

Two Failures of Self-Consistency in the Multi-Step Reasoning of LLMs

Angelica Chen¹, Jason Phang¹, Alicia Parrish^{1,2}, Vishakh Padmakumar¹,
Chen Zhao¹, Samuel R. Bowman^{1,3}, Kyunghyun Cho¹

¹New York University

²Google

³Anthropic

{ac5968, zp489, avp295, vp1271, cz1285, sb6065, kc119}@nyu.edu

Abstract

Large language models (LLMs) have achieved widespread success on a variety of in-context few-shot tasks, but this success is typically evaluated via correctness rather than consistency. We argue that self-consistency is an important criteria for valid multi-step reasoning in tasks where the solution is composed of the answers to multiple sub-steps. We propose two types of self-consistency that are particularly important for multi-step reasoning – hypothetical consistency (a model’s ability to predict what its output would be in a hypothetical other context) and compositional consistency (consistency of a model’s final outputs when intermediate sub-steps are replaced with the model’s outputs for those steps). We demonstrate that multiple variants of the GPT-3/4 models exhibit poor consistency rates across both types of consistency on a variety of tasks.

1 Introduction

An important property of logically valid machine learning systems is *self-consistency* – i.e., the requirement that no two statements given by the system are contradictory. Pre-trained large language models (LLMs), despite demonstrating impressive few-shot accuracy on a variety of multi-step reasoning tasks, often give inconsistent responses to questions (Mitchell et al., 2022; Kassner et al., 2021) and factual knowledge-seeking prompts (Elazar et al., 2021). Without self-consistency, it is difficult to consider LLMs reliable or trustworthy systems. Elazar et al. (2021) defines self-consistency as the invariance of an LLM’s responses across different types of semantics-preserving *prompt transformations*. In this work, we seek to introduce and explore LLM self-consistency over two new types of transformations (shown in Figure 1) that we argue are important for valid multi-step reasoning.

Hypothetical Transformations A *hypothetical transformation* is an indirect phrasing of a prompt

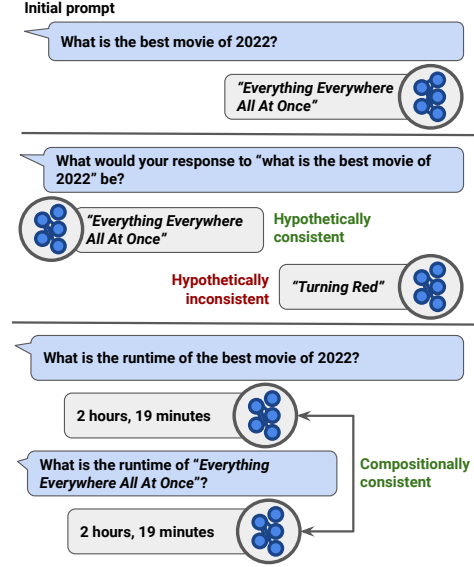


Figure 1: An overview of the two types of self-consistency failures we identify in LLMs.

that queries the model for what its response would hypothetically be in some other context, such as “what would your response to <prompt> be?” or “what would the next 5 words in your completion of <prompt> be?” Consistency over hypothetical transformations implies that an LLM has some stored knowledge or computational sub-graph for determining what its response would be to some prompt p without explicitly being prompted with exactly p itself. This can be useful for prompts involving multi-step reasoning, where the LLM must have knowledge of its responses to the earlier steps in order to compute its responses to downstream steps. Like in Figure 1, given the prompt “What is the runtime of the best movie of 2022” the LLM must either have stored or computed its response to “what is the best movie of 2022?” in order to answer the full prompt.

Compositional Transformations For a prompt that involves multiple interdependent steps of reasoning, a *compositional transformation* consists of

replacing some intermediate step with the model’s output to the previous step. In the previous example, if the LLM outputs the response “*Everything Everywhere All At Once*” to the prompt “What is the best movie of 2022?,” then the prompt “What is the runtime of *Everything Everywhere All At Once*?” is a compositional transformation of “What is the runtime of the best movie of 2022?” (See Figure 1.) Consistency over compositional transformations is also important for logically valid multi-step reasoning when the LLM must give a direct response – without it, the LLM may give contradictory direct responses to different multi-step prompts that are in fact querying for the same thing.

In this work, we investigate the degree to which LLMs are self-consistent on these prompt transformations across a variety of tasks. We show empirically that pre-trained language models demonstrate low consistency rates on both hypothetical transformations (Section 2) and compositional transformations (Section 3). Additionally, we provide formal definitions of hypothetical and compositional consistency in Appendix A.1.

2 Evaluating Consistency on Hypothetical Transformations

We first explore the degree to which LLM outputs are invariant to hypothetical transformations of the prompt. To test this kind of consistency, we devise a set of four hypothetical transformation prompt templates (Appendix A.2, Table 1).

