International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Cost, Complexity, and Efficacy of Prompt Engineering Techniques for Large Language Models

Milind Cherukuri

Business Analyst, Caris Life sciences
cherukurimilind@gmail.com

Abstract
This research investigates the impact of various prompt en- gineering techniques on the length, cost, complexity, and accuracy of responses from large language models (LLMs). By comparing direct prompting with zero-shot, few-shot, and chain-of-thought (CoT) meth- ods on tasks like GSM8K and creative writing, I analyze the trade-offs between token usage and response quality. Results show that while zero- shot CoT prompting is highly effective and cost-efficient, other meth- ods like Least-to-Most and Tree-of-Thought add significant length and complexity without proportional accuracy gains. Additionally, I discuss the financial implications, finding that GPT-4's unique pricing struc- ture narrows the cost difference between manual/few-shot and zero-shot methods. Complexity analysis reveals that more intricate prompts often lead to convoluted outputs, challenging human review and implementa- tion. Our findings guide the selection of prompt engineering strategies to optimize both performance and resource utilization.

1. Introduction

Prompt engineering, the practice of developing specialized prompts and queries to improve the performance of Large Language Models, is a prominent topic of interest in the NLP community and among the general public. The practice is believed to allow for improvements in LLM performance in a variety of domains without investment in underlying training [21]. It is not, however, without its critics. Some commentators believe that the practice will become irrelevant as models grow larger and more powerful, becoming more capable of directly inter- preting a user's intent without error [4]. Others question the need for specialized professionals or training to attain minimal improvements, which are often not repeatable across contexts [13].

Despite such controversy, it is difficult to find empirical analyses of the trade- off between costs and performance benefits associated with advanced prompting. Papers introducing new prompting techniques often only include performance benchmarks concerning the techniques' efficacy, typically within a limited do- main. Some authors briefly mention problems associated with involved prompt- ing, such as the human time and effort induced by increased complexity and limitations on creativity and randomness [5], and others suggest the automation of prompting to avoid some of these costs [14]. It is known that token costs, degradation of quality with increased

prompt context lengths, and the uncertain nature of performance gains are all important practical considerations [36]. How- ever, the extent of these issues for various techniques, so as to enable standardized comparison between them and with a control baseline (no special prompting), has not been (to my knowledge) quantified.

The quality, length, and complexity of LLM responses have been analyzed within several individual task and technique domains. Some research with GPT- 3 series models on math and non-math reasoning tasks suggests the addition of length and complexity through the introduction of extra reasoning steps for both input prompts and output responses improves accuracy when using chain- of-thought prompting techniques [37]. Effects are on the magnitude of several points of accuracy per added step, with generally low costs as long as prompt examples are selected carefully. Uniform improvements from complex chain-of- thought prompting are not fully accepted in the literature however - other work has noted a tendency for the method to lead to worse performance on simple questions [38].

Complexity has also been studied within the context of prompting for sum- marization and story-generation tasks. The Chain of Density prompting tech- nique seeks to optimize the named entity density of LLM-generated summaries through choice from a series of repeated, increasingly dense iterations [27]. Hu- man preferences tend to align with 0.1 - 0.15 named entities per token, a point near the middle of the usual sequence of generations, demonstrating the exis- tence of a tradeoff between informativeness and clarity. At the same time, other work has shown is difficult to control language model output complexity, mean- ing that the choice of specific techniques is important. Research demonstrates current models are not yet able to achieve compliance with desired readability instructions for the tasks of story generation, simplification, and summarization. However, a small amount of improvement is achievable through careful prompt word choice and the use of few-shot examples [28, 29].

This paper uses several metrics to evaluate the benefits and drawbacks of prompt engineering methods systematically and analyzes the tradeoffs inher- ent in their application to standardized data. Such an assessment is valuable on several dimensions. Beyond quantifiably testing the practicality of prompt engineering as a whole, it can be used to compare the performance of different approaches. It is useful in a world where so many competing techniques are avail- able. I also provide a newly constructed dataset summarizing the wide variety of existing methods and data on their popularity as measured by Semantic Scholar citations, which may be useful for future surveys of the field. Next, I offer a new look at these prompting methods in a period long after their introduction. The current environment is one in which far greater capabilities are native to underlying foundation models. Finally, I introduce and adapt some useful mea- sures of accuracy, quality, length, and complexity, such as inter-paragraph cosine similarity and the ratio of interaction length with prompting to the length of an accepted human-generated answer, to the challenge of LLM evaluation.

2. Data

To evaluate performance, I attempt to use tasks that are general-purpose, close to real-world applications, and standardized in the literature. To this end, I selected the GSM8K dataset, a collection of elementary-level math word problems [**?**], and a creative writing task involving the generation of a coherent two-paragraph passage with random, predetermined ending sentences for each paragraph. The original creative writing task in [12] uses four paragraphs and sentences. How- ever, this is too difficult for older models and the manual production of good answer demonstrations - I take the first two sentences for each original question. These tasks carry several key benefits. They cover both the mathematical and linguistic domains - two types of tasks that form the foundation of the standardized testing of humans. GSM8K is studied in the majority of the pa- pers introducing techniques used in this paper. It is among the most common datasets used in the far larger list of papers I initially surveyed. Text and story generation is a widespread foundational task in NLP. Importantly, these sets of tasks are known to be free of data contamination. The GSM8K test set has been intentionally withheld from the training of OpenAI's models [6]. The creative writing task was released only in 2023, and the code and data provided with the
associated paper include only questions and not LLM responses. [1]

I performed the analysis on one cutting-edge model and one older model from closer to the time that these techniques were introduced. This provides a picture of the changing costs and benefits of advanced prompting, a trend that may even be extrapolated into the future if current LLM scaling laws continue to hold. As the most widely used models and the ones behind much original work in the field, I select two models from the OpenAI series: GPT-4 (June 13th, 2023 version: 'GPT-4-0613') and text-DaVinci-003. All conversations were conducted programmatically via the OpenAI API.

3. Prompting Methods Assessed

The following methods were selected based on their popularity and ease of im- plementation. They are listed in order of initial paper release/discovery date below (according to Semantic Scholar - citation publication dates below may be for revisions).

– Few-Shot Prompting: The prompter provides a few examples of questions and correct answers before the main question/task. Notably featured in [30].

[1]It is also relatively simple to find random sentence generators online (such as https: //www.thewordfinder.com/random-sentence-generator/), which would work for this task - at the cost of comparability with the results of [12]. In any case, the 2-sentence task represents a modification of the original, and the test overall makes use of subjective human evaluations of the coherence of generated stories, limiting comparisons anyway - so perhaps testing the task on more truly novel pairings is a promising direction for future work.

- Few-Shot (Manual) Chain-of-Thought Prompting: The model provides worked examples of answers in which the reasoning steps are written out. [17] Note, however, that such steps are not explicitly planned out or mentioned, as is the case in least-to-most prompting.
- Least-to-Most Prompting: The model is given few-shot examples that demon- strate how first to break down the task into smaller and simpler subproblems and then solve them sequentially. [8] This is a few-shot and chain-of-thought method.
- Zero-Shot Chain of Thought Prompting (Original): Initial advances in the chain of thought prompting to improve reasoning were achieved by simply including the following before the question/task: ”Let’s think step by step.”
  [9] For the creative writing task, the prompt following the question/task is adapted to: ”Plan step-by-step before writing the passage.” [2]
- Zero-Shot Chain of Thought Prompting (Automatic Prompt Engineer): Au- tomated testing has indicated that an optimal zero-shot Chain of Thought prompt is ”Let’s work this out in a step-by-step way to be sure I have the right answer.” [23] For the creative writing task, the prompt following the question/task is adapted to: ”Plan step-by-step before writing the passage to be sure I have a correct and coherent answer.” [3]
- Self-Refine Prompting: The model produces an initial response and then is prompted for feedback, which it uses for refinement. [11] This is an iterative method.
- Tree of Thought Prompting: The language model traverses a tree of deci- sions - choosing among multiple steps or ideas it has generated to conclude. Backtracking is possible. [12] This is an iterative method.
- Zero-Shot Control Baseline/Direct Prompting: This method consists of just providing the question/task directly.

