A Causal Framework to Quantify the Robustness of Mathematical Reasoning with Language Models

Alessandro Stolfo[*]
ETH Zürich

Zhijing Jin[*]
ETH Zürich & MPI

Kumar Shridhar
ETH Zürich

Bernhard Schölkopf
ETH Zürich & MPI

Mrinmaya Sachan
ETH Zürich

Abstract

We have recently witnessed a number of impressive results on hard mathematical reasoning problems with large language models (LLMs). At the same time, the robustness of these models has also been called into question. Building on the idea of behavioral testing, we propose a novel framework, which pins down the causal effect of each factor in the input, e.g., the surface form of the problem text, the operands, and math operators, on the output. By grounding the behavioral analysis in a causal graph describing an intuitive reasoning process, we study the behavior of LLMs in terms of robustness and sensitivity to direct interventions in the input space. We apply our framework on a test bed of bivariate math word problems. Our analysis shows that robustness does not appear to continuously improve as a function of scale, but that the recent LLM, GPT-3-Instruct (175B), achieves a dramatic improvement in both robustness and sensitivity, compared to all other GPT variants. The full paper is available at `https://arxiv.org/abs/2210.12023`[2].

1 Introduction

Math reasoning has been a longstanding challenge for AI [2], as it requires both the linguistic ability to map a problem into a set of mathematical operations, and the ability to execute the math operations correctly. While there has been a lot of work on building supervised domain-specific solvers for these problems in the past decade [8; 10; 20; 25; 21–23, *inter alia*], recently, we have seen astounding progress in this area led by the development of large language models (LLMs) [4; 5] and nuanced ways to prompt them [6; 28; 30]. Yet, the robustness of these models on the math reasoning tasks remains questionable [14; 18].

A well-known way to check robustness of models is behavioral testing using a *CheckList* [19]. CheckLists are metamorphic tests (as in software engineering), such as invariance tests and directional expectation tests, used to identify critical failures in our models. Inspired by the robustness study presented in Patel et al. [14], we investigate the robustness of the reasoning in LLMs, building our approach on the idea of behavioral testing that underlies the CheckList framework.

To achieve this goal, we propose a causal framework to quantify the robustness of NLP models' math reasoning ability. Specifically, we first describe a causal graph formulation of math reasoning, where the goal is to quantify the difference in the structural causal models (SCMs) of human reasoning and model judgment. We consider causal factors such as the textual framing of the question, number operands, and operations. Then, we identify the set of interventions feasible in the context of math

[*]Equal contribution, correspondence to `stolfoa@ethz.ch`.

[2]Our code and data are available at `https://github.com/alestolfo/causal-math`.

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Workshop on Math-AI.

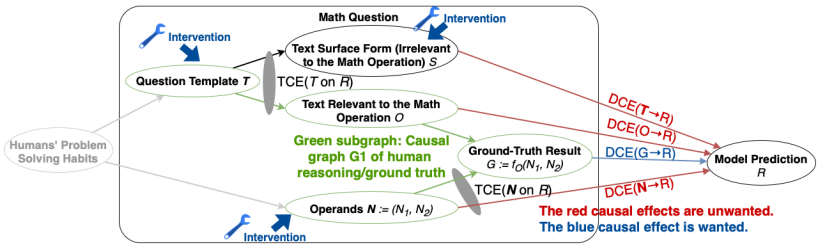

Figure 1: Causal graph of model predictions on math questions. Explained in detail in the text.

word problems (MWPs), and provide a causal inference framework to obtain causal influences of each factor via direct do-interventions [15] and causal mediation analysis [16]. Using our framework, we disentangle the factors affecting the model's predictions and measure their influence. This way, we are able to provide insights into the model's reasoning in terms of *robustness* and *sensitivity*.

Finally, we apply our framework and evaluate a series of GPT models. We show that larger GPT models tend to be more sensitive to changes in the ground-truth result of a MWP, but not significantly more robust. An exception to thus phenomenon is the most recent and largest variant of Instruct-GPT-3 [13], which shows a remarkable improvement in both sensitivity and robustness.

2 The Framework

We present our framework for bivariate MWPs with a single arithmetic operation (addition, subtraction, multiplication or division). This framework can be extended to more variables and other math problems in future work.

We consider a dataset $\mathcal{D}$ of MWPs, where each problem is denoted as a question $\boldsymbol{Q}$. $\boldsymbol{Q}$ is an ordered list $(\boldsymbol{t}, (n_1, n_2), g)$ consisting of a question template $\boldsymbol{t}$ with two operands n_1, n_2, and the ground-truth result g. Each question template $\boldsymbol{t} := (o, s)$ further contains two types of information: the arithmetic operation type $o \in \{+, -, \times, \div\}$ implicitly expressed in the question, and the text surface form s irrelevant to the arithmetic operation. The ground-truth result $g = f_o(n_1, n_2)$ is calculated by applying the operation $f_o(\cdot, \cdot)$ on the two operands. An example math question in this form from Patel et al. [14] is: ($\boldsymbol{t}$ = "Mark has n_1 trees in his backyard. If he plants n_2 more, how many trees will he have?", $(n_1 = 12, n_2 = 13), g = f_o(n_1, n_2) = n_1 + n_2 = 25$).