To measure hypothetical consistency, we use a multiple-choice set-up. One answer choice is the continuation of the initial prompt (denoted by “<prompt>”) sourced from a text dataset, one choice is the model’s own greedily decoded completion for <prompt>, and the three remaining choices are the other models’ completions for <prompt>. As discussed before, a model that is hypothetically consistent can, in a sense, predict its own completion. As such, it should be more likely to generate the answer choice that corresponds to its own completion than to the other answer choices. These templates are designed both to query the model on what its completion would hypothetically be for a given prompt and to evaluate whether the model can distinguish its own completions from those of other models.

We conduct our hypothetical consistency experiments with original prompts sourced from two language modeling tasks – Wikipedia and DailyDi-

alog (Li et al., 2017) (Appendix A.3). We use only the prompts for which all five answer choices are distinct. We also vary the number of words m in the original completion that the model is asked to distinguish from 1 to 6, since the difficulty may vary depending on the length of the original completion. We then compute the *hypothetical consistency accuracy* by calculating the proportion of the time that the model generated the letter of the answer choice corresponding to its own completion.

2.1 Experimental Setup

In all experiments, we evaluate four model sizes of the OpenAI GPT-3 model (Brown et al., 2020) – ada-001, babbage-001, curie-001, and davinci-003 (in order of increasing capacity).¹ All experiments are run using greedily decoded completions obtained from the OpenAI API from Aug. 2022 to Jun. 2023. Initial prompts are 0-shot, whereas hypothetical consistency prompts range from 1-shot to 10-shot. We source the <prompt>s from DailyDialog (Li et al., 2017) and Wikipedia (Wikimedia Foundation). Full dataset details can be found in Appendix A.3.

2.2 All Model Sizes Perform Poorly At Distinguishing Their Own Completions

Figure 2 shows GPT-3’s hypothetical consistency rates averaged over all few-shot prompts. Notably, all model sizes smaller than davinci-003 perform at about random chance on this task, regardless of how many words of the original completion the model is tasked with predicting. davinci-003 is the only model size that consistently performs above random chance, but even then its accuracy ranges only from 26% to 31% for Wikipedia and 30% to 37% for DailyDialog.

We also inspect the frequency with which each model selects each possible answer choice, as shown in Figure 3. For both tasks, only davinci-003 demonstrates a noticeable preference for its own completion over others.

3 Evaluating Compositional Self-Consistency

3.1 Experimental Setup

We evaluate compositional self-consistency across six models (ada-001, babbage-001,

¹We select this particular set of models since it is the most recent set of text completion (rather than chat) models available for each size of GPT-3. However, we also compare against text-davinci-001 and gpt4 in Appendix A.5.

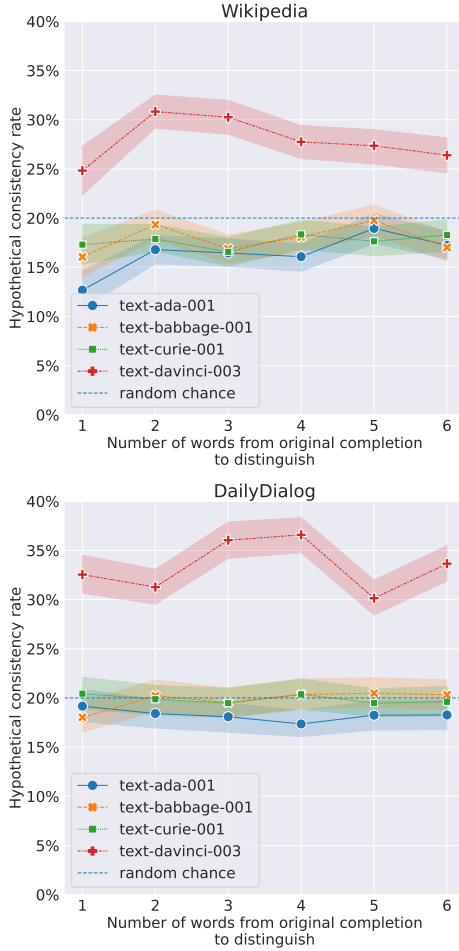


Figure 2: Hypothetical consistency rates on multiple-choice self-knowledge prompts for the Wikipedia and DailyDialog datasets, across the four GPT-3 model sizes.

curie-001, davinci-001, davinci-003, and gpt4) and two tasks. We specifically choose arithmetic and semantic parsing because they are both tasks where valid compositional reasoning are important.

