

5 Ablation Studies

We perform ablation studies to better understand how our prompting technique performs on smaller LLMs, namely, Llama 2 7B and Llama 2 7B Chat, in Section 5.1. Furthermore, we attempt to fuse 0CoT and ∅-shot prompting together, coined as null-shot chain-of-thought (∅CoT) prompting, and assess its performance against 0CoT prompting, its baseline, in Section 5.2. We also demonstrate that the position of placing the ∅-shot phrase affects performance in Section 5.3. Finally, we conduct an experiment to assess how each component in the ∅-shot phrase contribute to the performance of ∅-shot prompting in Section 5.4.

5.1 Scaling Analysis

In the scaling analysis, we follow the main experiment procedure described in Section 3 but use Llama 2 7B and the Llama 2 7B Chat. We include

both Llama 2 models in our study to examine the effectiveness of ∅-shot prompting on smaller models. The results from the study are shown in Table 2. We observe increases in performance across all tasks with ∅-shot prompting using Llama 2 7B, except for AQuA-RAT and WinoGrande. In the case of GSM8K, we observe an increase in performance when using ∅-shot prompting with Llama 2 7B Chat. However, the rest of the datasets do not show an improvement when using ∅-shot prompting together with Llama 2 7B Chat.

Results obtained from experimenting with Llama 2 7B and Llama 2 7B Chat exhibit a similar pattern to PaLM 2 and PaLM 2 for Chat, where we observe a higher performance increase in the base version compared to the chat version of the model within the same family and potentially the same size. As discussed earlier, these results also show that Llama 2 7B Chat is better at handling hallucination. This aligns with a study proposing Llama 2 (Touvron et al., 2023) where various measures have been incorporated, such as ensuring the quality and safety of the pre-training dataset, incorporating safety supervised fine-tuning for both pre-trained and fine-tuned models, and adding a safety-specific reward model during the reinforcement learning with human feedback (RLHF) pipeline for Llama 2 Chat.

From the results, we observe effectiveness across types of tasks from all safety measures in reducing hallucination, especially noting how effective the RLHF pipeline is for reducing hallucination. However, it is noteworthy that there is less effectiveness in math word problems where the model needs to respond with arbitrary number output, as can be observed from the increased performance for both models.

Furthermore, when comparing the increased performance in the reading comprehension task, we observe a notable increase in the performance of the pre-trained version, Llama 2 7B. This is likely due to the fact that the pre-trained model suffers more from longer context, i.e., it hallucinates more when engaging in long-context scenarios. However, we notice that the fine-tuned version, Llama 2 7B Chat, does not suffer from this trait, similar to PaLM 2 for Chat. Despite the resource constraint that prevented us from employing Llama 2 models at 13B and 70B sizes in our study, we conjecture that the effectiveness of our ∅-shot prompting relies more on the inherent hallucination of the model rather than its size. In other words, the higher the model is hallucinating, the more effective our approach is likely to be.

5.2 Null-Shot CoT Prompting

As 0CoT prompting shows promising increases in performance, we conduct an experiment to investigate if modifying the original 0CoT phrase to include an instruction similar to ∅-shot prompting will increase performance or not. The modified 0CoT phrase to incorporate ∅-shot prompting, named ∅CoT prompting, is shown in Appendix D. Similar to 0CoT prompting, we decide to place the ∅CoT phrase at the end of the task instruction and input. We conduct an experiment following the description in Section 3, but use 0CoT prompting as a baseline instead of zero-shot prompting. The results from the experiment are shown in Table 4 in Appendix D.

We observe that ∅CoT prompting rarely maintains effectiveness over 0CoT prompting. This could be due to the fact that both prompting approaches require the models to reason and explain in steps, and our ∅CoT prompting may hinder the abilities of the models to reason, resulting in subpar performance compared to 0CoT prompting. Alternatively, we can formulate one observation: eliciting the models to reason also decreases the chances of hallucination.

However, we also perceive that ∅CoT prompting is effective for PaLM 2 in reading comprehension as well as adversarial commonsense reasoning tasks and the Gemini Pro models in the AQuA-RAT dataset. As discussed earlier, base models tend to hallucinate more. By utilizing ∅CoT with long-context inputs, as in RACE datasets and a task that require long reasoning steps, as in WinoGrande, it may guide the models' hallucination in the correct path to the answer. This could also be the reason for the increased performance in the commonsense reasoning and reading comprehension tasks in PaLM 2 for Chat as well. As for the Gemini Pro models, we believe multiple choices in arithmetic questions may intervene with reasoning abilities.