Our goal is to quantify the reasoning abilities of a model $\mathcal{M}$ on the set of problems $\boldsymbol{Q} \in \mathcal{D}$. We assume that $\mathcal{M}$ takes $\boldsymbol{Q}$ as input and predicts a probability distribution of the result R: $P(R|(\boldsymbol{t}, (n_1, n_2)))$. Our formulation below will be easier to understand using this finite discrete set, and can be generalized to infinite or continuous sets for other types of operands in future work.

2.1 Question Reformulation

We address the research question: *Is a model reasoning robustly on MWPs?* by comparing the causal mechanisms of the model's decisions to an hypothesized human reasoning mechanism. Note that we do not claim to know how humans reason about these problems. We simply propose a reasonable and intuitive mechanism inspired by studies on the independence of language and mathematical reasoning in humans [3; 12].

Human reasoning mechanism. The causal mechanisms of how humans might solve $\mathbf{Q}$ include $o = f_{\text{abstract}}(\boldsymbol{Q})$ and $g = f_o(n_1, n_2)$, where they first abstract the arithmetic operation o from the problem $\boldsymbol{Q}$ by some cognitive process f_{abstract}, and then apply the operation to the operands to obtain the result g. We show these mechanisms in the green subgraph $\mathcal{G}_1$ of Figure 1.

Model reasoning mechanism. In contrast, the causal mechanisms of how a model might solve $\boldsymbol{Q}$ are $r = f_{\text{blackBox}}(\boldsymbol{t}, (n_1, n_2))$, where we are unsure about *what* part(s) of $\boldsymbol{Q}$ the model takes into account, and *how* it operates over the relevant variables.
Thus, we draw all possible causal mechanisms that might take place in the black-box model f_{blackBox} in the model causal graph $\mathcal{G}_2$ in Figure 1. (1) The model might attend over the question template $\boldsymbol{t}$

in two ways: paying attention to the text surface form s via the causal path $\boldsymbol{T} \rightarrow S \rightarrow R$, or text relevant to the math operation o via the causal path $\boldsymbol{T} \rightarrow O \rightarrow R$. Moreover, (2) the model might also attend to the operands $\boldsymbol{n} := (n_1, n_2)$ via a causal path $\boldsymbol{N} \rightarrow R$. Finally, (3) if the model learns the correct causal mechanisms as in the human cognitive process, it should capture how the operator and the operands matter to the ground-truth result g (via $O \rightarrow G$ and $\boldsymbol{N} \rightarrow G$) and then the model prediction should be sensitive to any changes in the ground truth, namely $G \rightarrow R$. No spurious correlations can directly affect R without going through the mediator G.

Hence, to answer the question "How robust is the mathematical reasoning of a model on MWPs?" we can answer the following subquestions:

1. How does R change in response to G? By quantifying this, we assess the *sensitivity* (correct responsiveness) of the model to changes in the problem. In other words, does the model correctly adjust its prediction in response to a change in the correct solution of the problem?
2. What is the (unwanted) direct causal effect size of $S \rightarrow R$, and $\boldsymbol{N} \rightarrow R$? We see the quantities as a measure of the *brittleness* (i.e., wrong responsiveness) of the model to result-preserving changes in the input. The lower the direct causal effect of S and $\boldsymbol{N}$, more *robust* the model is.

2.2 Interventions and Causal Effect Measured

After formulating the causal graph, we identify the feasible actions that allow us to perform our causal analysis. In the context of MWPs, we perform (1) direct intervention on all possible n_1, n_2, and (2) partially controllable interventions on $\boldsymbol{T}$. We can replace the template $\boldsymbol{T}$ in one of the two ways: (2a) both S and O are affected, or (2b) S is affected but O is not affected.

Causal Effects of the Operands. When intervening on the operands $\boldsymbol{N} := (N_1, N_2)$, we can obtain the size of the **total causal effect** (TCE, i.e., the joint effect through all the directed causal paths from a variable to another) of $\boldsymbol{N}$ on R, namely

$$\mathrm{TCE}(\boldsymbol{N} \text{ on } R) := \mathbb{E}^{\mathrm{int}+}_{\boldsymbol{N}}[R] - \mathbb{E}^{\mathrm{int}-}_{\boldsymbol{N}}[R]. \tag{1}$$

Here, $\mathbb{E}^{\mathrm{int}+}_{\boldsymbol{N}}[R]$ denotes the expected result after intervention on N and $\mathbb{E}^{\mathrm{int}-}_{\boldsymbol{N}}[R]$ denotes the expected result prior to the intervention.