Synthetic Arithmetic We generate a set of 400 randomly-nested arithmetic expressions as the initial prompts. We then collect model completions for all possible sub-expressions of each expression using k -shot prompts, with k ranging from 3 to 10. (Full details are given in Appendix A.3.) For example, for the original arithmetic expression " $(2 \times 3) + (6/2)$," we prompt each model with the following three sub-expression prompts, using parentheses to force the correct order of operations:

$$\begin{aligned} p_1 &= \text{"Q: } 2 \times 3 \text{ \n A: "} \\ p_2 &= \text{"Q: } 6/2 \text{ \n A: "} \\ p_3 &= \text{"Q: } (2 \times 3) + (6/2) \text{ \n A: "} \end{aligned}$$

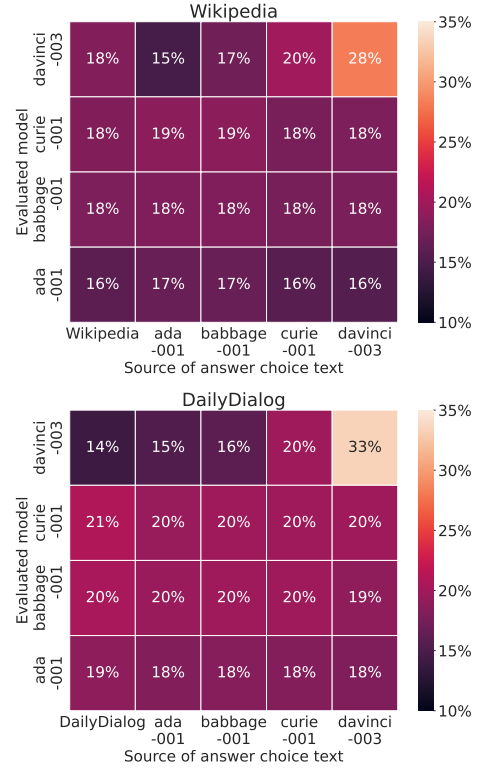


Figure 3: The proportion of the time that each model selects each possible answer choice when prompted with a hypothetical consistency prompt, averaged across different numbers of in-context examples. Model outputs that could not be parsed into an answer choice are not included.

For each non-root sub-expression (*i.e.* p_1 and p_2), we then create a new *compositional consistency prompt* by replacing that sub-expression in the original expression (*i.e.* p_3) with the model’s completion. For the previous example, if the model answered p_1 and p_2 correctly, this would result in the following two compositional consistency prompts:

$$p_{CC}^{(1)} = \text{"Q: } 6 + (6/2) \text{ \n A: "}$$

$$p_{CC}^{(2)} = \text{"Q: } (2 \times 3) + 3 \text{ \n A: "}$$

For this example, we then compute a model’s *compositional consistency rate* as the proportion of the time that the model’s output for p_i is correct, and its outputs for $p_{CC}^{(i)}$ and p_3 are the same. Further formal definitions are given in Appendix A.1.

GeoQuery GeoQuery (Zelle and Mooney, 1996) is a semantic parsing dataset consisting of 880 natural language questions about US geography, paired with FunQL (Kate et al., 2005) parses of those questions. Similar to the synthetic arithmetic task, we first collect model parses for the spans corresponding to each sub-parse of a sample of 400 Geo-



Figure 4: Compositional consistency rates versus the number of in-context examples on the arithmetic and GeoQuery tasks.

Query training examples via k -shot prompts, for k ranging from 2 to 10. For example, consider the GeoQuery example “Which state has the city with the most population?” with corresponding FunQL `state(loc_1(largest_one(population_1(city(all)))))`. Then two of the initial prompts we create include the following:

p_1 = “Create a FunQL query for the following question: ‘Which state has the city with the most population?’ A: ”

p_2 = “Create a FunQL query for the following question: ‘city with the most population’ A: ”

where each prompt is sourced from a non-leaf sub-parse of the original gold FunQL expression (and the other sub-parses are omitted here for brevity).

We then compute a model’s compositional consistency rate as the proportion of the time that the model’s parse for p_2 is correct and is a sub-parse of its parse for p_1 (regardless of whether the parse for p_1 is correct).

3.2 All Models Exhibit Poor Compositional Consistency

The compositional consistency rates for all six models are shown in Figure 4. While `davinci-003` and `gpt4` exhibit the highest compositional consistency rates, both are compositionally consistent less than 50% of the time on average. However, `davinci-003` appears to improve in compositional consistency on the GeoQuery task as the number of in-context examples increases. Lastly, compositional consistency correlates strongly with correctness (Appendix A.4) on the arithmetic task, which may be an explanation for `gpt4`’s strong compositional consistency on solely arithmetic.

4 Related Work

Our work is inspired by an extensive body of literature that has defined and evaluated consistency in a variety of ways. [Elazar et al. \(2021\)](#) defined consistency as the ability for the LLM to give consistent responses to semantically equivalent contexts, such as paraphrased contexts. [Jang et al. \(2022\)](#) supplements this definition with multiple other categories of logical consistency. Yet other work has highlighted the inconsistency of LLM predictions across paraphrases of the same input for a variety of downstream tasks, including knowledge extraction ([Elazar et al., 2021](#); [Fierro and Søgaard, 2022](#); [Newman et al., 2022](#)), truthfulness ([Raj et al., 2022](#)), summarization ([Kryscinski et al., 2020](#)), and natural language understanding ([Jang et al., 2021](#); [Zhou et al., 2022](#)).