4. Analyses

I provide performance scores as well as summary statistics (mean, standard deviation) of the metrics discussed below for each prompting method by model by question/task type. Statistical inference is uncommon in the prompting literature
- most papers simply report performance scores. [4] However, in this work, I do check for significant differences in metrics between prompting methods and the direct prompting baseline, reporting at the 95% level. For GSM8K accuracy

[2] In initial experiments, this modification was found necessary to elicit any step-by- step behavior from the model.

[3] Again, in initial experiments, this modification was found necessary to elicit any step-by-step behavior from the model.

[4] Effect sizes and sample sizes are usually sufficiently large to merit confidence in the statistical significance of results. All paired t-tests comparing prompting methods and human responses were significant in the one paper I did find with inference, [28].

scores, I performed McNemar's tests. [5] For sample means of creative writing scores and other metrics, I perform paired t-tests. [6]

5. Quality and Accuracy

Results concerning the accuracy and/or quality of each method can be found in Table 1.
For GSM8K problems, I report correct/incorrect accuracy at the point a technique has been fully implemented (the end of the tree of thought or, after all, Self-Refinement, etc.). Chain-of-thought methods (including Least-to-Most) provide substantial gains for this task when implemented both manually and with a zero-shot approach. [7] Few-shot prompting - implemented in this paper as the provision of questions and answers without reasoning - is ineffective and even harmful. The iterative Self-Refine and Tree-of-Thought methods are generally well suited to this task, with the exception of Self-Refine checking when imple- mented with GPT-4, which can bring results up to the level of chain-of-thought methods.

My initial analysis of creating writing coherence consisted of the author's as- sessment of generated passage coherence (on a scale of 1 to 10, 1 being incoherent and 10 being very coherent) based on several guidelines - unconnected ideas or abrupt changes in characters or settings received low scores, and opposite cases received high scores.
Grader coherence scores are inherently noisy, as noted in [14], but I take several measures to alleviate this problem.

In this and other analyses, I provide adjustments for scores accounting for whether or not the task constraints were followed

- whether responses did, indeed, contain two paragraphs with specified ending sentences. 10 For human-preferred scores, adjustments for non-compliance give the outcomes in the case in which non-compliant responses are reduced to the lowest possible score.

Figure 1 shows the number of conversations where each method was one of the most preferred. It appears to be difficult to attain any gains from prompt engineering on this task, at least as far as human preferences for coherence are concerned. The provision of few-shot examples indicating preferences seems to be a minimally consistently promising approach, while other, more complex methods may be detrimental.

For my main results on creative writing, however, I introduce more objective measures of passage coherence that give a good sense of the concept on various levels - a novel approach for this creative writing task. To assess local coherence between sentences, I compute the average cosine similarity between consecutive sentence-level BERT embeddings (all-distilroberta- v1) [22], [23].

For passage with n sentences indexed by i, each with embedding si, average inter-sentence cosine similarity is given by n−1 First, I limit reported results for these grades to information concerning which method was preferred for each model and task question comprising a set of ending sentences. This limits

$$CS_{IS} = \frac{1}{n-1} \sum_{i=1}^{n-1} cos(s_i, s_{i+1})$$

demands on the data to only the ordinal ranking of methods. Second, I check the consistency of my methodology and results by soliciting a secondary opinion. I fine-tuned GPT-3.5 to learn my methodology for grading passage coherence (see Appendix Section ?? for prompts) - a difficult task, given the slippery nature of coherence as a concept and the wide variety of LLM responses and associated formats resulting from different models and methods. 8 I created two fine-tuned models, each trained and validated on a randomly selected 50% fold of responses (stratified by method) and human coherence scores. I deployed each model on the other 50% fold. The results demonstrate the consistency and learnability of human preferences, for 76.5% of model-sentence pairings, human and automated scores agree on at least one of the methods as among For a more global measure, I compute embeddings for each paragraph of the LLM response by averaging the sentence embeddings and then the average cosine similarity between these paragraph embeddings for consecutive paragraphs. [11] I believe this custom method best captures success on this task, as outlined in the initial prompt and question, and it produced logical results (see Appendix Section **??** for examples to aid in interpretation) that also had the highest correlation with human scores of any metric. [12] It is my preferred metric for the rest of this paper.

For passage with p paragraphs indexed by j each with number of sentences l_j, comprised of sentences indexed by i each with embedding $s_{j,i}$, average inter-paragraph cosine similarity is given by the top preferred. Simulations using empirical probabilities of scores under independence indicate that this would occur in only 21.84% of cases by chance. [9]

$$CSIP = \frac{1}{p-1}\sum_{j=1}^{p-1} \cos\left(\frac{1}{l_j}\sum_{i=1}^{l_j} s_{j,i}, \frac{1}{l_{j+1}}\sum_{i=1}^{l_{j+1}} s_{j+1,i}\right)$$

[5]The same questions are administered for each prompting method, so there is a dependence that this paired test accounts for.

[6]Some other metrics remain, such as the token cost of performance or change in performance divided by change in tokens for each method versus direct prompting. Bootstrapping confidence intervals for this sort of value seems possible, but the dependent nature of the data poses challenges.

[7]Surprisingly, APE-Improved Zero-Shot CoT is not more effective than the simple "Let's Think Step by Step approach, as was the case in [20].

[8]GPT-4 has been prompted to produce scalar scores for this task [14], but I generally found its grading to be inconsistent, even with few-shot prompts providing a few examples included.

[9]Contact the author for simulation details.

[10]I additionally tried to use GPT-4 to assess adherence to the original instructions - to check if the exact sentences specified in the prompt were used. In my initial experiments - in contrast to [14] - I found that GPT-4 was not able to do this, repeatedly missing deviations of one or a few words, even when told to perform the check in a step-by-step manner carefully! I instead implemented simple logic using regular expressions, paired with a small amount of manual cleaning, and recommend this approach for future work.

[11]Other structural methods for assessing global coherence were considered but not implemented due to their limited additional value and overall feasibility for short, 2-paragraph

passages.

[12]Contact the author for details on correlations.

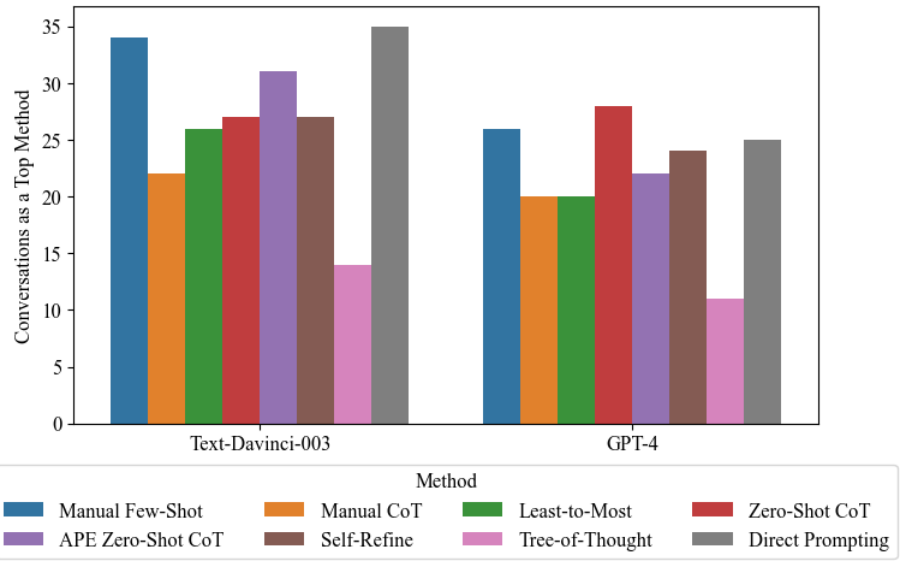
TABLE I: Mean and Standard Deviation of Accuracy/Quality Scores

