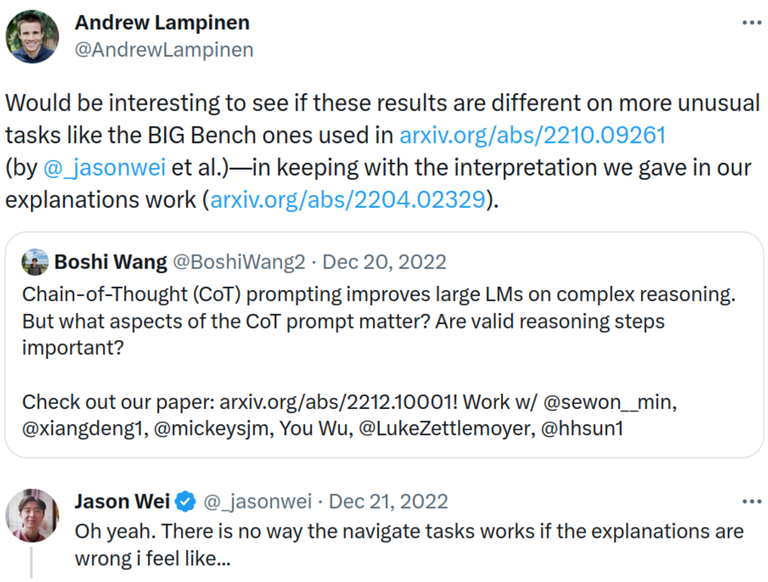
Invalid Logic, Equivalent Gains: The Bizarreness of Reasoning in Language Model Prompting

Rylan Schaeffer* [1] Kateryna Pistunova* [2] Samar Khanna* [1] Sarthak Consul* Sanmi Koyejo [1]

[1]Computer Science, Stanford [2]Physics, Stanford

Summary

- Language models can be prompted to reason through problems in a manner that improves performance
- *Why* such prompting improves performance is unclear
- Wang et al. [4] recently showed that *logically invalid* Chain-of-Thought (CoT) prompting [1, 5] improves performance almost as much as logically valid CoT prompting



Andrew Lampinen
@AndrewLampinen

Would be interesting to see if these results are different on more unusual tasks like the BIG Bench ones used in arxiv.org/abs/2210.09261 (by @_jasonwei et al.)—in keeping with the interpretation we gave in our explanations work (arxiv.org/abs/2204.02329).

> **Boshi Wang** @BoshiWang2 · Dec 20, 2022
> Chain-of-Thought (CoT) prompting improves large LMs on complex reasoning. But what aspects of the CoT prompt matter? Are valid reasoning steps important?
>
> Check out our paper: arxiv.org/abs/2212.10001! Work w/ @sewon__min, @xiangdeng1, @mickeysjm, You Wu, @LukeZettlemoyer, @hhsun1

Jason Wei @_jasonwei · Dec 21, 2022
Oh yeah. There is no way the navigate tasks works if the explanations are wrong i feel like...

- Critics responded Wang et al.'s finding was based on too few & too easily solved tasks to draw conclusions
- To resolve this dispute, we test whether *logically invalid* CoT prompts offer the same performance gains on the hardest tasks in BIG-Bench [2], termed BIG-Bench Hard (BBH) [3]
- Logically invalid CoT prompts **DO** achieve similar performance gains on BBH
- We also discover some CoT prompts used by previous works contain logical errors

Background: BIG-Bench [2] & BIG-Bench Hard (BBH) [3]

