

Code Pretraining Improves Entity Tracking Abilities of Language Models

Najoung Kim*

Department of Linguistics
Boston University
najoung@bu.edu

Sebastian Schuster*

Department of Linguistics
University College London
s.schuster@ucl.ac.uk

Shubham Toshniwal*

NVIDIA
stoshniwal@nvidia.com

Abstract

Recent work has provided indirect evidence that pretraining language models on code improves the ability of models to track state changes of discourse entities expressed in natural language. In this work, we systematically test this claim by comparing pairs of language models on their entity tracking performance. Critically, the pairs consist of base models and models trained on top of these base models with additional code data. We extend this analysis to additionally examine the effect of math training, another highly structured data type, and alignment tuning, an important step for enhancing the usability of models. We find clear evidence that models additionally trained on large amounts of code outperform the base models. On the other hand, we find no consistent benefit of additional math training or alignment tuning across various model families.

1 Introduction

Entity tracking, the capacity to track how properties of discourse entities and their relationships change as a discourse unfolds, is an important ability for understanding longer contexts as well as other critical capabilities such as planning. For example, to successfully parse the following recipe, an agent needs to track what happens to the different entities, such as ingredients.

- (1) Put the eggs, sugar, flour, and baking powder in a bowl and mix to form a light batter. Make sure that the final batter does not contain any lumps of flour or sugar.

Kim & Schuster (2023) showed that several Transformer-based large language models (LLMs), such as GPT-3.5, exhibit a non-trivial entity tracking capacity. At the same time, they found that similar models, such as GPT-3, seem to lack this ability. Based on the limited information available about the differences between the GPT-3 and GPT-3.5 models, Kim & Schuster (2023) hypothesized that pretraining on large amounts of code imbues LLMs with entity tracking abilities. However, due to the opacity of training data specifications of these models, it remains unclear whether code pretraining indeed is the critical difference. In this work, we re-evaluate this claim with open-source LLMs for which more information about the pretraining process is available. We extend our analysis to the effect of math and instruction tuning as well as code. Upon comparing pairs of base models and models additionally trained on code, math, or alignment tuned,¹ we find a clear benefit of code training but no consistent benefit of math training or alignment tuning.

*Equal contribution.

¹We use the term alignment tuning to refer to various methods of making language models more useful for interactive settings, including supervised instruction finetuning (SFT) (Wei et al., 2022a), reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), and direct preference optimization (DPO) (Rafailov et al., 2023).

Exp	Model	Size	Additional Training			Base Model
			+Code	+Math	+Instruct/Chat	
Exp 1 (code)	Llama 2	7B–70B				-
	Code Llama	7B–70B	✓			Llama 2
	DeepSeek	7B				-
	DeepSeek-Coder	7B	✓			DeepSeek
	Gemma	8B				-
	CodeGemma	8B	✓			Gemma
Exp 2 (math)	Llama	7B				-
	FLoat	7B		✓ (instruct)		Llama
	Mistral	7B				-
	OpenMathMistral	7B		✓ (instruct)		Mistral
	DeepSeek-Coder	7B	✓			DeepSeek
	DeepSeek-Math	7B	✓	✓		DeepSeek-Coder
	Code Llama	7B, 34B	✓			Llama 2
	Llemma	7B, 34B	✓	✓		Code Llama
Exp 3 (alignment)	Llama 2	7B–70B				-
	Llama 2-Chat	7B–70B			✓	Llama 2
	Code Llama	7B–70B	✓			Llama 2
	Code Llama-Instruct	7B–70B	✓		✓	Code Llama 2
	Gemma	8B				-
	Gemma-Instruct	8B			✓	Gemma
	CodeGemma	8B	✓			Gemma
	CodeGemma-Instruct	8B	✓		✓	CodeGemma
	DeepSeek	7B				-
	DeepSeek-Chat	7B			✓	DeepSeek
	DeepSeek-Coder	7B	✓			DeepSeek
	DeepSeek-Coder-Instruct	7B	✓		✓	DeepSeek-Coder

Table 1: Summary of the models compared and their pretraining data composition.