We also note that ∅CoT prompting is very effective for GPT-4 Turbo, previously discussed in Section 4.2 as a model that seems to have less hallucination, in WinoGrande dataset. The sudden performance increase may suggest that this task likely require both reasoning (from the 0CoT part) and null examples (from the ∅-shot part), i.e., easier to elicit hallucination. In contrast, Gemini Pro performs better at handling hallucination in Strate-

Model	AQuA	GSM8K	StrategyQA	WinoGrande	RACE-m	RACE-h	ANLI	TriviaQA
Llama 2 7B	-18.46%	**10.53%**	**0.09%**	**-2.09%**	**12.60%**	**14.83%**	**3.80%**	**0.93%**
Llama 2 7B Chat	**-5.88%**	2.11%	-5.88%	-8.05%	-15.96%	-16.03%	-4.23%	-2.18%

Table 2: This table presents relative results indicating the performance changes compared to the zero-shot prompting baseline for Llama 2 7B and Llama 2 7B Chat.

gyQA and TriviaQA. As previously discussed, the models are better at detecting non-factual content in the ∅CoT phrase and decline to continue the task. Overall, strong models seem to suffer more from hallucination when using ∅CoT prompting for the AquA-RAT dataset.

5.3 Positions of Null-Shot Phrase

To determine the effects of the position of the ∅-shot phrase, we conduct an experiment following what is described in Section 3. However, we compare between placing the phrase before the task instruction, as in our original experiment, and at the end of the prompt, similar to what was done in Section 5.2. We only use the GPT-3.5 Turbo model to reduce the cost of the experiment. We compare the obtained performance against the same zero-shot prompting baseline as described in our main experiment. The results are shown in Table 5 in Appendix E.

We observe that placing the ∅-shot phrase at the beginning shows superior effectiveness across datasets, except for GSM8K, where models are required to produce arbitrary numeric answers. 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).

5.4 Prompt Components

To assess the contribution of each component in the ∅-shot phrase, we conduct an experiment similar to the one described in Section 3, but only use GPT-3.5 Turbo to save the cost. We decompose our ∅-shot phrase into two main components: "Look at examples in the "Examples" section" and "utilize examples and information from that section." We prepare three additional variants of the ∅-shot phrase, as shown in Appendix E. v1 and v2 remove the first and second components, respectively; and v3 removes both components. Results from the experiment are shown in Table 6 in Appendix E.

We notice that removing both components, as in v3, reduces the effectiveness of ∅-shot prompting on all datasets compared to the full ∅-shot phrase. 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 ∅-shot phrase provides the best performance, i.e., it requires both *look* and *utilize* instructions. Overall, the full ∅-shot phrase may provide the best balance as it encompasses all of the components, making it suitable across tasks.

6 Conclusion

We propose ∅-shot prompting, making use of hallucination in LLMs to improve performance across tasks. Our experiments show the effectiveness of our approach, especially in models with higher innate hallucination, specifically the base models. We also discuss the possibility of utilizing ∅-shot prompting for hallucination detection, which can repurpose existing natural language benchmarks for this purpose and does not require task-specific hallucination detection datasets, which are more costly to construct.

We also provide a discussion on the LLMs' behaviors elicited through ∅-shot prompting. Furthermore, we conduct various ablation studies exploring scaling effects, reasoning variants of ∅-shot prompting, effects of positioning the ∅-shot phrase, and the performance contribution of each component in the phrase. Future studies should explore the possibilities of utilizing this approach for detecting hallucination in LLMs and integrating this approach with other PE techniques.

Limitations

While the study that introduced 0CoT prompting (Kojima et al., 2022) used a two-stage prompting approach for improved result extraction, we do not utilize this approach in our study to reduce costs, which may result in some cases of unsuccessful result extraction. However, we compensate it with very flexible output extraction scripts instead (cf. Appendix B). As stated in Appendix C, we only utilize the 7B version of Llama 2 models, and implications made in this study may change with larger variants of Llama 2. However, we believe that it likely does not change our conclusions about the relationship between innate hallucination and increased performance across tasks.

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, it 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 ∅-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 inner workings of LLMs in general, which is an active area of research, utilizing ∅-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, et al. 2023. PaLM 2 Technical Report.