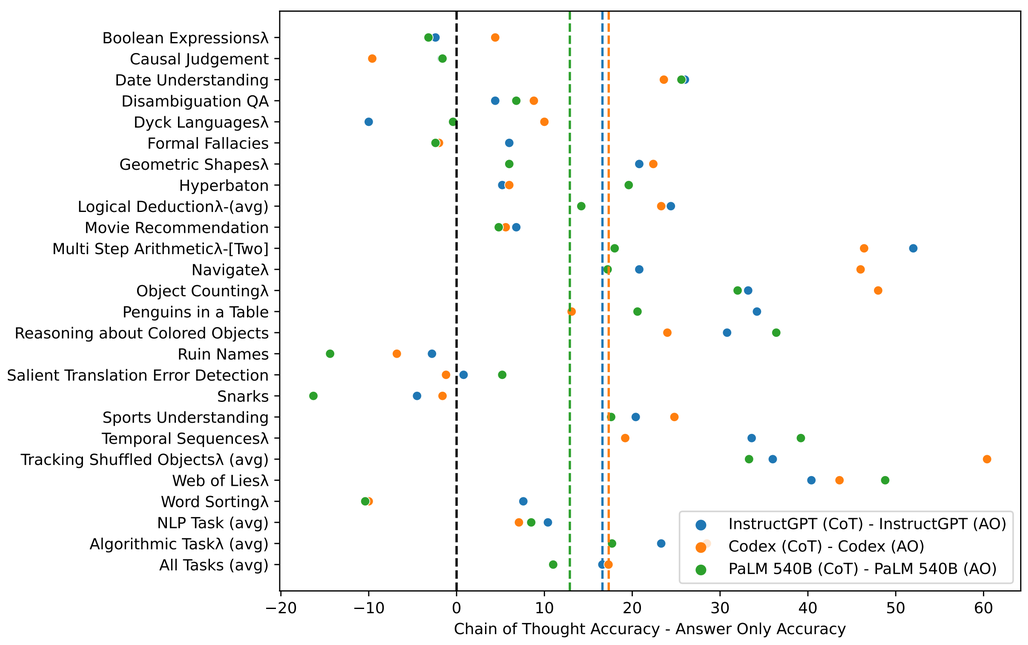
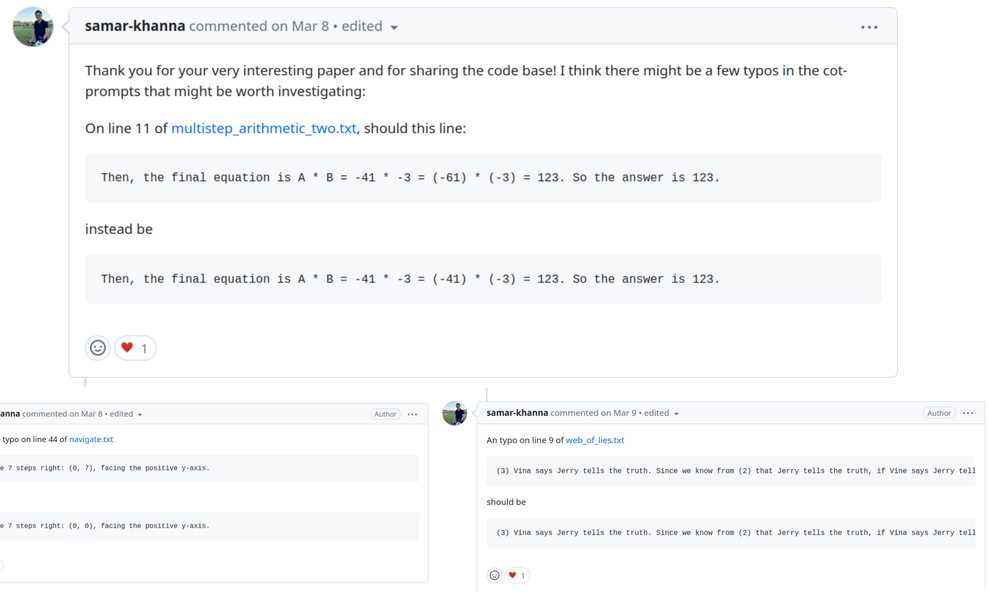


Figure 1. Chain-of-Thought (CoT) prompting significantly outperforms answer-only (AO) prompting on BIG-Bench Hard [3] = 23 of the hardest tasks in Beyond the Imitation Game Benchmark [2].