Task	Metric	Model	Manual Few-Shot (May 2020)	Manual CoT (Jan 2022)	Least-to-Most (May 2022)	Zero-Shot CoT (May 2022)	APE Zero-Shot CoT (Nov 2022)	Self-Refine (Mar 2023)	Tree-of-Thought (May 2023)	Direct Prompting
GSM8K	Accuracy	Text-Davinci-003	0.18	0.6*	0.58*	0.62*	0.49*	0.2	0.23	0.23
		GPT-4	0.49*	0.93*	0.95*	0.95*	0.93*	0.89*	0.4*	0.73
Creative Writing	Average Inter-Sentence Cosine Similarity	Text-Davinci-003	0.364 (0.063)	0.345 (0.061)	0.357 (0.059)	0.366 (0.077)	0.382* (0.077)	0.369 (0.064)	0.357 (0.096)	0.363 (0.064)
		GPT-4	0.346 (0.062)	0.35* (0.066)	0.341 (0.061)	0.347* (0.057)	0.35* (0.066)	0.356* (0.06)	0.351* (0.061)	0.333 (0.049)
	Average Inter-Paragraph Cosine Similarity	Text-Davinci-003	0.476* (0.163)	0.479* (0.161)	0.48* (0.162)	0.41* (0.192)	0.42* (0.192)	0.366 (0.179)	0.43 (0.222)	0.359 (0.178)
		GPT-4	0.386* (0.146)	0.422 (0.147)	0.422 (0.151)	0.465* (0.151)	0.464* (0.148)	0.412 (0.161)	0.447 (0.145)	0.42 (0.158)
	Average Inter-Sentence Cosine Similarity (Compliance	Text-Davinci-003	0.363 (0.064)	0.351 (0.063)	0.369 (0.059)	0.386 (0.076)	0.391* (0.072)	0.393 (0.054)	0.397 (0.104)	0.368 (0.072)

	Adjusted)									
		GPT-4	0.349 (0.058)	0.355* (0.069)	0.35* (0.06)	0.344 (0.059)	0.356 (0.065)	0.351 (0.059)	0.348 (0.061)	0.334 (0.046)
	Average Inter-Paragraph Cosine Similarity (Compliance Adjusted)	Text-Davinci-003	0.433* (0.177)	0.401 (0.191)	0.414 (0.168)	0.357 (0.182)	0.389 (0.183)	0.34 (0.204)	0.224 (0.143)	0.371 (0.178)
		GPT-4	0.366* (0.154)	0.423 (0.15)	0.404 (0.159)	0.463* (0.155)	0.449 (0.159)	0.398 (0.158)	0.455 (0.143)	0.42 (0.157)
	Compliance	Text-Davinci-003	0.43	0.19*	0.25*	0.43	0.44	0.32*	0.04*	0.5
		GPT-4	0.63	0.51	0.52	0.57	0.56	0.48	0.26*	0.56

Stars indicate a significant difference from the direct prompting baseline from McNemar's tests at the 95% level for GSM8K accuracy scores and creative writing task compliance scores and paired t-tests at the 95% level for all other metrics.

Fig. 1: Human-Preferred Method by Model - Creative Writing Coherence - Ad- just for Non-Compliance



For both cosine similarity measures, I adjust for task compliance by removing non-compliant

conversations.

These measures are again reported in Table I. It isn't easy to attain much improvement in inter-sentence cosine similarity, particularly when accounting for task compliance, but chain-of- thought methods and the APE Zero-Shot CoT method (adapted to include an extra request for correctness and coherence for this task) perform well. Self-refinement also appears to be somewhat helpful. Gains from prompting are larger for inter-paragraph cosine similarity, and chain-of-thought methods (including least-to-most) tend to fairly consistently outperform, with some improvement coming from simple few- shot prompting on older models. Task compliance rates are low for both new and old models, suggesting careful adherence to complex instructions is an area with room for significant improvement in future models and methods. No prompt engineering method demonstrates substantial improvement in task compliance, and prompting often leads to actively worse performance - potentially distracting models or adding additional complication.

Do larger/more modern models benefit more from prompt engineering, or are the techniques becoming obsolete? Gains from prompting on GSM8K questions have fallen as default performance on the task has improved dramatically. The creative writing task, however, is not a solved problem, and yet the gains from prompting still appear to be larger for older models - though task compliance also seems to suffer more under almost all techniques. Earlier evidence demonstrated that gains from few-shot learning increased with model scale - but this paper does not seem to support proof that the trend has held. [16] Few-shot benefits have, for the most part, decreased or flatlined with model improvement.

One might expect more recent prompt engineering techniques (as measured by paper release/publication date) to be more powerful and useful, but the evidence shows this is not the case. Chain-of-thought prompting, popularized throughout mid- late 2022 (and including least-to-most prompting) seems to have been the most significant innnovation not only for math- based reasoning tasks as might be expected, but also for the language-based creative writing challenge. The culmination of chain-of-thought improvements in zero-shotting is able to attain comparable and sometimes superior results relative the provision of manual/few-shot chain-of-thought examples. Few- shotting does still have its uses, however, particularly for creative writing tasks and when working with older models, where the construction of a coherent passage (a potentially subjective target) can be demonstrated. Complex iterative techniques such as Self-Refinement and traversal of the tree of thoughts, both introduced in 2023, largely do not lead to improvement (or attain less improvement than other methods), at least for these tasks. They may be appropriate only for some more specialized applications and can only be successfully applied for the most recent models. 13

13Inspection of results found significant non-compliance with prompt instructions for these methods, especially with older models.

I can see that a method's performance seems fairly repeatable across tasks, but what about the variability of performance within a task? Very high accuracy/compliance scores indicate consistent performance and are present for only a few models and methods—perhaps only GPT-4 with one of the chain of thought methods can be relied upon, and only for the GSM8K task. Though inter-sentence results are consistent, variability in inter-paragraph cosine similarity is moderately large, and it is not entirely clear whether any method is sufficiently reliable.

Grader coherence scores are inherently noisy, as noted in [12], but I take several measures to alleviate this problem. First, I limit reported results for these grades to information concerning which method was preferred for each model and task question comprising a set of ending sentences. This limits de- mands on the data to only the ordinal ranking of methods. Second, I check the consistency of my methodology and results by soliciting a secondary opinion. I fine-tuned GPT-3.5 to learn my methodology for grading passage coherence
- a difficult task, given the slippery nature of coherence as a concept and the wide variety of LLM responses and associated formats resulting from different models and methods. [8] I created two fine-tuned models, each trained and vali- dated on a randomly selected 50% fold of responses (stratified by method) and human coherence scores. I deployed each model on the other 50% fold. The results demonstrate the consistency and learnability of human preferences, for of model-sentence pairings, human and automated scores agree on at least one of the methods as among the top preferred. Simulations using empirical probabili- ties of scores under independence indicate that this would occur in only of cases by chance. [9]

[8] GPT-4 has been prompted to produce scalar scores for this task [12], but I generally found its grading to be inconsistent, even with few-shot prompts providing a few examples included.
[9]Contact the author for simulation details.