We can quantify the **direct causal effect** (DCE, i.e., the effect from the directed causal path from a variable to another that does not go through any intermediate variables) [16] of $\boldsymbol{N}$ on R, namely the strength of the direct causal path $\boldsymbol{N} \rightarrow R$, by controlling for G to be fixed when we intervene on $\boldsymbol{N}$:

$$\mathrm{DCE}(\boldsymbol{N} \rightarrow R) := \sum_{g} P(G)(\mathbb{E}^{\mathrm{int}+}_{\boldsymbol{N}}[R|G=g] - \mathbb{E}^{\mathrm{int}-}_{\boldsymbol{N}}[R|G=g]) \,. \tag{2}$$

Causal Effects of the Text Surface Form. As for the operands, we can compute both the direct and indirect effects of the surface form representing the math problem. In particular, intervening on $\boldsymbol{T}$ without controlling for O (intervention 2a above), we can compute the total effect, i.e.,

$$\mathrm{TCE}(\boldsymbol{T} \text{ on } R) := \mathbb{E}^{\mathrm{int}+}_{\boldsymbol{T}}[R] - \mathbb{E}^{\mathrm{int}-}_{\boldsymbol{T}}[R]. \tag{3}$$

Controlling for the operation O (intervention 2b above) will instead allow us to obtain the direct causal effect of the surface text:

$$\begin{aligned} \mathrm{DCE}(S \rightarrow R) &:= \mathbb{E}^{\mathrm{int}+}_{S}[R] - \mathbb{E}^{\mathrm{int}-}_{S}[R] && (4) \\ &= \sum_{o} P(O)(\mathbb{E}^{\mathrm{int}+}_{\boldsymbol{T}}[R|O=o] - \mathbb{E}^{\mathrm{int}-}_{\boldsymbol{T}}[R|O=o]) \,. && (5) \end{aligned}$$

The only adaptation that we need to make with regard to the MWPs is that it is not feasible to enumerate all possible perturbations of S. Therefore, the practical results that researchers can achieve are over a certain subset of S. In practice, we obtain this effect by replacing the original template with different one describing the same operation o. We provide the details and the formal description of our intervention procedure in Appendix C.

2.3 Quantifying the Causal Influence

Given a pair of problems $\boldsymbol{Q} : \{\boldsymbol{t}, (n_1, n_2), g\}$ and $\boldsymbol{Q'} : \{\boldsymbol{t'}, (n'_1, n'_2), g'\}$ representing an intervention $\mathrm{do}(X : x \to x')$, where $X \in \{\boldsymbol{T}, S, \boldsymbol{N}\}$, denote the distribution before the intervention as $P(R \mid (\boldsymbol{t}, (n_1, n_2)))$ as P and the distribution after intervention $P(R \mid (\boldsymbol{t'}, (n'_1, n'_2)))$ as P'. The support of R is $\mathcal{R}$, the set of possible results. We quantify the causal effect of a factor X on the model's prediction R in two ways. The first one is by assessing the change in the the predicted result. That is, we compute $d_{\mathrm{cp}}(P, P') := \mathbb{1}(r \neq r')$, where $r = \arg\max_{x \in \mathcal{R}} P(x)$, and $r' = \arg\max_{x \in \mathcal{R}} P'(x)$. The second metric that we use is the relative change in the probability assigned by the model to g and g' (d_{rcc}). We provide a detailed definition of the metric in Appendix D.

3 Experiments

For our analyses, we use instances of math word problems from three popular datasets: ASDiv-A [11], MAWPS [9], and SVAMP [14]. We use our framework to assess the robustness of reasoning in eleven pretrained LMs: five sizes of GPT-2 [17] (distilled [24], regular, medium, large, and XL), GPT-Neo 1.3B and 2.7B [1], GPT-J-6B [26], and the Instruct versions [13] of GPT-3 [4] (Babbage, Curie and Davinci).[3]

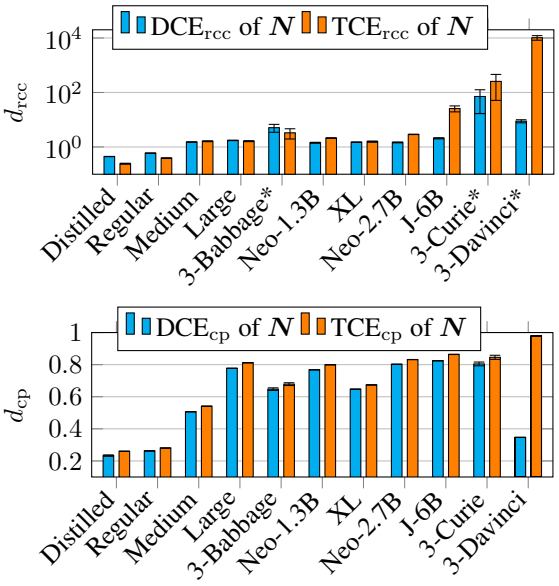

Figure 2: Comparison of $\mathrm{DCE}(\boldsymbol{N} \to R)$ and $\mathrm{TCE}(\boldsymbol{N}$ on $R)$. We use some approximation method for GPT-3 (denoted by *) which is explained in Appendix Appendix F.