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, and Jared Kaplan. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback.

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.

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, and Dario Amodei. 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.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse Autoencoders Find Highly Interpretable Features in Language Models.

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.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Wes Gurnee and Max Tegmark. 2023. Language Models Represent Space and Time.

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.

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*.

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.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated Knowledge Prompting for Commonsense Reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169, Dublin, Ireland. 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.

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 Altenschmidt, et al. 2023. GPT-4 Technical Report.

OpenAI. 2022. Introducing ChatGPT.

OpenAI. 2023. New models and developer products announced at DevDay.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

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, et al. 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. 2023. 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.

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 Schnider, Christine von Däniken, and Klemens Gutbrod. 1996. The mechanisms of spontaneous and provoked confabulations. *Brain*, 119(4):1365–1375.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. *Nature*, 623(7987):493–498.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2023. "Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, 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, et al. 2023. Gemini: A Family of Highly Capable Multimodal Models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023a. PlanBench: An Extensible Benchmark for Evaluating Large Language Models on Planning and Reasoning about Change. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023b. On the Planning Abilities of Large Language Models - A Critical Investigation. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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.

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, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022b. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022c. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT.

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.

Appendix A Related Work

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 (Bai et al., 2022; 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.

PE: 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., 2022c), 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.

Additional context, e.g., article, context, and hypothesis
Question: {question}
Choices: {choices}
Answer:

Figure 3: 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

Table 3: 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. 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.

Appendix B Datasets

Figure 3 displays the format of task instructions and inputs for the datasets. This format is inspired by the procedure done in the 0CoT prompting study (Kojima et al., 2022). Choices and additional context are only provided in the prompts when applicable. All included datasets are in English. Additional details on the chosen testing set and the number of records are presented in Table 3.

We note that AQuA-RAT, WinoGrande, and TriviaQA are under the Apache License, Version 2.0. GSM8K and StrategyQA are under the MIT License. RACE datasets are avail-

able 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/

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, two 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.

Appendix C 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. Six LLMs included in the main experiment are PaLM 2 (`text-bison-001`), PaLM 2 for Chat (`chat-bison-001`), Gemini Pro (`gemini-pro`) via `generateContent` method, Gemini Pro (Chat) (`gemini-pro` via `start_chat` method), GPT-3.5 Turbo (`gpt-3.5-turbo-1106`), and GPT-4 Turbo (`gpt-4-1106-preview`). 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 the models. Furthermore, all of these models are relatively large and are utilized in many real-world products and scenarios.

PaLM 2 and PaLM 2 for 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 conversation. This could further give us a way 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 relatively larger in both parameter size and training data. This could provide insights into the effects of scaling models further. We include Gemini Pro because its performance is

∅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 4: The ∅CoT phrase shares the majority with the ∅-shot phrase. However, it has "step-by-step", highlighted in yellow, added at the end, inspired by 0CoT prompting.

likely positioned between that of GPT-3.5 Turbo and GPT-4 Turbo. All of these six aforementioned LLMs are utilized via their respective API-wrapper Python libraries[1].

Finally, Llama 2 7B[2] and Llama 2 7B Chat[3] are included for the scaling analysis where we attempt to evaluate ∅-shot prompting on smaller models. Due to resource constraints, we are only able to evaluate using the smallest variants of Llama 2 models (7B), despite the existence of the 13B and 70B variants. Both Llama 2 models are utilized via Hugging Face's `transformers`[4] pipelines, i.e., the `text-generation` pipeline for Llama 2 and the `conversational` pipeline for Llama 2 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. Llama 2 models are subject to Meta's community license agreement, which permits our use cases.

Appendix D Null-Shot CoT Phrase

This section contains phrases used for the ∅CoT prompting experiment and relative results of the experiment described in Section 5.2. The ∅CoT and 0CoT phrases are shown in Figures 4 and 5, respectively. The results from the experiment are presented in Table 4.

[1]GPT-3.5 Turbo and GPT-4 Turbo: https://github.com/openai/openai-python
PaLM 2, PaLM 2 for Chat, Gemini Pro, and Gemini Pro (Chat): https://github.com/google/generative-ai-python

[2]https://huggingface.co/meta-llama/Llama-2-7b-hf

[3]https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

[4]https://github.com/huggingface/transformers/

0CoT Phrase

Let's think step by step.

Figure 5: The 0CoT phrase used as a baseline for the experiment detailed in Section 5.2. This phrase is taken from Kojima et al. (2022).