In this and other analyses, I provide adjustments for scores accounting for whether or not the task constraints were followed - whether responses did, indeed, contain two paragraphs with specified ending sentences. [10] For human-preferred scores, adjustments for non-compliance give the outcomes in the case in which non-compliant responses are reduced to the lowest possible score.

Figure 1 shows the number of conversations where each method was one of the most preferred. It appears to be difficult to attain any gains from prompt engineering on this task, at least as far as human preferences for coherence are concerned. The provision of few-shot examples indicating preferences seems to be a minimally consistently promising approach, while other, more complex methods may be detrimental.

For my main results on creative writing, however, I introduce more objective measures of passage coherence that give a good sense of the concept on various levels - a novel approach for this creative writing task. To assess local coherence between sentences, I compute the average cosine similarity between consecutive sentence-level BERT embeddings (all-distilroberta-v1) [39, 40].

For passage with n sentences indexed by i, each with embedding s_i, average inter-sentence cosine similarity is given by For a more global measure, I compute embeddings for each paragraph of the LLM response by averaging the sentence embeddings and then the average cosine similarity between these paragraph embeddings for consecutive paragraphs. [11] I believe this custom method best captures success on this task, as outlined in the initial prompt and question, and it produced logical results for examples to aid in interpretation) that also had the highest correlation with human scores of any metric. [12] It is my preferred metric for the rest of this paper.

For passage with p paragraphs indexed by j each with number of sentences l_j, comprised of sentences indexed by i each with embedding $s_{j,i}$, average inter- paragraph cosine similarity is given by[10]I additionally tried to use GPT-4 to assess adherence to the original instructions
- to check if the exact sentences specified in the prompt were used. In my initial experiments - in contrast to [12] - I found that GPT-4 was not able to do this, repeatedly missing deviations of one

or a few words, even when told to perform the check in a step-by-step manner carefully! I instead implemented simple logic using regular expressions, paired with a small amount of manual cleaning, and recommend this approach for future work.

[11] Other structural methods for assessing global coherence were considered but not implemented due to their limited additional value and overall feasibility for short, 2-paragraph passages.

[12] Contact the author for details on correlations.

For both cosine similarity measures, I adjust for task compliance by removing non-compliant conversations.

These measures are again reported in Table 1. It isn't easy to attain much improvement in inter-sentence cosine similarity, particularly when accounting for task compliance, but chain-of-thought methods and the APE Zero-Shot CoT method (adapted to include an extra request for correctness and coherence for this task) perform well. Self-refinement also appears to be somewhat helpful. Gains from prompting are larger for inter-paragraph cosine similarity, and chain- of-thought methods (including least-to-most) tend to fairly consistently outper- form, with some improvement coming from simple few-shot prompting on older models. Task compliance rates are low for both new and old models, suggesting careful adherence to complex instructions is an area with room for significant improvement in future models and methods. No prompt engineering method demonstrates substantial improvement in task compliance, and prompting often leads to actively worse performance - potentially distracting models or adding additional complication.

Do larger/more modern models benefit more from prompt engineering, or are the techniques becoming obsolete? Gains from prompting on GSM8K questions have fallen as default performance on the task has improved dramatically. The creative writing task, however, is not asolved problem, and yet the gains from prompting still appear to be larger for older models - though task compliance also seems to suffer more under almost all techniques. Earlier evidence demonstrated that gains from few-shot learning increased with model scale - but this paper does not seem to support proof that the trend has held. [30] Few-shot benefits have, for the most part, decreased or flatlined with model improvement.

One might expect more recent prompt engineering techniques (as measured by paper release/publication date) to be more powerful and useful, but the evidence shows this is not the case. Chain-of-thought prompting, popularized throughout mid-late 2022 (and including least-to-most prompting) seems to have been the most significant innnovation not only for math-based reasoning tasks as might be expected, but also for the language-based creative writing challenge. The culmination of chain-of-thought improvements in zero-shotting is able to attain comparable and sometimes superior results relative the provision of manual/few-shot chain-of-thought examples. Few-shotting does still have its uses, however, particularly for creative writing tasks and when working with older models, where the construction of a coherent passage (a potentially sub- jective target) can be demonstrated. Complex iterative techniques such as Self-Refinement and traversal of the tree of thoughts, both introduced in 2023, largely do not lead to improvement (or attain less improvement than other methods), at least for these tasks. They may be appropriate only for some more specialized applications and can only be successfully applied for the

most recent models. [13] I can see that a method's performance seems fairly repeatable across tasks, but what about the variability of performance within a task? Very high ac-

[13] Inspection of results found significant non-compliance with prompt instructions for these methods, especially with older models.

curacy/compliance scores indicate consistent performance and are present for only a few models and methods—perhaps only GPT-4 with one of the chain of thought methods can be relied upon, and only for the GSM8K task. Though inter-sentence results are consistent, variability in inter-paragraph cosine sim- ilarity is moderately large, and it is not entirely clear whether any method is sufficiently reliable.

Length

Table II contains results on the length and cost of conver- sations. I begin with an analysis of the length of the entire interaction in tokens (using OpenAI's tiktoken tokenizer for the appropriate model). Prompt engineering generally requires anywhere from 1-10 times as many tokens as direct prompting, which ends up having downstream effects for the amount of human effort and direct financial costs of tokens. Zero-shot methods show moderate increases in length - factors of only 1-2 times that of direct prompting. Self-Refine and Tree-of- Thought methods exhibit massive amounts of variability across questions and models that can lead to very long conversations - likely a result of differences in LLM decisions and backtracking. Few-shot methods generally have long lengths as a result of the space occupied by examples - often, many times, that of the actual question and results. This is clear from the results on input length, which are the vast majority of overall length for these methods - unlike the results for others.

For the GSM8K task, I can compare the length of con- versations for each method with the length of the provided question + answer in Figure 2, providing another perspective as to the extent to which prompt engineering can "stretch out" interactions. For text-davinci-003, direct prompting generally produces LLM conversations and responses somewhat shorter than provided answers, but zero-shot chain-of-thought prompt- ing brings lengths up into the expected range. Other methods typically add large multiples. For GPT-4, direct responses are already near the expected length and all forms of prompting add more tokens.

In Table II, the cost of conversations was computed using rates as of November 11, 2023 - 2 cents per 1000 tokens for text-davinci-003, and 3 cents per 1000 tokens (input), 6 cents per 1000 tokens (output) for GPT-4. Prompting is the difference between spending a fraction of a penny on every conservation to several cents or more. Zero-shot methods do not increase costs much at all, and GPT-4's relative discount on input tokens makes manual/few-shot methods closer in expense to zero-shot ones, though there are still some differences. Similar to the results on length, the iterative Self-Refine and Tree-of-Thought methods have variable conversation costs.

Figure 3 considers the average change in accuracy or quality divided by the change in tokens between the prompt engineering technique and the direct prompting baseline. Is any stretching of output adding value/improving accuracy/quality?

$$\frac{AQ_{PE} - AQ_B}{Tokens_{PE} - Tokens_B}$$

Table 2: Conversation- and input-lengths and token-costs for prompting strate- gies on GSM8K and creative-writing tasks. Asterisks (*) mark the best result per model per metric.