3.1 Results

$\mathrm{DCE}(\boldsymbol{N} \to R)$ represents the undesired effect for a model to being mistakenly responsive to a change in $\boldsymbol{N}$ not leading to a change in the result g (low robustness), whereas higher values of $\mathrm{TCE}(\boldsymbol{N}$ on $R)$ indicate a higher ability of the model to correctly adjust the probability weight assigned to the new solution g' after the intervention (high sensitivity). From the results in Figure 2 we notice that larger models exhibit a larger $\mathrm{TCE}_{\mathrm{rcc}}/\mathrm{DCE}_{\mathrm{rcc}}$ ratio. In particular, in GPT-3 Curie and GPT-J-6B, the TCE is, respectively, 3.5x and 12x larger than the DCE. In GPT-3 Davinci, the total causal effect grows as much as 1000x larger than the DCE. The magnitude of the two effects in terms of change of predictions d_{cp} is comparable for all models except GPT-3 Davinci. For small models (Distilled and Regular GPT-2) $\mathrm{DCE}_{\mathrm{cp}}$ and $\mathrm{TCE}_{\mathrm{cp}}$ are considerably smaller than for other models, indicating high robustness but low sensitivity. Contrarily, for InstructGPT-3 we observe a remarkable 63% absolute difference between direct and total effect. We report a different visualization of the direct causal effect of $\boldsymbol{N}$ on the the model's prediction in Appendix E.2.

The results relative to the total causal effect of the question $\boldsymbol{T}$ and the direct causal effect of the irrelevant text elements S on the model's prediction are reported in Appendix E.1. The results observed for the two kinds of intervention $\mathrm{do}(\boldsymbol{T} : \boldsymbol{t} \to \boldsymbol{t'})$ and $\mathrm{do}(\boldsymbol{N} : (n_1, n_2) \to (n'_1, n'_2))$ show similar trends. Small models (Distilled and Regular GPT-2) exhibit low sensitivity to interventions. Larger models (from GPT-2 Medium to GPT-Neo) appear to be more influenced by changes in both $\boldsymbol{N}$ and $\boldsymbol{T}$. However, they display similar sensitivity to both result-altering and result-preserving interventions. An improvement in sensitivity is noticeable in GPT-J and GPT-3 Curie, though not accompanied by an improvement in robustness. A remarkably different behaviour is instead showed by GPT-3 Davinci, which demonstrates substantially higher sensitivity to result-altering interventions (high TCE), and higher robustness (in terms of prediction change). These results seem to support the so-called *emergent abilities* hypothesis [27], which postulates the existence of skills that are displayed

[3]Experiments with GPT-3 are carried out under the constraints set by the OpenAI APIs (`https://openai.com/api/`), which prevent us from computing the causal effect using the same procedure as for the other models. We report the details about how the metrics were computed for GPT-3 in Appendix F.

by large-scale models but are not present in smaller-scale models, and thus cannot be predicted by simply extrapolating the performance improvements on smaller-scale models. In our case, the ability of reasoning robustly appears to develop in an emergent way. Stronger evidence supporting this theory could be obtained evaluating models with size in the range 6-175B parameters.

4 Conclusion

In this paper, we proposed a framework to disentangle and separately measure the effect of different factors influencing the predictions of LLMs. Our framework provides a set of robustness indicators, and also opens new future directions to design behavioral tests of models in a more causal, principled way.

Acknowledgments

This material is based in part upon works supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039B; by the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645; by the John Templeton Foundation (grant #61156); by a Responsible AI grant by the Haslerstiftung; and an ETH Grant (ETH-19 21-1). Alessandro Stolfo is supported by armasuisse Science and Technology through a CYD Doctoral Fellowship. Zhijing Jin is supported by PhD fellowships from the Future of Life Institute and Open Philanthropy, as well as the travel support from ELISE (GA no 951847) for the ELLIS program. We also thank OpenAI Researcher Access Program for granting our team credits to their API.

References

[1] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. URL `https://doi.org/10.5281/zenodo.5297715`. If you use this software, please cite it using these metadata.

[2] Daniel G. Bobrow. Natural language input for a computer problem solving system. Technical report, USA, 1964.

[3] Elizabeth M. Brannon. The independence of language and mathematical reasoning. *Proceedings of the National Academy of Sciences*, 102(9):3177–3178, 2005. doi: 10.1073/pnas.0500328102. URL `https://www.pnas.org/doi/abs/10.1073/pnas.0500328102`.

[4] 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. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

[5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[6] Iddo Drori, Sunny Tran, Roman Wang, Newman Cheng, Kevin Liu, Leonard Tang, Elizabeth Ke, Nikhil Singh, Taylor L Patti, Jayson Lynch, et al. A neural network solves and generates mathematics problems by program synthesis: Calculus, differential equations, linear algebra, and more. *arXiv preprint arXiv:2112.15594*, 2021.

[7] Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*

and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1828–1843, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.144. URL https://aclanthology.org/2021.acl-long.144.

[8] Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. Learning to solve arithmetic word problems with verb categorization. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533, 2014.

[9] Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. MAWPS: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1136. URL https://aclanthology.org/N16-1136.

[10] Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. Learning to automatically solve algebra word problems. In *Association for Computational Linguistics (ACL)*, 2014.

[11] Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.92. URL https://aclanthology.org/2020.acl-main.92.

[12] Martin M Monti, Lawrence M Parsons, and Daniel N Osherson. Thought beyond language: Neural dissociation of algebra and natural language. *Psychological science*, 23(8):914–922, 2012.

[13] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155, 2022. doi: 10.48550/arXiv.2203.02155. URL https://doi.org/10.48550/arXiv.2203.02155.

[14] Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168.