∅-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 ∅-shot phrase with the first components highlighted in yellow and the second components highlighted in green.

Appendix E Null-Shot Phrase's Positioning and Null-Shot Phrase Variants

In this section, we present a result table, Table 5, of the experiment described in Section 5.3 to determine the position placement of the ∅-shot phrase. We also present prompt variants used for the experiment described in Section 5.4 to determine the effects of each component in Figures 7, 8, and 9, and their results in Table 6.

For the prompt variants, the original ∅-shot phrase is shown in Figure 2 and has been broken down into two main components, as shown in Figure 6. The first variant, `v1`, removes the first component. The second variant, `v2`, removes the second component. Finally, the third variant, `v3`, removes both components. The full phrases are shown in Figures 7, 8, and 9.

∅-shot Phrase: First Variant (v1)

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

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

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%**	-4.50%
PaLM 2 for Chat	-5.88%	-7.55%	**14.75%**	-4.49%	0.49%	0.79%	-2.49%	-2.12%
GPT-3.5 Turbo	-3.42%	-4.00%	-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%
Gemini Pro	**8.47%**	-9.99%	-98.42%	-99.62%	-3.02%	-1.06%	-7.07%	-98.54%
Gemini Pro (Chat)	8.06%	-11.66%	-98.42%	-99.62%	-2.32%	-0.73%	-8.01%	-98.55%

Table 4: This table presents relative results indicating the performance changes when using ∅CoT prompting compared to the 0CoT prompting baseline.

Prompting	AQuA	GSM8K	StrategyQA	WinoGrande	RACE-m	RACE-h	ANLI	TriviaQA
∅-shot	**33.94%**	15.19%	**3.14%**	**-1.84%**	**-1.79%**	**-1.19%**	**-3.61%**	**1.23%**
∅-shot-after	30.28%	**19.20%**	-6.21%	-69.08%	-4.81%	-4.62%	-46.47%	-3.09%

Table 5: This table presents relative results indicating the performance changes compared to the zero-shot prompting baseline. We find that placing the phrase before the task instruction and input allows GPT-3.5 Turbo to perform better in most cases.

∅-shot Phrase: Second Variant (v2)

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

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

∅-shot Phrase: Third Variant (v3)

Perform the following task as demonstrated in the "Examples" section.

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

Appendix F 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 F.1, F.2, and F.3, respectively.

F.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., 2022b) 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 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 ∅-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.

Prompting	AQuA	GSM8K	StrategyQA	WinoGrande	RACE-m	RACE-h	ANLI	TriviaQA
∅-shot	33.94%	15.19%	3.14%	-1.84%	-1.79%	-1.19%	-3.61%	**1.23%**
∅-shot-v1	**36.70%**	**16.85%**	2.73%	-3.95%	-2.12%	1.75%	-6.37%	0.62%
∅-shot-v2	10.09%	8.98%	**4.57%**	**1.84%**	**-1.14%**	**-0.80%**	**0.52%**	-0.37%
∅-shot-v3	27.52%	15.88%	1.23%	-8.82%	-1.47%	-1.43%	-2.07%	-0.25%

Table 6: This table presents relative results indicating the performance changes compared to the zero-shot prompting baseline. We note that each prompt component has its own strengths. In particular, v2, where the second prompt component is removed, shows the best performance in most datasets.

F.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 ∅-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 ∅-shot prompting to increase performances of LLMs by exploiting 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 ∅-shot prompting shows the possibility of uses in hallucination detection as well.

F.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 ∅-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 ∅-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., 2023). 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 G Empirical Results of Expected Behaviors When Using Null-Shot Prompting

This section contains qualitative examples generated by either GPT-4 Turbo or Gemini Pro from our main experiment and from ChatGPT web version. When we utilize ∅-shot prompting, these models may inform users 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 GPT-4 Turbo and Gemini Pro have the ability to inform users about its inaccessibility to the instructed null "Examples" section. This behav-

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 7: 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% (0)	64.7% (647)

Table 8: Number of instances when Gemini Pro's response includes a phrase informing the user about the unavailability of the instructed "Examples".

ior exhibits less hallucination compared to other models, specifically, context-conflicting hallucination. The numbers of instances for each dataset where this event occurred, i.e., GPT-4 Turbo starts the response with "I'm sorry, but" or Gemini Pro generates something similar to "The provided context does not contain", are presented in Tables 7, 8, and 9, for GPT-4 Turbo, Gemini Pro, and Gemini Pro (Chat), respectively. Additional qualitative examples are shown in Figures 10, 11, 12, 13, and 14.