Task	Metric	Model	Manual Few-Shot	Manual CoT	Least-to-Most	Zero-Shot CoT	APE Zero-Shot CoT	Self-Refine	Tree-of-Thought	Direct Prompting
GSM8K	Conversation Length	Text-Davinci-003	533.94* (21.162)	722.7* (35.469)	272.27* (49.824)	166.02* (47.204)	181.76* (54.995)	125.99* (36.175)	248.8* (116.245)	87.06 (42.023)
		GPT-4	579.69* (21.036)	850.04* (50.96)	332.36* (58.196)	223.88* (59.148)	243.93* (58.879)	344.86* (116.243)	764.72* (312.361)	146.69 (79.543)
	Input Length	Text-Davinci-003	531.48* (21.043)	655.48* (21.043)	158.48* (21.043)	67.48* (21.043)	80.48* (21.043)	99.48* (21.043)	137.68* (77.523)	59.48 (21.043)
		GPT-4	567.5* (21.012)	741.5* (21.012)	181.5* (21.012)	80.5* (21.012)	93.5* (21.012)	167.7* (38.215)	407.42* (154.353)	72.5 (21.012)
	Conversation Cost	Text-Davinci-003	0.011* (0.0)	0.014* (0.001)	0.005* (0.001)	0.003* (0.001)	0.004* (0.001)	0.003* (0.001)	0.005* (0.002)	0.002 (0.001)
		GPT-4	0.018* (0.001)	0.029* (0.003)	0.014* (0.003)	0.011* (0.003)	0.012* (0.003)	0.016* (0.006)	0.034* (0.014)	0.007 (0.004)
Creative Writing	Conversation Length	Text-Davinci-003	695.41* (31.517)	960.51* (36.447)	1025.02* (34.559)	256.53* (46.982)	274.04* (58.593)	382.19* (141.596)	916.81* (157.042)	200.41 (30.565)
		GPT-4	751.04* (35.272)	968.75* (55.611)	1091.41* (46.794)	459.98* (49.628)	470.72* (53.057)	520.06* (210.22)	1325.17* (158.559)	337.33 (43.738)

	Input Length	Text-Davinci-003	504.49* (7.697)	683.49* (7.697)	736.49* (7.697)	63.49* (7.697)	73.49* (7.697)	140.48* (38.088)	209.49* (7.697)	52.49 (7.697)
		GPT-4	522.21* (7.492)	701.89* (7.453)	754.89* (7.453)	75.21* (7.492)	84.89* (7.453)	160.37* (43.864)	264.89* (7.453)	65.89 (7.453)
	Conversation Cost	Text-Davinci-003	0.014* (0.001)	0.019* (0.001)	0.021* (0.001)	0.005* (0.001)	0.005* (0.001)	0.008* (0.003)	0.018* (0.003)	0.004 (0.001)
		GPT-4	0.029* (0.002)	0.037* (0.003)	0.043* (0.003)	0.025* (0.003)	0.026* (0.003)	0.026* (0.011)	0.072* (0.009)	0.018 (0.003)

Stars indicate a significant difference from the direct prompting baseline from paired t-tests at the 95% level.

Fig. 2: Average Length vs. Provided Length - GSM8K

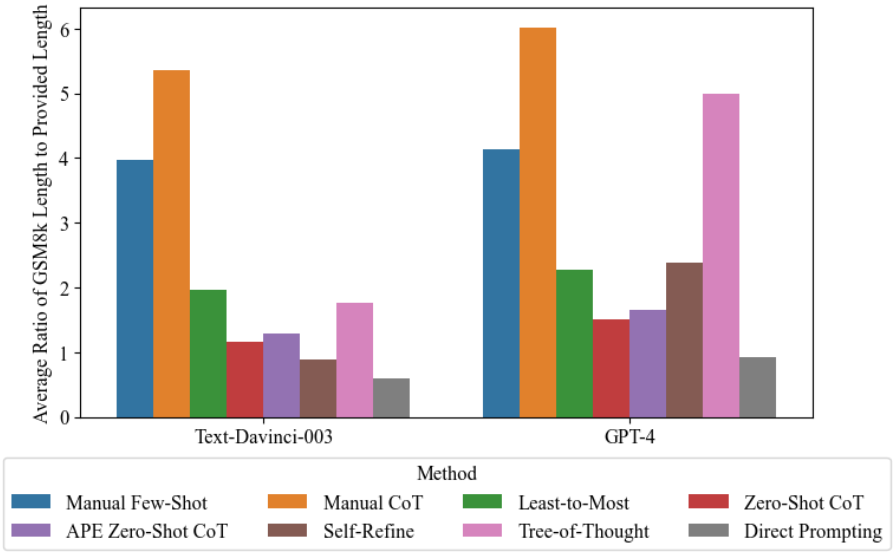
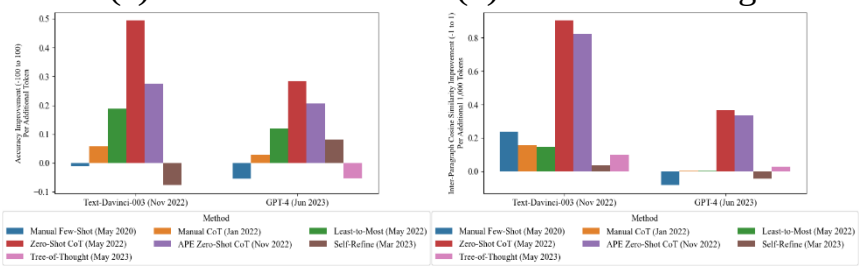


Fig. 3: Gains Per Token v. Direct Prompting

(a) GSM8K (b) Creative Writing



For GSM8K, the results are in percentage points per additional token. Zero- shot chain-of-thought prompting (particularly with the original think step-by- step prompt) is cheap and extremely effective, and it outperforms techniques introduced earlier and later. The extra details and length of Least-to-Most and Manual Chain-of-Thought prompting do not provide the same benefit. Improve- ments in creative writing cosine similarity are harder to come by - note that changes are per 1,000 additional tokens, longer than most conversations ever last. However, I can still see that Zero-Shot Chain-of-Thought prompting is the most effective, with some benefit for the provision of examples using few-shot methods with the older text-davinci-003 model.

Figure 4 repeats the same calculation, but per additional cent of financial cost and with different y-axis scales.

$$\frac{AQ_{PE} - AQ_B}{Cost_{PE} - Cost_B}$$

The unique price structures for GPT-4 make this plot different, and the new y-axis scale provides additional insight

- though one should be careful with extrapolation given the small cost (fraction of to a few cents) of most conversa- tions (these plots should be interpreted in the context of earlier tables for any given practical decision). The intricacies of pricing structure reduce the relative gains on GPT-4 and level out comparisons between methods. On older mod- els, prompting and zero-shot-prompting are very likely cost-effective, but newer models have changed this calculation.

Complexity

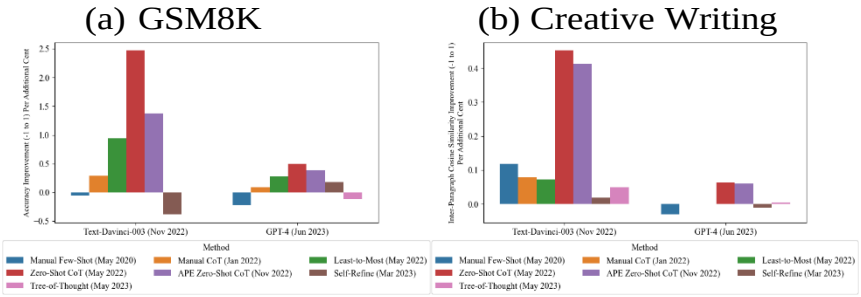
Other metrics beyond just length may be of interest - for example, in under- standing how methods work, or when considering the important task of human review/checking of LLM responses. In Table **??**, I report information on the com- plexity of responses. For both tasks, I examine the number of reasoning steps - linebreaks, sentences (NLTK sentence tokenizer), and strings ”step i” [37] and ”1. ”, ”2. ”, ”3. ”, etc. in the response. [14] Due to the intricacies of results for various methods, none of these measures provides a full picture of complexity on its own, but their combination is more insightful. Here I report meaningful results based on knowledge of the underlying data.