[15] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

[16] Judea Pearl. Direct and indirect effects. In Jack S. Breese and Daphne Koller, editors, *UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, University of Washington, Seattle, Washington, USA, August 2-5, 2001*, pages 411–420. Morgan Kaufmann, 2001. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1\&smnu=2\&article_id=126\&proceeding_id=17.

[17] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

[18] Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*, 2022.

[19] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442.

[20] Subhro Roy, Tom Vieira, and Dan Roth. Reasoning about quantities in natural language. *Transactions of the Association for Computational Linguistics (TACL)*, 1, 2015.

[21] Mrinmaya Sachan and Eric Xing. Learning to solve geometry problems from natural language demonstrations in textbooks. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pages 251–261, 2017.

[22] Mrinmaya Sachan, Kumar Dubey, and Eric Xing. From textbooks to knowledge: A case study in harvesting axiomatic knowledge from textbooks to solve geometry problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 773–784, 2017.

[23] Mrinmaya Sachan, Kumar Avinava Dubey, Tom M Mitchell, Dan Roth, and Eric P Xing. Learning pipelines with limited data and domain knowledge: A study in parsing physics problems. *Advances in Neural Information Processing Systems*, 31, 2018.

[24] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC^{2}Workshop*, 2019.

[25] Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. Solving geometry problems: Combining text and diagram interpretation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1466–1476, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1171. URL `https://aclanthology.org/D15-1171`.

[26] Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, May 2021.

[27] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[28] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903, 2022. URL `https://arxiv.org/abs/2201.11903`.

[29] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. HuggingFace's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[30] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models, 2022. URL `https://arxiv.org/abs/2205.10625`.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section A
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section B
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section G

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Section C

(b) Did you mention the license of the assets? [Yes] See Section B

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Supplementary material

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The data is available under the MIT license.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Section B

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Limitations

A key limitation in our work is that LLMs might have seen these math problems. Our work theoretically assumes this is not the case. Another limitation is that for sake of simplicity, our work makes some assumptions. For example, we assume all numbers in the range of integers 0 to C=300. This would not cover every MWP out there. And future work is needed to generalize our framework to other forms of MWPs. In this work, we are also constrained by the limitations of the OpenAI policy on the GPT-3 API. This limits the number of perturbations we consider in this work as well as the accuracy with which we can estimate our causal distributions. Finally, our work is restricted to English, and extending it to other languages will require us to create a MWP dataset in that language.

B Ethical Considerations

As for the ethical practice in this work, the data involved are from existing MWP datasets with no private user information, and available under the MIT license. As for the ethical impact of the use of this work, the study is about providing a metric and analyzing existing models' robustness, so there is less concern over harmful usage. Rather, it is more about putting checks on existing AI models and helping humans understand them better before use. Potential stakeholders that could benefit from this research include NLP researchers working on math models, and people involved with applications about math questions in text and e-learning design.

C Details About the Data

Datasets For our analyses, we use instances of math word problems from three popular datasets: ASDiv-A [11], MAWPS [9], and SVAMP [14]. The examples contained in these collections are pairs $(\boldsymbol{t}, o)$ consisting of a question template $\boldsymbol{t}$ with its annotated operation o. Each of these pairs can be instantiated multiple times into problems $\boldsymbol{Q} : ((\boldsymbol{t}, (n_1, n_2)), g)$ by filling the template with numerical values n_1, n_2 and computing the ground-truth result $g = f_o(n_1, n_2)$.

The textual template $\boldsymbol{t}$ consists of a context (describing a real-world state and/or actions) and a question. In order to obtain suitable prompts for the models, we convert the problems' questions into statements where the result of the problem is expected to be the first token after the prompt. E.g., in

the example in section 2, *how many trees will he have?* is converted into *the number of trees that he will have is*. We consider templates describing a two-variable expression from the union of the three datasets, and we filter out instances for which the conversion into statement is not possible.

Prompt Creation From the MWP templates of the SVAMP/ASDiv-A/MAWPS collection (we consider all splits), we select the templates describing a simple two-variable expression. We then filter out the templates whose questions do not start with *How many...*, and we use spaCy[4] to identify the subject, the object and the verbs in the sentence. This allows us to convert the last sentence of the template from *The number of... is*. This way, we obtain 437 statement-based MWP templates. We manually checked a subset of the templates to identify possible mistakes in the conversion procedure.

We obtain in this way a set of $\sim$400 template-expression pairs that we use to generate pairs of prompts representing an intervention. For the sake of consistency, we keep the notation $\boldsymbol{t}$ to refer to the statement-converted template, and we use $(\boldsymbol{t}, (n_1, n_2))$ to refer to an instantiated template that we use as prompt.

C.1 Intervention Data

Given an MWP $\boldsymbol{Q}$: $((\boldsymbol{t}, (n_1, n_2)), g)$, we generate a second problem instance $\boldsymbol{Q}' \in \{((\boldsymbol{t}', (n_1', n_2')), g') \mid \mathcal{C}\}$ using a set of constraints $\mathcal{C}$ depending on the type of causal effect CE we want to measure and on the considered variable.