Discovery: BIG-Bench Hard's CoT Prompts Contain Errors!



samar-khanna commented on Mar 8 • edited

Thank you for your very interesting paper and for sharing the code base! I think there might be a few typos in the cot-prompts that might be worth investigating:

On line 11 of multistep_arithmetic_two.txt, should this line:

```
Then, the final equation is A * B = -41 * -3 = (-61) * (-3) = 123. So the answer is 123.
```

instead be

```
Then, the final equation is A * B = -41 * -3 = (-41) * (-3) = 123. So the answer is 123.
```

❤ 1

samar-khanna commented on Mar 8 • edited — Author

A possible typo on line 44 of navigate.txt

```
(4) Take 7 steps right: (0, 7), facing the positive y-axis.
```

should be

```
(4) Take 7 steps right: (0, 0), facing the positive y-axis.
```

❤ 1

samar-khanna commented on Mar 9 • edited — Author

An typo on line 9 of web_of_lies.txt

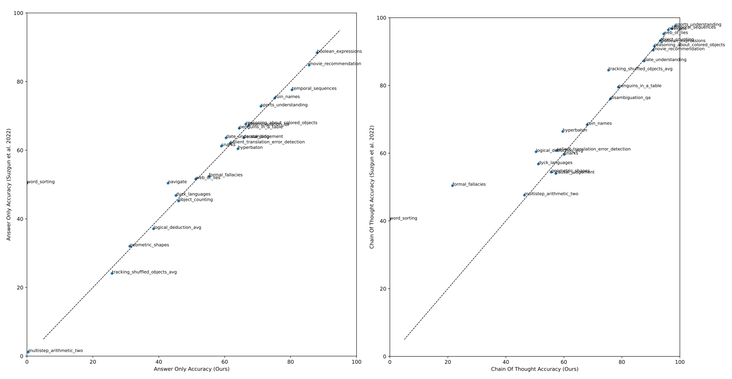
```
(3) Vina says Jerry tells the truth. Since we know from (2) that Jerry tells the truth, if Vine says Jerry tell
```

should be

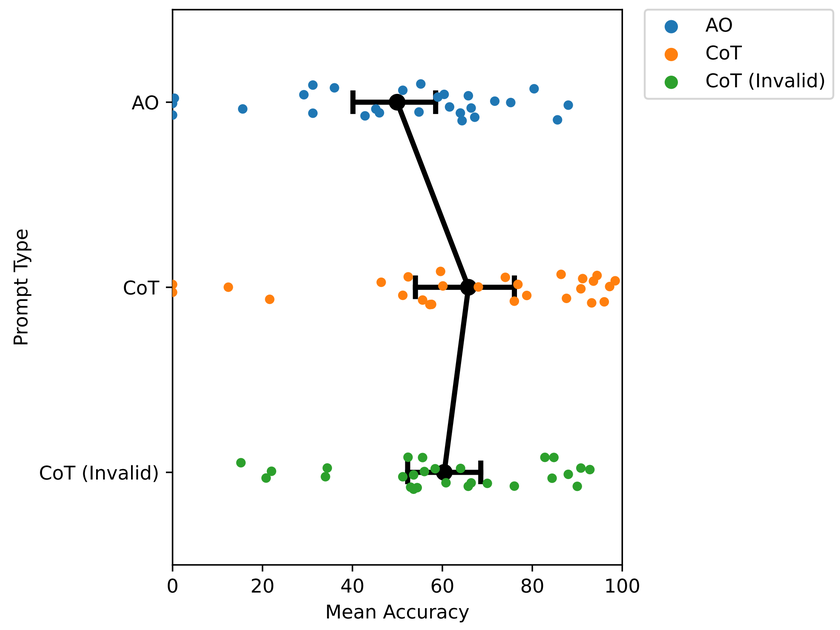
```
(3) Vina says Jerry tells the truth. Since we know from (2) that Jerry tells the truth, if Vina says Jerry tell
```