2 Related work

Including code in the pretraining data mixture, even for models not explicitly specialized for code, has become increasingly customary in LLM training (Chowdhery et al., 2023; Touvron et al., 2023b; Gemini Team et al., 2024; Groeneveld et al., 2024, *i.a.*). In addition to serving the popular use case of LLMs in code completion and generation (Chen et al., 2021), adding code to the pretraining data mixture has been claimed to improve general reasoning capacities of LLMs (Fu et al., 2022; Ma et al., 2024; Yang et al., 2024). Kim & Schuster (2023) hypothesized that a concrete capacity that can benefit from code is entity tracking: converging evidence towards this claim is contributed by observations from Madaan et al. (2022) (code pretrained models like Codex perform better than models primarily trained on language data on ProPara (Dalvi et al., 2019)), Sap et al. (2022) (GPT-3.5 performs better on object tracking than GPT-3), and Muennighoff et al. (2023) (adding code to the pretraining data improves performance the on bAbI tasks (Weston et al., 2016)). Furthermore, Prakash et al. (2024) observed that a base model finetuned on arithmetic tasks improved performance on a simplified version of the entity tracking task by Kim & Schuster (2023), suggesting that structured data in general beyond code may contribute to the development of an entity tracking capacity in language models.

3 Experiments

We aim to systematically test the hypothesis that code pretraining leads to better entity tracking put forward by Kim & Schuster (2023), through a series of experiments comparing base models and models continued to be trained on code on top of the base models. We additionally test the hypothesis that pretraining on math, another type of structured data, leads to better entity tracking performance through similar comparisons.

2-shot prompt

Given the description after "Description:", write a true statement about all boxes and their contents to the description after "Statement:".

Description: Box 0 contains the car, Box 1 contains the cross, Box 2 contains the bag and the machine, Box 3 contains the paper and the string, Box 4 contains the bill, Box 5 contains the apple and the cash and the glass, Box 6 contains the bottle and the map.

Statement: Box 0 contains the car, Box 1 contains the cross, Box 2 contains the bag and the machine, Box 3 contains the paper and the string, Box 4 contains the bill, Box 5 contains the apple and the cash and the glass, Box 6 contains the bottle and the map.

Description: Box 0 contains the car, Box 1 contains the cross, Box 2 contains the bag and the machine, Box 3 contains the paper and the string, Box 4 contains the bill, Box 5 contains the apple and the cash and the glass, Box 6 contains the bottle and the map. Remove the car from Box 0. Remove the paper and the string from Box 3. Put the plane into Box 0. Move the map from Box 6 to Box 2. Remove the bill from Box 4. Put the coat into Box 3.

Statement: Box 0 contains the plane, Box 1 contains the cross, Box 2 contains the bag and the machine and the map, Box 3 contains the coat, Box 4 contains nothing, Box 5 contains the apple and the cash and the glass, Box 6 contains the bottle.

Description: {description}

Statement: Box 0 contains

Table 2: Prompts with 2-shot in-context demonstrations.

3.1 Models

We selected model pairs that have been reported to vary only in terms of their pretraining data. For testing the code hypothesis, we compared the following pairs of models: (Llama 2, Code Llama), (DeepSeek, DeepSeek-Coder), and (Gemma, CodeGemma), where the second model in each pair is obtained by continuing to train the first model on additional code data. We tested 7B, 13B, and 70B models in the Llama 2 series. For testing the math hypothesis, we compared the following four pairs of models: (Code Llama, Llemma), (DeepSeek-Coder, DeepSeek-Math), (Llama, FLoat), and (Mistral, OpenMathMistral). Again, the second model in each pair is obtained by training the first model on additional math data. For alignment tuning, we compared (Llama 2, Llama 2-chat), (Code Llama, Code Llama-Instruct), (Gemma, Gemma-Instruct), (CodeGemma, CodeGemma-Instruct), (DeepSeek, DeepSeek-Chat), and (DeepSeek-Coder, DeepSeek-Coder-Instruct). These comparisons are summarized in Table 1. See Appendix A.1, Table 4 for more details about the models.