Intervening on $\boldsymbol{N}$. When intervening on the numbers in the problem, the sets of constraints $\mathcal{C}$ take the following form:

$$\begin{aligned}\text{CE} = \text{DCE}(\boldsymbol{N} \to R) &\implies \mathcal{C} = \{s = s', o = o', n_1' \neq n_1, n_2' \neq n_2, g' = g\} \\ \text{CE} = \text{TCE}(\boldsymbol{N} \text{ on } R) &\implies \mathcal{C} = \{s = s', o = o', n_1' \neq n_1, n_2' \neq n_2, g' \neq g\}.\end{aligned}$$

That is, the text of the problem is kept unaltered and a set of new numbers $\boldsymbol{N} = \{n_1, n_2\}$ is sampled in such a way that the result g is affected or not depending on the effect what is being measured.

Intervening on $\boldsymbol{T}$. When changing the textual description of the problem, we have:

$$\begin{aligned}\text{CE} = \text{DCE}(S \to R) &\implies \mathcal{C} = \{s \neq s', o = o', n_1' = n_1, n_2' = n_2, g' = g\} \\ \text{CE} = \text{TCE}(\boldsymbol{T} \text{ on } R) &\implies \mathcal{C} = \{s \neq s', o \neq o', n_1' = n_1, n_2' = n_2, g' \neq g\}.\end{aligned}$$

In other words, we change $\boldsymbol{t}$ such that either $o' = o$, or $o' \neq o$. In the former case we sample a different template $\boldsymbol{t}' = (s', o)$ from the set of templates describing the same operation o, in the latter case we sample a new $\boldsymbol{t}'$ describing a different operation.

Given a model P, we use the pair $(\boldsymbol{Q}, \boldsymbol{Q}')$ to obtain a pair of distributions $P(R|(\boldsymbol{t}, (n_1, n_2)))$ and $P(R|(\boldsymbol{t}', (n_1', n_2')))$, which we use to measure the causal effect of the intervention. We consider the result space $\mathcal{R} = \{1, 2, \ldots, C\}$ consisting of integer values, following the setup of several existing MWP datasets [11; 9; 14]. To control our experimental costs and make sure the models keep the number as one token, we set $C = 300$. And we additionally enforce $N_i \in \{1, 2, \ldots, C\}, \forall N_i \in \boldsymbol{N}$. From all the tokens in a model's vocabulary, we focus on the probability assigned to the numbers in our result space $\mathcal{R}$, and thus we use $P(R = r)$ to denote the normalized probability $P_{\text{raw}}(R = r)/Z$, where $Z = \sum_{r=1}^{C} P_{\text{raw}}(R = r)$, and $P_{\text{raw}}(x)$ is the raw probability score assigned to the vocabulary token x. For each intervention type, we generate a dataset $\boldsymbol{D}$ consisting of $(\boldsymbol{Q}, \boldsymbol{Q}')$ pairs. Unless otherwise specified, for our experiments we generate 500 intervention pairs for each template, and results are average over three seeds.

D Details About the Metrics

We use the same notation as in Section 2.3.

Relative Change in Confidence. Inspired by Finlayson et al. [7], we highlight the change in terms of the relative difference in the probability assigned to g and g'. We formulate two types of relative

[4] `https://spacy.io`

change, one quantifying the relative change in the confidence of g, and the other quantifying the relative change in the confidence of g':

$$\Delta_{\text{rel}} = \frac{P(g) - P'(g)}{P'(g)} \tag{6}$$

$$\Delta'_{\text{rel}} = \frac{P'(g') - P(g')}{P(g')} \,. \tag{7}$$

We quantify the overall relative change in confidence (RCC) as the average of the two relative changes above:

$$d_{\text{rcc}}(P, P') = \begin{cases} \frac{1}{2}\left(\Delta_{\text{rel}} + \Delta'_{\text{rel}}\right) & \text{if } g \neq g' \\ \max\left(\Delta_{\text{rel}}, \Delta'_{\text{rel}}\right) & \text{if } g = g' \,. \end{cases} \tag{8}$$

A Unified Form. We are interested in the average causal effect of the intervention across all problems in $\boldsymbol{D}$:

$$\text{CE}_{\text{metric}}(R \mid \text{do}(X : x \to x')) \tag{9}$$

$$= \text{CE}_{\text{metric}}(X \text{ on } R) \tag{10}$$

$$= \mathbb{E}_{\boldsymbol{Q}_i \in \boldsymbol{D}}\left[d_{\text{metric}}(P_i, P'_i)\right], \tag{11}$$

$\forall$ metric $\in \{\text{rcc}, \text{cp}\}$, where P_i and P'_i are the pre- and post-intervention distribution for $\boldsymbol{Q}_i \in \boldsymbol{D}$.

E Additional Results

E.1 Effect of $\boldsymbol{T}$ on R

In Figure 3 we report the total causal effect of the question $\boldsymbol{T}$ and the direct causal effect of the irrelevant text elements S on the model's prediction. The considerations made for the effects of $\boldsymbol{N}$ can be drawn in this case as well: larger models show a larger TCE_{rcc}/DCE_{rcc} ratio. For models smaller than GPT-J, this ratio is ≤ 1, which indicates that an intervention in the textual description of the MWP leads to a comparable effect both when affecting the ground truth result (i.e. when $g = g'$) and when $g \neq g'$. The large TCE_{rcc}/DCE_{rcc} ratio of GPT-3 Davinci ($\sim$280) suggests that the model tends to adjust its prediction accordingly after a result-altering intervention, more than varying the probability score assigned to the correct solution after an intervention that does not affect the result of the problem. For d_{cp}, GPT-3 Davinci shows a substantial difference (57%) between direct and total effect, as observed for $\boldsymbol{N}$.