5 Conclusion

We have proposed two types of language model self-consistency that are important for the reliability and logically valid reasoning of LLMs on multi-step tasks. Despite the GPT-3 and GPT-4 models’ generally impressive performance on a wide variety of tasks, these models still perform inconsistently on both hypothetical and compositional consistency prompts, although larger model size appears to help. This suggests an additional reason not to trust the outputs of LLMs on complex compositional tasks, especially without extensive empirical validation. Further work is required in order to improve the logical consistency of LLM reasoning, and to investigate whether further scaling improves hypothetical or compositional consistency.

References

- 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.
- OpenAI Help Center. [Do the openai api models have knowledge of current events?](#)
- Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhishek Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. [Measuring and Improving Consistency in Pretrained Language Models](#). *Transactions of the Association for Computational Linguistics*, 9:1012–1031.
- Constanza Fierro and Anders Søgaard. 2022. [Factual consistency of multilingual pretrained language models](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3046–3052, Dublin, Ireland. Association for Computational Linguistics.
- Myeongjun Jang, Deuk Sin Kwon, and Thomas Lukasiewicz. 2021. [Accurate, yet inconsistent? consistency analysis on language understanding models](#).
- Myeongjun Jang, Deuk Sin Kwon, and Thomas Lukasiewicz. 2022. [BECEL: Benchmark for consistency evaluation of language models](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3680–3696, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Nora Kassner, Øyvind Tafjord, Hinrich Schütze, and Peter Clark. 2021. [BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8849–8861, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Rohit J. Kate, Yuk Wah Wong, and Raymond J. Mooney. 2005. Learning to transform natural to formal languages. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*, AAAI’05, page 1062–1068. AAAI Press.
- Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. [Evaluating the factual consistency of abstractive text summarization](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. [DailyDialog: A manually labelled multi-turn dialogue dataset](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Eric Mitchell, Joseph J. Noh, Siyan Li, William S. Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher D. Manning. 2022. [Enhancing self-consistency and performance of pretrained language models with nli](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.
- Benjamin Newman, Prafulla Kumar Choubey, and Nazneen Rajani. 2022. [P-adapters: Robustly extracting factual information from language models with diverse prompts](#). In *International Conference on Learning Representations*.
- OpenAI. Model index for researchers. <https://platform.openai.com/docs/model-index-for-researchers>.
- OpenAI. 2023. [Gpt-4 technical report](#).
- Harsh Raj, Domenic Rosati, and Subhabrata Majumdar. 2022. [Measuring reliability of large language models through semantic consistency](#). In *NeurIPS ML Safety Workshop*.
- Wikimedia Foundation. [Wikimedia downloads](#).
- John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI’96, page 1050–1055. AAAI Press.
- Chunting Zhou, Junxian He, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. [Prompt consistency for zero-shot task generalization](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2613–2626, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Appendix

A.1 Formalization of consistency definitions

To make more precise our definitions of consistency and the semantics-preserving transformations that they entail, we attempt to formalize our definitions here.

A.1.1 Preliminaries

Let vocabulary \mathcal{V} be a finite set of tokens, \mathcal{V}^* be the set of all possible finite-length sequences formed by concatenating zero or more tokens from \mathcal{V} , and $p_\theta : \mathcal{V} \rightarrow \{0, 1\}$ be an auto-regressive language model that defines a probability distribution over tokens $v \in \mathcal{V}$. p_θ can be used to generate a sequence via greedy decoding as follows:

$$\tilde{y}_t = \arg \max_{v \in \mathcal{V}} \log p_\theta(y_t = v \mid c; \tilde{y}_{<t}) \quad (1)$$

given some context sequence $c \in \mathcal{V}^*$, until some time step T for which $\tilde{y}_T = [EOS]$, the end-of-sequence token. For ease of notation, we denote the greedy decoding output of a model p_θ as

$$g_{p_\theta}(c) = (\arg \max_{v \in \mathcal{V}} p_\theta(y = v \mid c), \dots, \arg \max_{v \in \mathcal{V}} p_\theta(y = v \mid c; \tilde{y}_{<T})). \quad (2)$$

We also define an operator \sim that indicates when two strings are *semantically equivalent*. Although the precise definition of semantic equivalence will vary across different tasks, we use it to loosely refer to pairs of strings that can be used interchangeably (give or take syntactic adjustments) without changing the meaning of the overall utterance. The \sim operator is also reflexive, symmetric, and transitive.