❤ 1

Reproducing BIG-Bench Hard's Results



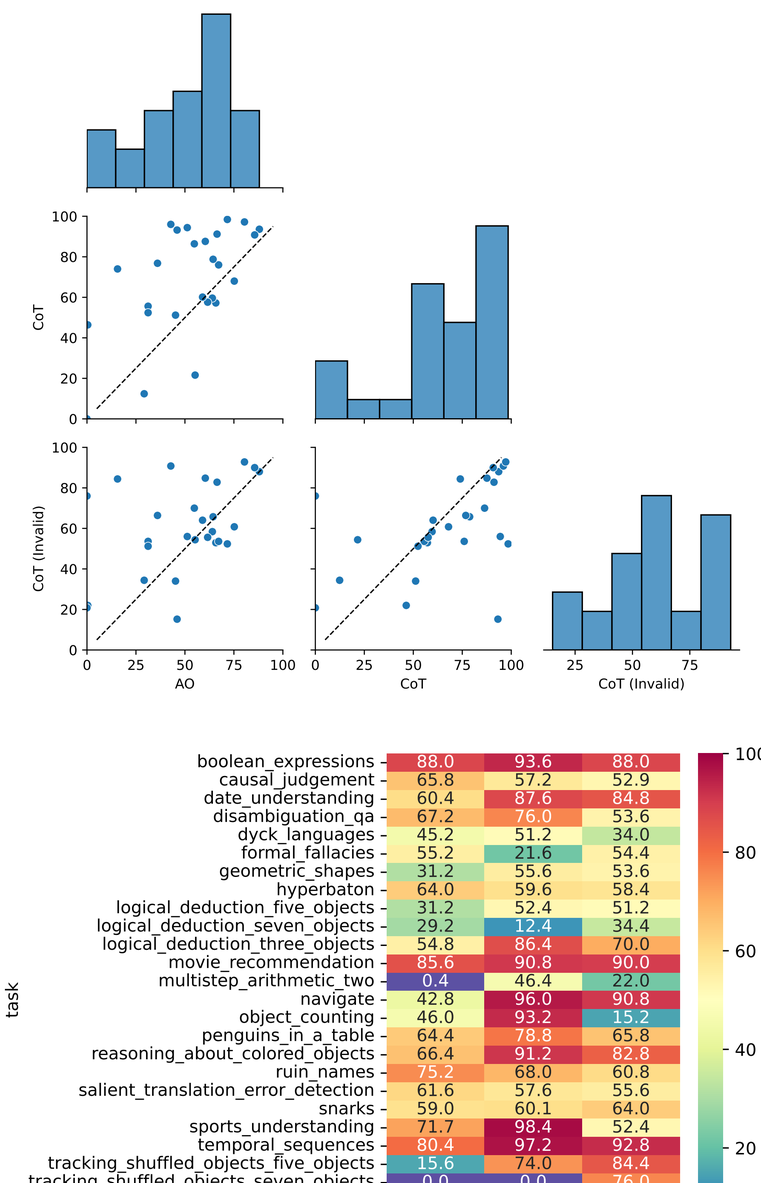
Logically Invalid CoT Prompting Almost Matches CoT on BBH



Note: We do *not* optimize logically invalid CoT prompts whatsoever. Write, run, done.

Example: Evaluate the result of a random Boolean expression. Q: not ((not not True)) is

Answer Only (AO)	False
Scratchpad/Chain-of-Thought (CoT)	A: Let's think step by step. Remember that (i) expressions inside brackets are always evaluated first and that (ii) the order of operations from highest priority to lowest priority is "not", "and", "or", respectively. We first simplify this expression "Z" as follows: "Z = not ((not not True)) = not ((A))" where "A = not not True". Let's evaluate A: A = not not True = not (not True) = not False = True. Plugging in A, we get: Z = not ((A)) = not ((True)) = not True = False. So the answer is False.
Logically-Invalid Chain-of-Thought (Invalid CoT)	A: Let's think step by step. Remember that (i) expressions inside brackets are always evaluated first and that (ii) the order of operations from highest priority to lowest priority is "not", "and", "or", respectively. Because English does not permit multiple negatives, the expression "(not not True)" evaluates to "(not True)". The expression "not ((not not True))" therefore evaluates to "not ((not True))". By the same logic, the expression "not ((not True))" simplifies to "not True". In Boolean logic, "not True" is False. So the answer is False.



References

[1] M. Nye, A. J. Andreassen, G. Gur-Ari, H. Michalewski, J. Austin, D. Bieber, D. Dohan, A. Lewkowycz, M. Bosma, D. Luan, et al. Show your work: Scratchpads for intermediate computation with language models, 2021.

[2] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2022.

[3] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. V. Le, E. H. Chi, D. Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022.

[4] B. Wang, S. Min, X. Deng, J. Shen, Y. Wu, L. Zettlemoyer, and H. Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters, 2022.

[5] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models, 2022.