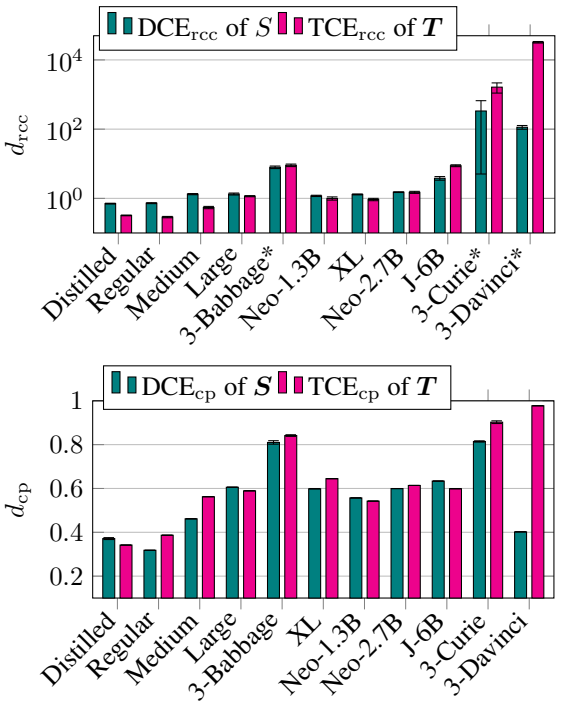

Figure 3: Comparison of $\text{DCE}(S \to R)$ and $\text{TCE}(\boldsymbol{T}$ on $R)$. *approx values, see Appendix F.

E.2 Heatmaps

For a different visualization of the direct causal effect of $\boldsymbol{N}$ on the the model's prediction. We report the heatmaps showing

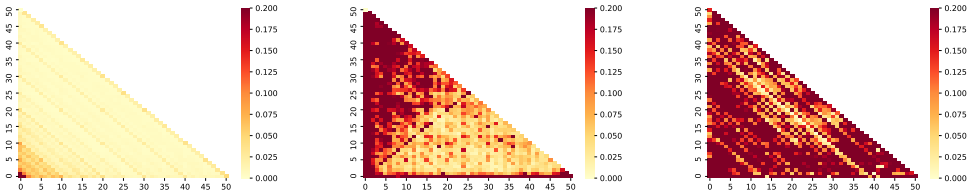

Figure 5: Heatmaps displaying $P(g)$ for Distil-GPT-2 (left) and GPT-J-6B (center) and GPT-3 Davinci (right). The probability values for each combination of $((n_1, n_2), g)$ are averaged over 20 different templates. Probability values over 0.2 are displayed with the darkest color.

the probability assigned by the model to the result g of a math problem $(\boldsymbol{t}, (n_1, n_2), g) \mid g = n_1 + n_2,\ \forall g \in \{0, 1, \ldots, 50\},\ \forall (n_1, n_2) \in \{0, 1, \ldots, 50\} \times \{0, 1, \ldots, 50\}$. For Distil-GPT-2 we observe low overall probability assigned to g and diagonal patterns indicating a consistency in assigning higher probability to specific results (e.g., 10, 20, 30, 40, 50). For the two larger models we notice higher probability mass assigned to the problem's result, but less consistency on the prediction of the same result with different sets of operands (this is true for GPT-J in particular). This result is consistent with the observed higher DCE and TCE in larger models: $P(g)$ might vary more considerably when intervening on the $\boldsymbol{N}$ without affecting g, but overall the model assigns higher probability weight to the correct result, which correlates with higher sensitivity.

E.3 Quantitative Validation of our Framework

We examine the relationship between performance and robustness, computing the Pearson correlation coefficient between accuracy (precision@10) and the relative confidence change (RCC) metric. On a per-template basis (500 instances for each template), we found accuracy to be positively correlated with $\mathrm{TCE}(\boldsymbol{N}$ on $R)$ and $\mathrm{TCE}(T$ on $R)$ (0.24 and 0.49, respectively) and negatively correlated with $\mathrm{DCE}(\boldsymbol{N} \to R)$and $\mathrm{DCE}(S \to R)$ (-0.26 and -0.36, respectively). We see these results as a quantitative validation of the intuition behind our framework: the better the model's performance, the more the model tends to correctly adjust its prediction after a result-altering intervention (higher sensitivity) and to correctly not change its prediction after a result-preserving intervention (higher robustness).

Moreover, We conduct another sanity check as in Patel et al. [14]: removing the question from the MWP templates, we observe a sensitivity-robustness degradation to random guessing. This indicates that the measurement of the causal effects within our framework is not affected by patterns in the templates that might have been picked up or memorized by large models.