A.1.2 Composing prompts

Reasoning with language often also involves composing prompts – for instance, we might ask “what is the answer to $2 \times 3 + 4$?”, which can be seen as the composition of a *prompt template* “what is the answer to $_ + 4$?” with the prompt “ 2×3 ”, where the “ $_$ ” symbol in the former string is substituted with the latter string. This corresponds to a multi-step task where the model might first answer the prompt “ 2×3 ” (yielding $g_{p_\theta}(\text{“}2 \times 3\text{”})$), substitute $g_{p_\theta}(\text{“}2 \times 3\text{”})$ into the template (yielding the composed prompt “what is the answer to $g_{p_\theta}(\text{“}2 \times 3\text{”) + 4$,” where the $g_{p_\theta}(\text{“}2 \times 3\text{”})$ is replaced with the actual output string), and then answer the filled-in template.

To denote such prompt templates, we define \mathcal{P}' , the set of prompts $p \in \mathcal{V}^*$ that contain exactly one

“ $_$ ” symbol. Additionally, the function $f(p', p) : \mathcal{P}' \times \mathcal{V}^* \rightarrow \mathcal{V}^*$ denotes substitution of p for the “ $_$ ” symbol in p' .²

f also has some useful properties that we will use in our later definitions:

- We can trivially represent any prompt $p \in \mathcal{V}^*$ as the substitution of itself into the *identity prompt template* “ $_$ ” by writing $p = f(\text{“}_\text{”}, p)$.
- $p \sim q$ if and only if $f(p', p) \sim f(p', q)$ for all $p, q \in \mathcal{V}^*$.

A.1.3 Definitions

We start out by restating the general definition of self-consistency, as it has been commonly defined in past literature (Elazar et al., 2021; Jang et al., 2022).

Definition A.1 (Self-consistency). p_θ is *self-consistent* if $g_{p_\theta}(p) \sim g_{p_\theta}(q) \iff p \sim q$ for all $p, q \in \mathcal{V}^*$.

In other words, a self-consistent model gives semantically-equivalent responses to semantically equivalent prompts. These semantically equivalent pairs of prompts (p, q in Definition A.1) can take many forms, including hypothetical and compositional transformations.

Definition A.2 (Hypothetical Transformation). Let \mathcal{P}'_I denote the set of *hypothetical transformation prompt templates*, which are prompt templates $p' \in \mathcal{P}'$ such that $f(p', p) \sim f(_, p) \forall p \in \mathcal{V}^*$. Then the set of *hypothetical transformations* of prompt p can be denoted as $\mathcal{P}_I(p) := \{f(p', p) \mid p' \in \mathcal{P}'_I\}$.

Since $f(_, p) \sim p$, a model that is self-consistent must yield $g_{p_\theta}(f(p', p)) \sim g_{p_\theta}(p)$ for all $p' \in \mathcal{P}'_I$.

Although we defined hypothetical transformations with respect to all prompts $p \in \mathcal{V}^*$, our definition of compositional transformations must be more restricted, since we care only to apply compositional transformations to prompts that implicitly encode a compositional task.

Definition A.3 (Compositional transformation). For a prompt composition $f(p', p)$, the *compositional transformation* with respect to model p_θ is $f(p', g_{p_\theta}(p))$.

Given the above two types of prompt transformations, we can define narrower types of LLM self-consistency.

²In practice, substituting p into p' may require minor syntactic adjustments for linguistic acceptability, but we omit these in our notation since the semantics remain the same.

Hypothetical Transformation Prompt Templates

"I predict that the next $\langle m \rangle$ words after " $\langle \text{prompt} \rangle$ " would be $\langle \text{answer_choices} \rangle$. Answer:"
 "Given the prompt " $\langle \text{prompt} \rangle$ ", my next $\langle m \rangle$ words would be $\langle \text{answer_choices} \rangle$. Answer:"
 "Given the context " $\langle \text{prompt} \rangle$ ", my next $\langle m \rangle$ words would be $\langle \text{answer_choices} \rangle$. Answer:"
 "I predict that after the context " $\langle \text{prompt} \rangle$ " the next $\langle m \rangle$ words I would say are $\langle \text{answer_choices} \rangle$. Answer:"

Table 1: Prompt templates used to evaluate whether LLMs are consistent across hypothetical prompt transformations. $\langle \text{prompt} \rangle$ is a prompt sourced from a dataset (e.g. Wikipedia, DailyDialog), $\langle m \rangle$ is the number of words of its own completion that the model is asked to predict, and $\langle \text{answer_choices} \rangle$ are the multiple-choice answer choices that the model is given. Answer choices are shuffled and formatted like "A) ... B) ... C) ... D) ... E) ..."

Definition A.4 (Hypothetical consistency). A model p_θ is *hypothetically consistent* if $g_{p_\theta}(p) \sim g_{p_\theta}(f(p', p))$ for any prompt $p \in \mathcal{V}^*$ and hypothetical transformation prompt template $p' \in \mathcal{P}'_I$.

Claim A.5. If p_θ is self-consistent, then p_θ is also hypothetically consistent.