For the GSM8K task, which often lacks standard formatting, drawing con- clusions from the number of linebreaks is difficult, though I can clearly see that prompting tends to add lines. By combining this information with the number of sentences (which may have its problems regarding tokenization accuracy), I get a clearer picture. All prompting methods, except for manual few-shot prompting (which encourages the model just to state a number directly), add line breaks or sentences. The long structures of methods such as Least-to-Most and Tree- of-Thought prompting add the most. For the creative writing task, data on the number of line breaks shows that prompting adds a significant amount of complexity through additional steps, with multiple chain-of-thought and Tree-of- Thought methods adding the most. Considering the number of sentences, there is some doubt about this conclusion. Still, for Creative Writing, sentences likely mostly correspond to passage rather than planning length - an introduction of additional noise that makes them less useful as a metric. Despite the usage of the appearance of language such as ”step i” as a

metric in prior literature, it appears sparsely in my data, even for zero-shot methods without formatting. My new metric of the number of ”1. ”, ”2. ”, etc. is more useful. The written planning of explicit steps elucidated by Least-to-Most and other chain-of-thought meth- ods on GSM8K is clear, and there is also some planning for many methods in creative writing.

For the creative writing task, I also examine the sentence length (using NLTK word and sentence tokenizers) in the response and the Flesch Reading Ease (implemented via the textstat Python package) of the reaction [31, 32]. Most prompting methods produce slightly shorter sentences, except for the Tree-of- Thoughts, which makes longer ones - potentially through inclinations of the model to choose more complex paths and drafts of passages. Higher FRE scores indicate that the material is easier to read. When few-shot examples are provided for the manual prompting methods and Least-to-Most prompting, scores and readability increase somewhat relative to direct prompting (a modest effect, far [14] Sentences seem to be a better measure of complexity than just periods as were used in prior work (decimals, abbreviations, etc. present challenges, though the problems can be mitigated somewhat with regex). Semicolons were also considered, but in experiments, these did not appear unless models were specifically prompted towards including them. ”step i” comes from [37], but ”1. ”, ”2. ” etc. are novel metrics, to the best of my knowledge.

Fig. 4: Gains Per Cent v. Direct Prompting



textstat Python package) of the reaction [24], [25]. Most prompting methods produce slightly shorter sentences, except for the Tree-of-Thoughts, which makes longer ones - potentially through inclinations of the model to choose more complex paths and drafts of passages. Higher FRE scores indicate that the material is easier to read. When few-shot examples are provided for the manual prompting methods and Least-to-Most prompting, scores and readability increase somewhat relative to direct prompting (a modest effect, far less than a grade level). Still, for the other methods, they remain similar or fall. Controlling readability may be difficult, but in line with the literature, it is clear that few-shot examples can provide a starting point. [12]

Table IV compares a selected set of the metrics above in responses to provided answers and prompts. GSM8K responses are compared to the provided answer. Manual few- shot prompting leads to short and simple responses by design. Other methods generally introduce more reasoning steps and sentences than the provided answers - Least-to-Most prompting, in particular, shows a dramatic increase. The iterative, choice- based setup of Tree-of-Thought prompting is again notable in the addition of steps. For the creative writing task, sentence length is generally somewhat longer in the responses relative to the prompts for few-shot methods (data on the other techniques, which do not contain examples, is not as interpretable). For text-davinci-003, the few-shot responses are somewhat more

readable or roughly in line with the readability of the prompts, but for GPT-4 they are less so - a concerning trend for the prospect of controlling readability in future.

The final components of my analysis of complexity are human-provided ratings of the ease of review and difficulty of implementation for each method. For ease of review, each technique was rated in Appendix Section **??**. These results concur with the finding that Least-to-Most and Tree-of-Thought prompting add complexity, while few-shot methods often decrease it. Though chain-of-thought reasoning may add steps, these evaluations note the fact that they are usually not too difficult to follow. On the other hand, steps from least to most and Tree-of-Thought prompting are often difficult to piece back together. An additional note is that the provision of examples in few-shot prompting can, aside from readability, help with the formatting of responses. Appendix Section **??** offers some observations on the difficulty of implementing each method. Zero-shot methods are the easiest to implement. Few-shot methods require a few examples - though providing chains of reasoning can be tricky. Iterative Self-Refine and Tree-of- Thought methods are complicated and can require specialized skills and time investment.

Length or Complexity?

As is the case in the study of the chain-of-thought setting of [37], are any gains in performance coming from reasoning steps as opposed to length in tokens? A generalized answer to this question across methods is central to our understand- ing of if and how prompt engineering works. Relationships may be complex. For language understanding tasks, prompt length has been found to improve model performance, with optimality achieved somewhere in the range of 20-100 tokens. Evidence shows that in longer conversations, models may get distracted, go off on tangents, or get stuck in a loop repeating themselves (to the extent some platforms have imposed length limitations). Accounting for non-linearity and the relation of length with reasoning steps to understand these phenomena may be helpful.

Table V implements regressions (with quadratic terms to model non-linearity) controlling for length and complexity to begin to address this question. To limit collinearity with my preferred length variable of thousands of tokens, I selected the number of steps/ideas as my desired complexity metric. I summed the "step i" and "1.", "2. ", etc. measures into this single measure to improve data quality.

For GSM8K, increases in linear conversation length do tend to improve performance, but the far larger squared term indicates that further increases are detrimental beyond a certain point. The coefficients are fairly small, being per thousand tokens, and less significant than those for the number of reasoning steps/ideas. Reasoning steps are more clearly linearly beneficial for performance, and their quadratic term is small.

TABL III: Mean and Standard Deviation of Complexity Metrics

Task	Metric	Model	Manual Few-Shot	Manual CoT	Least-to-Most	Zero-Shot CoT	APE Zero-Shot CoT	Self-Refine	Tree-of-Thought	Direct Prompting

GSM8K	Number of Linebreaks	Text-Davinci-003	0.0* (0.0)	0.0* (0.0)	4.16* (1.85)	3.25* (1.74)	4.44* (2.37)	1.16* (0.66)	1.39* (0.93)	0.16 (0.72)
		GPT-4	0.0* (0.0)	1.17 (1.69)	5.64* (1.99)	3.83* (3.28)	4.64* (3.16)	5.37* (3.55)	12.26* (5.37)	1.34 (1.73)
	Number of Sentences	Text-Davinci-003	1.0* (0.0)	4.96* (1.22)	12.31* (3.06)	4.66* (2.31)	4.38* (3.01)	2.25* (1.05)	8.41* (3.38)	1.49 (0.97)
		GPT-4	1.0* (0.0)	3.5* (1.54)	8.59* (2.57)	2.84* (1.61)	3.15* (1.86)	5.12* (2.31)	8.03* (4.27)	1.51 (0.58)
	Number of Step 1, Step 2, etc.	Text-Davinci-003	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.32* (1.02)	0.56* (1.25)	0.0 (0.0)	2.0* (0.0)	0.0 (0.0)
		GPT-4	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.39* (1.14)	0.0 (0.0)	0.22 (1.32)	0.0 (0.0)
	Number of 1., 2., etc.	Text-Davinci-003	0.04* (0.2)	2.54* (2.63)	6.96* (3.06)	1.84* (2.62)	2.33* (3.24)	0.5 (0.99)	1.0 (0.0)	0.72 (1.77)
		GPT-4	0.04* (0.2)	2.8* (3.7)	6.95* (3.36)	1.99 (3.63)	1.95* (3.21)	2.36* (4.39)	3.5* (4.23)	1.28 (2.98)
Creative Writing	Number of Linebreaks	Text-Davinci-003	1.07* (0.26)	6.01* (0.5)	7.03* (0.3)	4.67* (2.47)	4.37* (2.97)	3.1* (1.54)	11.29* (2.75)	0.98 (0.25)
		GPT-4	2.01* (0.17)	3.69* (2.5)	7.5* (1.55)	10.93* (2.38)	10.77* (2.97)	4.37* (2.36)	18.54* (5.19)	2.08 (0.27)
	Number of Sentences	Text-Davinci-003	10.08* (1.86)	15.8* (1.72)	17.82* (1.88)	10.03* (2.38)	10.12* (2.48)	13.54* (5.37)	31.37* (6.9)	7.6 (1.33)
		GPT-4	10.25* (1.89)	11.93 (2.7)	14.76* (2.45)	15.92* (3.26)	15.43* (3.9)	15.97* (7.73)	39.42* (6.18)	11.32 (2.27)
	Number of Step 1, Step 2, etc.	Text-Davinci-003	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	1.71* (1.71)	1.62* (1.79)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
		GPT-4	0.0	0.0	0.0	0.66* (1.72)	0.9* (1.99)	0.0	0.0	0.0