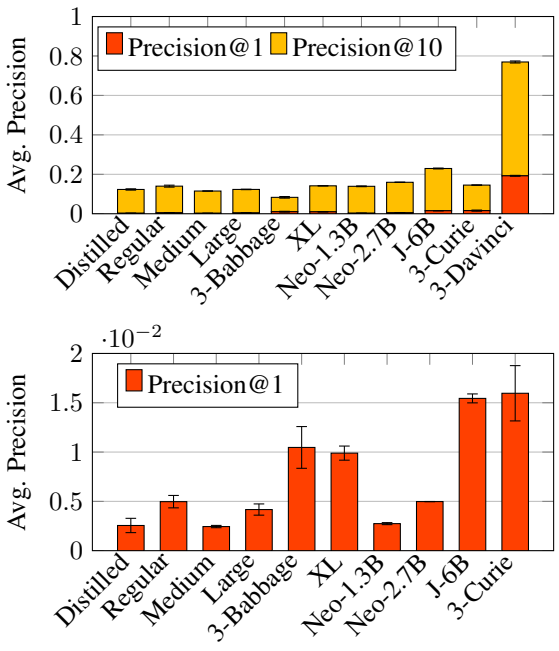

Figure 4: Average precision of the models on the generated instances of MWPs. Results are averaged over two sets consisting of 500 problem instances generated for each template. The lower figures shows a zoomed-in visualization of the precision at 1.

We additionally report in Figure 4 the precision of the models on the generated instances of MWPs, which shows an improvement with the model sizes that follows a similar trend as the robustness/sensitivity changes we observed.

F Computation of Causal Effects for GPT-3

We accessed GPT-3 through the OpenAI APIs, which allow a user to prompt the model and obtain the probabilities assigned by the model to the k-th most likely vocabulary entries, for each token generated. To overcome this limitation, we approximated the the relative probability change d_{rcc} as follows, depending on the kind of effect measured.

The limit for k is set by OpenAI to 5. However, for our main set of experiments (i.e., computing the causal effects of $\boldsymbol{N}$, S, and $\boldsymbol{T}$) we were granted an increased limit of k to 100. This allowed us to obtain reasonable estimates for the causal effects, as the number of cases in which $P(g)$ is not defined are less than 10% of the number of examples that we consider.

F.1 $\mathrm{TCE}(\boldsymbol{N}$ **on** $R)$ **and** $\mathrm{TCE}(\boldsymbol{T}$ **on** $R)$

In cases when $P(g)$ is defined (i.e., when g appears in the top k token predictions) and $P'(g)$ is not defined, we compute a lower bound on the relative change using the upper bound on $P'(g)$ given by the probability of the k-th most likely token. This gives us a conservative estimate of Δ. For cases in which $P(g)$ is not defined, we cannot say anything about the relative change, and we set $\Delta = 0$. The same applies swapping P and P'. This procedure is illustrated by Algorithm 1.

Algorithm 1 Computation of d_{rcc} for GPT-3

$\boldsymbol{Q} = (\boldsymbol{t}, (n_1, n_2), g)$
$\boldsymbol{Q}' = (\boldsymbol{t}', (n_1', n_2'), g')$
if $P(g)$ *is defined* **then**
  if $P'(g)$ *is defined* **then**
    $\Delta = \frac{P(g) - P'(g)}{P'(g)}$
  else
    $\hat{P}' \leftarrow P'(k\text{-th most likely token})$
    $\Delta = \frac{P(g) - \hat{P}'}{\hat{P}'}$
  end
else
  $\Delta = 0$
end
if $P'(g')$ *is defined* **then**
  if $P(g')$ *is defined* **then**
    $\Delta' = \frac{P'(g') - P(g')}{P(g')}$
  else
    $\hat{P} \leftarrow P(k\text{-th most likely token})$
    $\Delta' = \frac{P'(g') - \hat{P}}{\hat{P}}$
  end
else
  $\Delta' = 0$
end
$d_{\mathrm{rcc}} = \frac{1}{2}(\Delta + \Delta')$

F.2 $\mathrm{DCE}(\boldsymbol{N} \to R)$ **and** $\mathrm{DCE}(S \to R)$

In this case we simply discard the examples for which $P(g)$ is not defined or $P'(g)$ are not defined. In that is not the case, then we compute d_{rcc} as in Section D.

F.3 Heatmap Illustration

The heatmap for GPT-3 displayed in Figure 5 was computed by taking the raw probability score produced by the model over the whole vocabulary, as the limit on the available top predicted tokens

makes it impossible to normalize it over the set $\{0, \ldots, 300\}$, as done for the other models. The probability was set to 0 when g did not appear in the model's top 5 predictions for the next token after the prompt.

G Computing Infrastructure & Inference Details

To run our experiments, we use a single NVIDIA TITANRTX with a 24GB memory for all the versions of GPT-2 and GPT-Neo. We use a single NVIDIA A100 with a 40GB memory for GPT-J-6B. We access GPT-3 using the OpenAI APIs. Running the largest locally-stored model (GPT-J-6B) on the four kinds of experiments related to the four kinds of effects measured took $\sim$12 hours, using 500 MWP instances for each of the 437 templates. Due to budget constraints, the experiments on GPT-3 were carried out using 20 examples generated for each template, and took $\sim$7 hours. We use HuggingFace Transformers [29] to access the models except GPT-3. Experiment tracking was carried out using Weights & Biases[5].

[5] `http://wandb.ai/`