G.1 Context-Conflicting Hallucination Detection Ability of GPT-4 Turbo

As can be observed from Table 7, GPT-4 Turbo is less prone to hallucination when using our ∅-shot prompting in StrategyQA and WinoGrande compared to other datasets, despite the fact that our ∅-shot prompting elicits context-conflicting hallucination. Typically, commonsense reasoning requires a use of implicit reasoning steps (Geva et al., 2021) or world knowledge (Levesque et al., 2012); performing this task may induced the model to utilize its associated weights of various reasoning types required by each question in the task. The use of reasoning may resulted 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.

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 9: Number of instances when Gemini Pro (Chat)'s response includes a phrase informing the user about the unavailability of the instructed "Examples".

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, eliciting GPT-4 Turbo to search the Internet or retrieve information from external sources, as this approach is common for this task (Yasunaga et al., 2021; Schick et al., 2023). As GPT-4 might 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 may 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 ∅-shot phrase, as we instructed the model to further look into something that sounds promising to exist given the long context. 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.

G.2 Context-Conflicting Hallucination Detection Ability of Gemini Pro

Similar to what can be observed with GPT-4 Turbo, Gemini Pro is able to detect context-conflicting hallucination in datasets, as shown in Tables 8 and 9. 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 Pro and Gemini 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 ∅-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 information generation in closed-book question answering tasks (Team et al., 2023).

G.3 Inability of Other LLMs to Detect Hallucination

One potential reason why other models could not detect context-conflicting hallucination when using our ∅-shot prompting could be due to the fact that these models are smaller compared to GPT-4 Turbo or Gemini Pro. In a previous study, it showed that smaller models may exhibit fewer reasoning capabilities and more hallucinated behaviors (Wei et al., 2022c). Therefore, even Llama 2 models were trained using a process designed to minimize hallucination, they exhibited more hallucinatory behaviors, likely due to the lack of scale.

As for PaLM 2 models and GPT-3.5 Turbo, it is unclear how their scale is comparable to GPT-4 Turbo or Gemini Pro due to a lack of public report. Nevertheless, it is worth noting that GPT-3.5 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 through the ChatGPT website is shown in Figure 15. The inconsistency in behaviors between GPT-3.5 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 H Error Analysis

We investigate failure cases of the main experiment. 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 2[5] models and Gemini Pro[6] models. Our further investigations also validate that GPT-3.5 Turbo and GPT-4 Turbo do not have this behavior. Table 10 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 ∅-shot or ∅CoT prompting decreases the effectiveness of the built-in safety mechanisms in the majority of cases. As shown in Table 10, the red color highlights the decrease in the numbers of empty responses across datasets when eliciting through ∅-shot or ∅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 ∅-shot and ∅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., 2023), 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 ∅-shot and ∅CoT prompting, we notice that ∅-shot prompting is more effective in breaking built-in safety measures, which is likely attributed to the fact that ∅CoT prompting induces reasoning during decoding. Therefore, we posit the same reasons for the reduced effectiveness of ∅CoT prompting in general; reasoning reduces the chances of hallucination. In addition, we observe that PaLM 2 for Chat generates fewer empty responses than PaLM 2 in general. In contrast, Gemini Pro is more consistent across text and chat generation. This observation may result from the fact that PaLM 2 and PaLM 2 for Chat are two different models, while both Gemini Pro variants are based on the same model. Nevertheless, we cannot confirm this fact due to a lack of public report.

[5] https://cloud.google.com/vertex-ai/docs/generative-ai/configure-safety-attributes-palm

[6] https://ai.google.dev/docs/safety_setting_gemini

Look at examples in the "Examples" section and utilize examples and information from that section to perform the following task.
Question: How much does it cost you for lunch today at Subway if you pay $40 for a foot-long fish sub and thrice as much for a six-inch cold-cut combo sub?
Answer:

I'm sorry, but there seems to be a misunderstanding. I don't have access to an "Examples" section or any other external content. However, I can help you calculate the cost based on the information you've provided.