3.2 Evaluation setup

We adopted the boxes task (the “base” version) from Kim & Schuster (2023) for testing the models’ entity tracking capacity. In this task, the input to the LLM is a textual description of the contents of seven boxes followed by 1–12 descriptions of operations that change the contents of the individual boxes. In response to this input, the LLM is prompted to state the contents of each box according to the initial description and the state-changing operations. We used the same prompt and 2-shot in-context learning examples as Kim & Schuster (2023) (see Table 2 for an example). We used a slightly different prompt format for chat-optimized models to align the task better to the input format the models were trained on. The inputs to the model are provided as “user” prompts, and the expected model outputs are formatted as “assistant” (see Table 5 in Appendix for an example prompt).

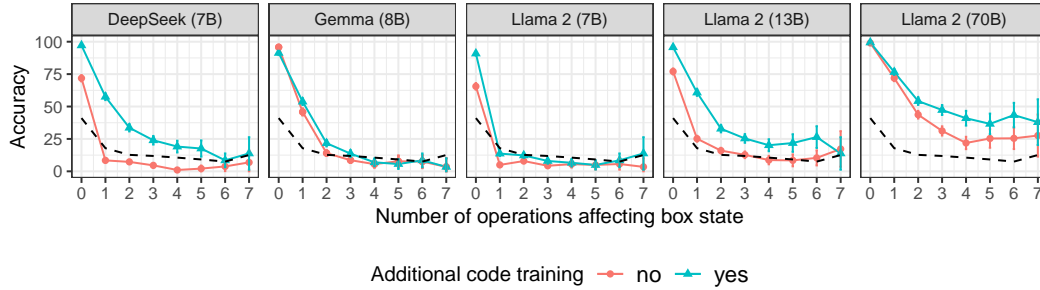


Figure 1: Entity tracking results for DeepSeek, Gemma, and Llama 2 models. Error bars indicate 95% confidence intervals, and the black dashed lines show the performance of the random baseline.

We noticed that smaller models suffered from formatting issues, often deviating from the format specified by the prompt or omitting the contents of some boxes. For this reason, we used regular expression-based constrained decoding using the `outlines` library (Willard & Louf, 2023).²

We report all results divided into the number of operations affecting the target box rather than reporting one aggregate accuracy metric. This is to distinguish trivial cases from cases that actually require tracking state changes—when the number of operations affecting the target box is 0, simply copying from the initial state description yields the correct answer. Furthermore, we compare the model results to the strong random baseline by Kim & Schuster (2023). For this baseline, we randomly sample 0 to 3 objects for each box from the set of objects that have been previously mentioned in a clause with the box in question.

3.3 Experiment 1: Effect of Code

Figure 1 compares the entity tracking performance of base models (red lines) and code models (blue lines) for models from the DeepSeek (DeepSeek-AI et al., 2024; Guo et al., 2024), Gemma (Gemma Team et al., 2024), and Llama 2 families of various sizes (Touvron et al., 2023b; Rozière et al., 2024). In general, we find clear evidence that continued training on large amounts of code improves entity tracking abilities, as can be seen for the Llama 2 13B and 70B models as well as for the DeepSeek models. In these model comparisons, the models trained on code consistently outperformed the base models on the nontrivial cases of entity tracking (number of operations affecting box state ≥ 1). In the case of 13B models, a boost in trivial cases is also observed (number of operations = 0); in 70B models, performance on the trivial cases is already saturated in the base model.

In Llama 2 7B models, the gains through additional code training are relatively minor, with most of the gains deriving from boosts in examples where the number of operations is either 0 or 1. Similarly minor gains were observed in CodeGemma 8B, except that the gains were observed in examples with 1 and 2 operations. For both of these models, we also observed that for number of operations greater than 0 (Llama 2 7B) and 2 (Gemma 8B), neither the base nor the code variants perform better than our random baseline. These results suggest that there is both a possible effect of scale in the effectiveness of code training as observed in the Llama 2 series, and an effect of the amount of additional code training (DeepSeek-Coder: 2T tokens, Code Llama: 500B tokens).