			(0.0)	(0.0)	(0.0)			(0.0)	(0.0)	(0.0)
	Number of 1., 2., etc.	Text-Davinci-003	0.01 (0.1)	2.9* (0.46)	2.99* (0.1)	0.85* (1.18)	0.51* (0.89)	0.63* (0.79)	2.84* (0.92)	0.0 (0.0)
		GPT-4	0.0 (0.0)	0.94* (1.35)	2.49* (0.7)	3.58* (2.36)	2.81* (2.68)	0.8* (1.02)	3.93* (1.48)	0.0 (0.0)
	Sentence Length	Text-Davinci-003	16.24 (2.31)	14.4* (1.54)	13.14* (1.36)	15.38* (3.24)	16.08 (3.6)	14.63* (2.08)	18.94* (2.97)	16.41 (2.46)
		GPT-4	17.85* (2.45)	17.8* (2.71)	17.61* (2.42)	18.64 (3.69)	19.76 (4.85)	17.38* (2.0)	20.78* (2.67)	19.11 (2.6)
	Flesch Reading Ease Score	Text-Davinci-003	76.66* (7.58)	74.1 (6.22)	75.95 (5.7)	72.39* (8.54)	71.93* (8.89)	73.69 (7.86)	66.68* (8.17)	74.73 (8.24)
		GPT-4	67.84* (7.43)	67.76* (7.14)	67.37* (5.97)	59.95* (7.08)	60.74* (6.69)	62.37 (7.16)	57.57* (6.62)	63.78 (7.16)

Stars indicate a significant difference from the direct prompting baseline from paired t-tests at the 95% level.

TABLE IV: Differences of Complexity Metrics

Task	Metric	Model	Manual Few-Shot	Manual CoT	Least-to-Most	Zero-Shot CoT	APE Zero-Shot CoT	Self-Refine	Tree-of-Thought	Direct Prompting
GSM8K	Difference in Number of Sentences (Responses - Provided Answer)	Text-Davinci-003	-1.68	2.28	9.63	1.98	1.7	-0.43	5.73	-1.19

		GPT-4	-1.68	0.82	5.91	0.16	0.47	2.44	5.35	-1.17
	Difference in Number of Step 1, Step 2, etc. (Responses - Provided Answer)	Text-Davinci-003	0.0	0.0	0.0	0.32	0.56	0.0	2.0	0.0
		GPT-4	0.0	0.0	0.0	0.0	0.39	0.0	0.22	0.0
	Difference in Number of 1., 2., etc. (Responses - Provided Answer)	Text-Davinci-003	-1.24	1.26	5.68	0.56	1.05	-0.78	-0.28	-0.56
		GPT-4	-1.24	1.52	5.67	0.71	0.67	1.08	2.22	0.0
Creative Writing	Difference in Sentence Length (Responses - Prompts)	Text-Davinci-003	4.54	2.82	1.64					
		GPT-4	3.37	2.62	2.14					
	Difference in Flesch Reading Ease Score (Responses - Prompts)	Text-Davinci-003	1.66	-0.14	2.07					
		GPT-4	-7.29	-5.56	-6.87					

TABLE V: Regression Results

Model	Conversation Length (Thousand	Conversation Length (Thousand	Number of Steps/Ideas	Number of Steps/Ideas Squared	Flesch Reading Ease	Flesch Reading Ease

	s of Tokens)	s of Tokens) Squared				Squared
GSM8K Correct, Logit	0.277 (0.143)	-0.498* (0.127)	0.029* (0.007)	-0.002* (0.0)		
GSM8K Correct, Linear	0.282* (0.114)	-0.418* (0.091)	0.043* (0.006)	-0.002* (0.0)		
Creative Writing Cosine Similarity	0.212* (0.045)	-0.119* (0.03)	0.01* (0.004)	-0.0 (0.001)	-0.011* (0.004)	0.0* (0.0)
Creative Writing Compliance, Logit	-0.557* (0.161)	0.23* (0.11)	-0.025 (0.018)	0.005 (0.003)	0.041* (0.017)	-0.0* (0.0)
Creative Writing Compliance, Linear	-0.531* (0.158)	0.226* (0.107)	-0.033* (0.016)	0.006* (0.003)	0.045* (0.016)	-0.0* (0.0)

For logit regressions, the average of marginal effects are reported, and a binary indicator for the model (text-davinci-003 or GPT-4) is included. Standard errors in parentheses. For linear models, fixed effects for the model and task question are included, and standard errors are clustered by task question by method.
Stars indicate significance at the 95% level.

Does increased length or complexity limit creativity and randomness? Table V also includes regression results on the creative writing task, which may be helpful - coherence requires creativity. Length and complexity (more steps/ideas and lower FRE scores) actually increase coherence, and the quadratic terms are weak. This comes at the cost of decreased task compliance, which follows the opposite relationship.

CONCLUSION

This broad standardized and quantitative evaluation of the trade-offs behind prompt engineering has revealed a fair amount of worthwhile benefit for a selected set of techniques. Chain- of-thought prompting outperforms other techniques and direct prompting on both math and language-based tasks, and it can be implemented quite cheaply and easily in a zero-shot manner. Why does chain-of-thought prompting do so well? One plausible explanation is that LLMs have many examples loosely following the method in their training data and paired with correct answers. It is easy to find examples of reasoning problems worked out step-by-step in humanity's collective corpus of text, but ostensibly

more difficult to think of examples where a tree of possible decisions is considered or where there is text laying out a conversation of responses, feedback, and
refinement.

In a similar vein, LLMs may struggle with the creative writing task - even with prompt engineering - and with compliance with task instructions (to an alarming degree) as it is a unique task that is difficult to find examples for. Though the skills behind it and the task of maximizing passage coherence are generalizable and well-studied, the highly random nature of sentences used and the need to follow instructions exactly make for a challenge. Gains on this task from prompting are statistically significant but small and variable and limited to coherence rather than compliance.

Aside from being a somewhat task-specific question, the overall efficacy of prompting has indeed shifted over time, with performance improvements decreasing along with en- hancements in base models. Prompting is also a length and terms of accuracy and quality per additional token and cent spent on prompting have clearly fallen.

In addition to increasing the length of responses, prompting introduces additional complexity - reasoning steps and structure- and worsens readability (though sentence length may fall somewhat). Steps and complexity are not always a serious problem for human review, especially if done in a structured, ordered manner (Chain-of-Thought methods) - though this issue and complexity of implementation are real issues for the Tree-of-Thought (and, to a lesser extent, Self-Refine and, Least- to-Most) method. Few-shot prompting is indeed a promising way to slightly reduce readability problems, in addition to improving formatting and the alignment of responses to human preferences. Though gains relative to direct prompting are somewhat larger, responsiveness to the readability of few-shot example passages has fallen for newer models.