If you paid $40 for a foot-long fish sub at Subway, and a six-inch cold-cut combo sub costs thrice as much, then the cost of the six-inch cold-cut combo sub would be:

3 times $40 = $120

So, the total cost for lunch at Subway would be the cost of the foot-long fish sub plus the cost of the six-inch cold-cut combo sub:

$40 (foot-long fish sub) + $120 (six-inch cold-cut combo sub) = $160

Therefore, it would cost you **$160** for lunch today at Subway based on the prices you've provided. However, please note that these prices seem unusually high for Subway sandwiches, and it's likely that there's been a mistake or miscommunication regarding the actual costs.

Figure 10: An output generated using ∅-shot prompting with GPT-4 Turbo for GSM8K dataset.

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 11: An output generated using ∅-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: 'When Will You Marry?' became the world's most expensive work of art when it was sold
Answer:

I'm sorry, but as an AI, I don't have the ability to look at sections or examples outside of the text provided to me in our conversation. However, I can tell you that the painting "When Will You Marry?" by **Paul Gauguin** was sold for a record price of nearly $300 million in 2015. At that time, it was reported to be one of the most expensive works of art ever sold. Please note that the art market is dynamic, and records for the most expensive works of art can change over time.

Figure 12: An output generated using ∅-shot prompting with GPT-4 Turbo for TriviaQA 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 13: An output generated using ∅-shot prompting with Gemini 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: "Which actor, best known for his role in ""The Godfather"" series of films, shares his name with a member of TV's ""Dragon's Den""?" Answer:

The context does not mention anything about actors in "The Godfather" series of films or members of TV's "Dragon's Den", so I cannot answer this question from the provided context.

Figure 14: An output generated using ∅-shot prompting with Gemini Pro (Chat) for TriviaQA 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 15: An output generated using ∅-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

Model	AQuA				GSM8K			
	0S	∅S	0CoT	∅CoT	0S	∅S	0CoT	∅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 for Chat	0.39% (1)	0% (0)	0% (0)	0% (0)	0.23% (3)	0.3% (4)	0.23% (3)	0.23% (3)
Gemini 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 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	∅S	0CoT	∅CoT	0S	∅S	0CoT	∅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 for 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 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 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	∅S	0CoT	∅CoT	0S	∅S	0CoT	∅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 for 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 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 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	∅S	0CoT	∅CoT	0S	∅S	0CoT	∅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 for 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 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 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 10: 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, ∅S, 0CoT, and ∅CoT denote zero-shot prompting, ∅-shot prompting, zero-shot CoT prompting, and ∅-shot CoT prompting, respectively.

Appendix I Absolute Results

This section provides absolute results for the main experiment and ablation studies. Table 11 presents absolute results from the main experiment. Tables 12, 13, 14, and 15 provide absolute results for the scaling analysis, the reasoning variant, the position analysis, and the prompt components analysis, respectively.

Appendix J Qualitative Examples of Null-Shot Prompting

In this section, we provide qualitative examples of generated responses from the datasets when utilizing ∅-shot prompting. The LLM used to generate each response is denoted in the figure caption. Figures 16, 17, 18, 19, 20, 21, 22, and 23 are examples of AQuA-RAT, GSM8K, StrategyQA, WinoGrande, RACE-m, RACE-h, ANLI, and TriviaQA, respectively.

Appendix K Raw Data and Source Code

Raw data and source code will be made public upon acceptance.

Appendix L 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.

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 for Chat	14.96	15.75	53.90	55.12	57.73	58.69	56.59	60.54
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	74.30	73.16	74.85	61.83	73.48	55.80
Gemini Pro	25.59	35.43	51.55	66.49	67.03	50.66	63.85	62.98
Gemini Pro (Chat)	25.59	37.01	52.39	67.02	67.60	50.44	63.69	62.98
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	50.50	64.20	68.70
PaLM 2 for Chat	73.54	74.30	67.04	67.50	42.67	43.33	70.20	70.10
GPT-3.5 Turbo	85.38	83.84	81.73	80.76	48.42	46.67	81.00	82.00
GPT-4 Turbo	92.97	93.25	88.59	88.97	64.17	64.00	85.40	84.60
Gemini Pro	83.01	84.61	77.82	79.47	50.58	51.67	70.20	25.30
Gemini Pro (Chat)	84.61	85.24	79.10	80.39	51.08	51.92	70.50	25.40

Table 11: Absolute results of the main experiment. **0S** denotes zero-shot prompting, **∅S** denotes ∅-shot prompting. Underline signifies cases where ∅-shot prompting results outperform its zero-shot prompting baseline.