Proof. Consider prompt $p \in \mathcal{V}^*$ and hypothetical transformation prompt template $p' \in \mathcal{P}'_I$. Since $f(p', p) \sim f(_, p)$ (by Definition A.2) and $f(_, p) \sim p$, then $f(p', p) \sim p$ by the transitive property of \sim . Then by Definition A.1, it follows that $g_{p_\theta}(f(p', p)) \sim g_{p_\theta}(p)$. \square

Definition A.6 (Consistency over compositional transformations). Given a prompt template p' and a prompt p that can be substituted into it, a model p_θ is *compositionally consistent* when:

1. $p \sim g_{p_\theta}(p)$ and $g_{p_\theta}(f(p', p)) \sim g_{p_\theta}(f(p', g_{p_\theta}(p)))$

and is *compositionally inconsistent* when:

1. $p \not\sim g_{p_\theta}(p)$ and $g_{p_\theta}(f(p', p)) \sim g_{p_\theta}(f(p', g_{p_\theta}(p)))$
2. $p \sim g_{p_\theta}(p)$ and $g_{p_\theta}(f(p', p)) \not\sim g_{p_\theta}(f(p', g_{p_\theta}(p)))$

but is undefined in other cases.

A.2 Hypothetical Transformation Prompt Templates

The full set of hypothetical prompt templates that we use to transform the original prompts (sourced from Wikipedia and DailyDialog) into hypothetical prompts are listed in Table 1. As an example, suppose the original prompt sourced from Wikipedia is "This quilt begun in 1856 when she was seventeen includes the autographs on

top of the blocks of many known celebrities and politicians of the day. Other". Suppose that the first three words of the completions generated by ada-001, babbage-001, curie-001, and davinci-003 are "notable quilt authors," "famous quilts include," "signatures include abolitionists," and "notable figures whose," respectively. Additionally, the next three words of the Wikipedia article are "figures represented on." Then a hypothetical transformation prompt that uses the first template in Table 1 might look like:

I predict that the next 3 words after "This quilt begun in 1856 when she was seventeen includes the autographs on top of the blocks of many known celebrities and politicians of the day. Other" would be

- A) famous quilts include
- B) figures represented on
- C) notable quilt authors
- D) signatures include abolitionists
- E) notable figures whose

Answer:

If the model being evaluated is davinci-003, then the correct answer would be E. As mentioned in Section 2.1, all hypothetical prompts are few-shot, and only case-insensitive exact-match answers (*i.e.* "a/A/b/B/c/C/d/D") are accepted as correct.

A.3 Dataset Details

Wikipedia Since language models are frequently pre-trained on Wikipedia archives, evaluating on a Wikipedia dataset can confound information memorized during pre-training with the skill being evaluated. To address this issue, we collect a sam-

ple of 400 English Wikipedia (Wikimedia Foundation) articles that were created on or after June 30, 2021, since the OpenAI documentation (Center) indicates that the latest pre-training data for the ada-/babbage-/curie-/davinci-001 and davinci-003 models contains documents dating up to June 2021. Each initial prompt is a randomly selected segment of a Wikipedia article consisting of two full sentences followed by a random-length prefix of the next sentence.

DailyDialog DailyDialog (Li et al., 2017) is a manually labeled dataset of multi-turn conversations about daily life. We choose this dataset because it contains language that is more colloquial in style and less factual in content than the Wikipedia dataset. We randomly sample 400 examples from the training split and use the first conversational turn as the initial prompt.

Synthetic Arithmetic The arithmetic expressions have a maximum nesting depth of 5 with a nesting probability of 0.5, and each nested expression is enclosed in parentheses to avoid ambiguity. Operators and operands are randomly selected with uniform probability from the sets $\{+, -, /, \times\}$ and $[1, 2, \dots, 999]$, respectively.

A.4 Correctness Versus Compositional Consistency

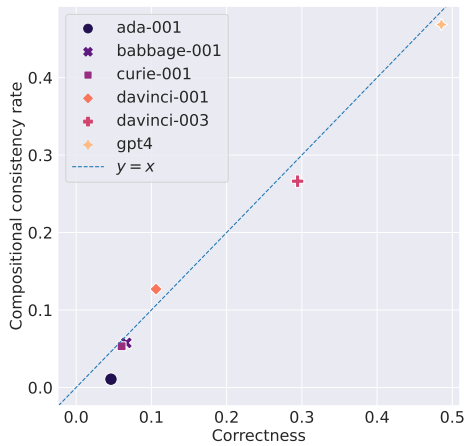


Figure 5: The correctness versus compositional consistency rate of each type of GPT-3 or GPT-4 model on the arithmetic task.

Since there exists a more precise definition of correctness for the synthetic arithmetic task, we can evaluate the relationship between correctness and compositional consistency on this task. Figure 5 shows the correctness versus compositional con-

sistency rate for all four model sizes. There exists a notable linear relationship between correctness and compositional consistency, but all models except for davinci-001 are slightly more correct than consistent.