Finally, this paper tested generalized models concerning the relationship between length and complexity and accu- racy/quality - offering further insight into the drivers of performance. On GSM8K, reasoning steps are indeed generally more important than length - which is also affected by a strong, negative quadratic term. For creative writing, coherence, length, and complexity improve cosine similarity but decrease task compliance. A chain of concise, organized, and well-chosen steps (or in a few cases, examples) can still bring out moderate gains from prompting, though real challenges and drawbacks remain.

REFERENCES

1. K. Martineau, “What is prompt tuning?” Feb. 2021. [Online]. Available: https://research.ibm.com/blog/what-is-ai-prompt-tuning
2. Ethan Mollick [@emollick], “I have a strong suspicion that “prompt engineering” is not going to be a big deal in the long-term & prompt engineer is not the job of the future AI gets easier. You can already see in Midjourney how basic prompts went from complex in v3 to easy in v4. Same with ChatGPT to Bing. https://t.co/BTtSN4oVF4,” Feb. 2023. [Online]. Available: https://twitter.com/emollick/status/1627804798224580608
3. C. Shackell, “Prompt engineering: is being an AI ’whisperer’ the job of the future or a short-lived fad?” Aug. 2023. [Online]. Available: http://theconversation.com/
4. cost multiplier, especially for few-shot and iterative/complex

5. prompt-engineering-is-being-an-ai-whisperer-the-job-of-the-future-or-a-short-lived-fad-
6. methods. Putting these facts together and further accounting for the differing cost structures for newer models, the gains in
7. O. A. Acar, "AI Prompt Engineering Isn't the Future," *Harvard Business Review*, Jun. 2023, section: Technology and analytics. [Online]. Available: https://hbr.org/2023/06/ai-prompt-engineering-isnt-the-future
8. T. Wu, M. Terry, and C. J. Cai, "AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts," in *CHI Conference on Human Factors in Computing Systems*. New Orleans LA USA: ACM, Apr. 2022, pp. 1–22. [Online]. Available: https://dl.acm.org/doi/10.1145/3491102.3517582
9. S. Diao, P. Wang, Y. Lin, and T. Zhang, "Active Prompting with Chain- of-Thought for Large Language Models," May 2023, arXiv:2302.12246 [cs]. [Online]. Available: http://arxiv.org/abs/2302.12246

A. Gao, "Prompt Engineering for Large Language Models," Rochester, NY, Jul. 2023. [Online]. Available: https://papers.ssrn.com/abstract= 4504303

10. Y. Fu, H. Peng, A. Sabharwal, P. Clark, and T. Khot, "Complexity-Based Prompting for Multi-Step Reasoning," Jan. 2023, arXiv:2210.00720 [cs]. [Online]. Available: http://arxiv.org/abs/2210.00720
11. K. Shum, S. Diao, and T. Zhang, "Automatic Prompt Augmentation and Selection with Chain-of-Thought from Labeled Data," Feb. 2023, arXiv:2302.12822 [cs]. [Online]. Available: http://arxiv.org/abs/2302. 12822
12. G. Adams, A. Fabbri, F. Ladhak, E. Lehman, and N. Elhadad, "From Sparse to Dense: GPT-4 Summarization with Chain of Density Prompting," Sep. 2023, arXiv:2309.04269 [cs]. [Online]. Available: http://arxiv.org/abs/2309.04269
13. D. Pu and V. Demberg, "ChatGPT vs Human-authored Text: Insights into Controllable Text Summarization and Sentence Style Transfer," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*. Toronto, Canada: Association for Computational Linguistics, Jul. 2023,
14. pp. 1–18. [Online]. Available: https://aclanthology.org/2023.acl-srw.1
15. J. M. Imperial and H. T. Madabushi, "Flesch or Fumble? Evaluating Readability Standard Alignment of Instruction-Tuned Language Models," Sep. 2023, arXiv:2309.05454 [cs]. [Online]. Available: http://arxiv.org/abs/2309.05454
16. K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser,
17. M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and
18. J. Schulman, "Training Verifiers to Solve Math Word Problems," Nov. 2021, arXiv:2110.14168 [cs]. [Online]. Available: http://arxiv.org/abs/ 2110.14168
19. S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and
20. K. Narasimhan, "Tree of Thoughts: Deliberate Problem Solving with Large Language Models," May 2023, arXiv:2305.10601 [cs]. [Online].
21. Available: http://arxiv.org/abs/2305.10601
22. OpenAI, "GPT-4 Technical Report," Mar. 2023, arXiv:2303.08774 [cs]. [Online]. Available: http://arxiv.org/abs/2303.08774
23. T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal,

24. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal,
25. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin,
26. S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford,
27. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," Jul. 2020, arXiv:2005.14165 [cs]. [Online]. Available: http://arxiv.org/abs/2005.14165
28. J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi,
29. Q. V. Le, and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models."
30. D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang,
31. D. Schuurmans, C. Cui, O. Bousquet, Q. Le, and E. Chi, "Least- to-Most Prompting Enables Complex Reasoning in Large Language Models," Apr. 2023, arXiv:2205.10625 [cs]. [Online]. Available: http://arxiv.org/abs/2205.10625
32. T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large Language Models are Zero-Shot Reasoners," Jan. 2023, arXiv:2205.11916 [cs]. [Online]. Available: http://arxiv.org/abs/2205.11916
33. Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba, "Large Language Models are Human-Level Prompt Engineers," Sep. 2022. [Online]. Available: https://openreview.net/forum?id=92gvk82DE-
A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe,
34. U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, S. Gupta, B. P. Majumder,
35. K. Hermann, S. Welleck, A. Yazdanbakhsh, and P. Clark, "Self-Refine: Iterative Refinement with Self-Feedback," May 2023, arXiv:2303.17651 [cs]. [Online]. Available: http://arxiv.org/abs/2303.17651
36. T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, no. 2-3, pp. 259–284, Jan. 1998. [Online]. Available: http://www.tandfonline.com/doi/abs/10. 1080/01638539809545028
37. "sentence-transformers/all-distilroberta-v1 · Hugging Face." [Online]. Available: https://huggingface.co/sentence-transformers/ all-distilroberta-v1
38. R. Flesch, "How to Write Plain English," Jul. 2016. [Online]. Available: https://web.archive.org/web/20160712094308/http:
39. //www.mang.canterbury.ac.nz/writing_guide/writing/flesch.shtml
40. S. B. Aggarwal, Chaitanya, "textstat: Calculate statistical features from text." [Online]. Available: https://github.com/shivam5992/textstat
41. B. Lester, R. Al-Rfou, and N. Constant, "The Power of Scale for Parameter-Efficient Prompt Tuning," Sep. 2021, arXiv:2104.08691 [cs]. [Online]. Available: http://arxiv.org/abs/2104.08691
42. F. Shi, X. Chen, K. Misra, N. Scales, D. Dohan, E. H. Chi,
43. N. Schärli, and D. Zhou, "Large Language Models Can Be Easily Distracted by Irrelevant Context," in *Proceedings of the 40th International Conference on Machine Learning*. PMLR, Jul. 2023, pp. 31 210–31 227, iSSN: 2640-3498. [Online]. Available:
44. https://proceedings.mlr.press/v202/shi23a.html
45. J. Mann, "Microsoft limits Bing chat exchanges and conversation lengths after 'creepy' interactions with some users." [Online]. Available: https://www.businessinsider.com/ microsoft-limits-bing-chat-exchanges-and-conversation-lengths-2023-2