Model	AQuA		GSM8K		StrategyQA		WinoGrande	
	0S	∅S	0S	∅S	0S	∅S	0S	∅S
Llama 2 7B	25.59	20.87	2.88	3.18	50.79	50.83	45.22	44.28
Llama 2 7B Chat	52.75	49.65	21.53	21.99	52.75	49.65	49.01	45.07
Model	**RACE-m**		**RACE-h**		**ANLI**		**TriviaQA**	
	0S	**∅S**	**0S**	**∅S**	**0S**	**∅S**	**0S**	**∅S**
Llama 2 7B	27.09	30.50	23.90	27.44	32.92	34.17	64.80	65.40
Llama 2 7B Chat	57.17	48.05	46.37	38.94	35.50	34.00	64.30	62.90

Table 12: Absolute results of the scaling analysis experiment. **0S** denotes zero-shot prompting, **∅S** denotes ∅-shot prompting. Underline signifies cases where ∅-shot prompting results outperform its zero-shot prompting baseline.

Model	AQuA		GSM8K		StrategyQA		WinoGrande	
	0CoT	∅CoT	0CoT	∅CoT	0CoT	∅CoT	0CoT	∅CoT
PaLM 2	35.43	16.14	60.20	43.52	62.71	59.34	63.93	76.01
PaLM 2 for Chat	13.39	12.60	58.23	53.83	52.40	60.13	59.83	57.14
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	64.80
Gemini Pro	46.46	50.39	69.07	62.17	66.42	1.05	62.83	0.24
Gemini Pro (Chat)	48.82	52.76	70.89	62.62	66.42	1.05	62.43	0.24
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	66.70	63.70
PaLM 2 for Chat	71.59	71.94	65.04	65.55	43.50	42.42	70.60	69.10
GPT-3.5 Turbo	83.36	80.57	77.67	77.79	42.25	22.42	80.30	80.90
GPT-4 Turbo	71.59	67.55	61.75	56.46	52.17	47.58	86.20	85.30
Gemini Pro	83.15	80.64	78.47	77.64	48.33	44.92	61.60	0.90
Gemini Pro (Chat)	84.05	82.10	78.36	77.79	48.92	45.00	62.00	0.90

Table 13: Absolute results of the ∅CoT prompting experiment. **0S** denotes zero-shot prompting, **∅S** denotes ∅-shot prompting. Underline signifies cases where ∅-shot prompting results outperform its zero-shot prompting baseline.

Prompting	AQuA	GSM8K	StrategyQA	WinoGrande	RACE-m	RACE-h	ANLI	TriviaQA
0-shot	42.91	54.89	64.02	59.98	85.38	81.73	48.42	81.00
∅-shot	57.48	63.23	66.03	58.88	83.84	80.76	46.67	82.00
∅-shot-after	55.91	65.43	60.04	18.55	81.27	77.96	25.92	78.50

Table 14: Absolute results of the effects of positions experiment. **0S** denotes zero-shot prompting, **∅S** denotes ∅-shot prompting. Underline signifies cases where ∅-shot prompting results outperform its zero-shot prompting baseline.

Prompting	AQuA	GSM8K	StrategyQA	WinoGrande	RACE-m	RACE-h	ANLI	TriviaQA
0-shot	42.91	54.89	64.02	59.98	85.38	81.73	48.42	81.00
∅-shot	57.48	63.23	66.03	58.88	83.84	80.76	46.67	82.00
∅-shot-v1	58.66	64.14	65.76	57.62	83.57	80.30	45.33	81.50
∅-shot-v2	47.24	59.82	66.94	61.09	84.40	81.07	48.67	80.70
∅-shot-v3	54.72	63.61	64.80	54.70	84.12	80.56	47.42	80.80

Table 15: Absolute results of the effects of components experiment. **0S** denotes zero-shot prompting, **∅S** denotes ∅-shot prompting. Underline signifies cases where ∅-shot prompting results outperform its zero-shot prompting baseline.

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 16: An output generated using ∅-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 17: An output generated using ∅-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 18: An output generated using ∅-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 19: An output generated using ∅-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,“AAAhhhhhhhhh!!!” 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 20: An output generated using ∅-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 21: An output generated using ∅-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 22: An output generated using ∅-shot prompting with PaLM 2 for 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 23: An output generated using ∅-shot prompting with GPT-3.5 Turbo for TriviaQA.