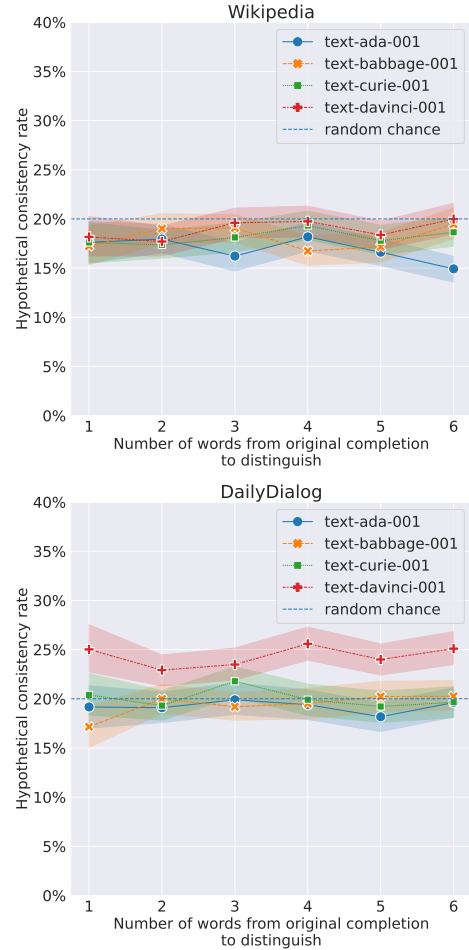


Figure 6: Hypothetical consistency rates of ada-001, babbage-001, curie-001, and davinci-001 on multiple-choice hypothetical consistency prompts for the Wikipedia and DailyDialog datasets. Each multiple-choice prompt contains answer choices generated by the aforementioned four models and an additional answer choice containing the actual continuation of the prompt in the dataset.

A.5 Comparison of Hypothetical Consistency Against davinci-001 and gpt4

We also run the same hypothetical consistency experiments on davinci-001 and gpt4. Hypothetical consistency rates for davinci-001 versus the smaller models are shown in Figure 6, where trends are similar, but davinci-001 performs at random chance on Wikipedia, like all the other -001 series models. On DailyDialog, however, davinci-001 performs noticeably better

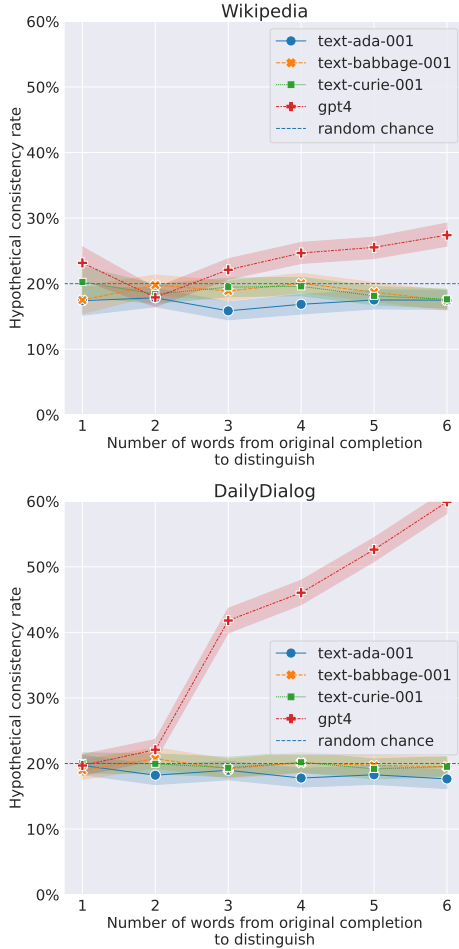


Figure 7: Hypothetical consistency rates of ada-001, babbage-001, curie-001, and gpt4 on multiple-choice hypothetical consistency prompts for the Wikipedia and DailyDialog datasets. Each multiple-choice prompt contains answer choices generated by the aforementioned four models and an additional answer choice containing the actual continuation of the prompt in the dataset.

than all the other model sizes. Similar trends occur in Figure 8, where most models are equally likely to select each answer choice on the Wikipedia dataset, and davinci-001 is more likely to select either its own or curie-001’s completion on the DailyDialog dataset.

In contrast, when gpt4 is tasked with distinguishing its own completions from those of ada-001, babbage-001, curie-001, and the dataset, gpt4 performs notably better than both davinci-001 and davinci-003 on DailyDialog, reaching 59.9% hypothetical consistency when the number of words to distinguish is 6 (Figure 7). However, its hypothetical consistency rate on Wikipedia is comparable to that of davinci-003 (Figure 2), ranging from 17.9%

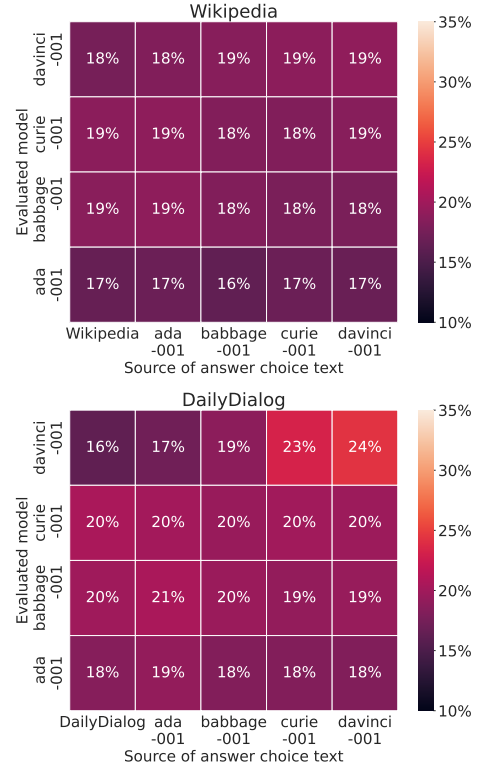


Figure 8: The proportion of the time that each model (ada-001, babbage-001, curie-001, and davinci-001) selects each possible answer choice when prompted with a hypothetical consistency prompt. Model outputs that could not be parsed into an answer choice are not included.

to 27.4%. Figure 9 also demonstrates that gpt4 is significantly more likely to select its own completion than the other models are.

It is unclear why gpt4 is more consistent on DailyDialog than previous models of similar capacity (*i.e.* davinci-001 and davinci-003). Little is known about gpt4’s architecture or training, aside from its multimodal abilities and training via reinforcement learning from human feedback (RLHF, OpenAI, 2023). Since davinci-003 was also trained with RLHF (OpenAI), it is possible that other changes in architecture or training may have also contributed to the significant improvement in hypothetical consistency.

It is also unlikely that gpt4’s improvements in hypothetical consistency on DailyDialog can be attributed to dataset memorization. Firstly, we selected only prompts from both Wikipedia and DailyDialog for which all five answer choices were distinct, so gpt4’s completion could not have been identical to that of the original DailyDialog dataset. Secondly, we computed the average percent edit distance (the edit distance divided by the

Table 2: Average percent edit distances between completions from davinci-003 versus completions from the three other models and two datasets.

Dataset	davinci-003 / ada-001	davinci-003 / babbage-001	davinci-003 / curie-001	davinci-003 / Dataset
Wikipedia	72.8%	69.7%	65.0%	70.3%
DailyDialog	75.8%	75.1%	74.2%	79.0%

Table 3: Average percent edit distances between completions from davinci-001 versus completions from the three other models and two datasets.

Dataset	davinci-001 / ada-001	davinci-001 / babbage-001	davinci-001 / curie-001	davinci-001 / Dataset
Wikipedia	73.6%	71.1%	64.4%	71.4%
DailyDialog	71.4%	70.1%	69.3%	79.4%

Table 4: Average percent edit distances between completions from gpt 4 versus completions from the three other models and two datasets.

Dataset	gpt 4 / ada-001	gpt 4 / babbage-001	gpt 4 / curie-001	gpt 4 / Dataset
Wikipedia	74.1%	71.7%	68.9%	68.9%
DailyDialog	81.3%	80.1%	77.8%	81.3%

length of the longer string) between the completions of gpt 4 versus the completions of the smaller models and the datasets, which are shown in Table 4. The average percent edit distance between gpt 4 completions and DailyDialog continuations was 81.3%, indicating that gpt 4 was generating strings that were substantially different from the dataset. Similar trends were found when comparing davinci-001 and davinci-003 against the completions of the other models and the datasets (Tables 3 and 2).

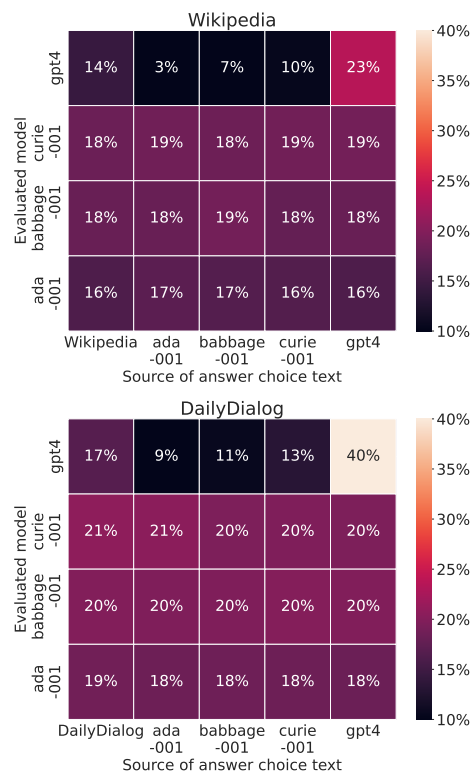


Figure 9: The proportion of the time that each model (ada-001, babbage-001, curie-001, and gpt4) selects each possible answer choice when prompted with a hypothetical consistency prompt. Model outputs that could not be parsed into an answer choice are not included.