3.4 Experiment 2: Effect of Math

In evaluating the effect of additional math training, we start by revisiting the claim of Prakash et al. (2024) that the FLoat model obtained by finetuning Llama 7B (Touvron et al., 2023a) on arithmetic tasks from Liu & Low (2023) yields superior entity tracking

²<https://github.com/outlines-dev/outlines>

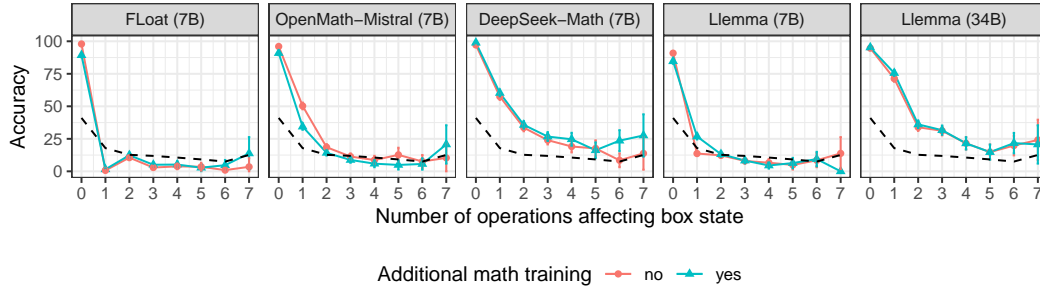


Figure 2: Entity tracking results for models trained with additional math data. See Table 1 for the model names of the base and math models. Error bars indicate 95% confidence intervals, and the black dashed lines show the performance of the random baseline.

performance. As can be seen in Table 3, FLoat did show slightly higher accuracy on the non-trivial tracking cases (number of operations ≥ 1) than the base model, but the gain was marginal.³ Furthermore, neither the base Llama model nor the FLoat model performed better than our random baseline on non-trivial entity tracking examples (Figure 2, far left).

Model	Aggregate	NumOps = 0	NumOps ≥ 1
Llama 7B	28.67	97.93	4.34
FLoat 7B	27.55	89.33	5.85

Table 3: Llama 7B vs. FLoat 7B results.

Following this observation, we compared Mistral and OpenMathMistral models where the latter is a model further trained on OpenMathInstruct-1, a synthetically generated instruction-tuning dataset containing 1.8M unique solutions to math problems sourced from MATH and GSM8K datasets (Toshniwal et al., 2024). As shown in Figure 2, OpenMathMistral only achieved marginal gains over the base Mistral model when there are 7 operations affecting the target box, and in most other cases, the base model consistently outperforms the math-finetuned model. Furthermore, neither model outperformed the random baseline for examples with more than 2 operations affecting the box of interest.

The unclear benefit of additional math training is further corroborated by marginal gains in models trained on math data that are not in “instruct” format like FLoat and OpenMathMistral. Figure 2 shows that DeepSeek-Math (Shao et al., 2024) performed close to the DeepSeek-Coder model for most cases. The gains are even more limited in the comparison between Code Llama vs Llemma (Azerbayev et al., 2024). Llemma 34B outperformed Code Llama 34B by a narrow margin for the non-trivial tracking cases (Llemma: 47.86, Code Llama: 45.46 for examples where the number of operations ≥ 1). These results suggest a limited benefit of additional math pretraining on entity tracking.

3.5 Experiment 3: Effect of Alignment Tuning

Finally, we explore the effect of alignment tuning on entity tracking. For models in the Llama 2 family, alignment tuning the base models led to minor gains (Figure 3, orange vs. green lines in the top row panels), whereas alignment-tuned code models did not consistently lead to gains and sometimes performed worse than the non-alignment-tuned counterparts (orange vs. green lines in the bottom row panels). Nevertheless, the best-performing model was CodeLlama 70B-instruct (64.9 accuracy on 1+ operations), combining code and alignment tuning.

³These numbers are not expected to align with numbers reported in Prakash et al. (2024) because they used a modified version of the original task.

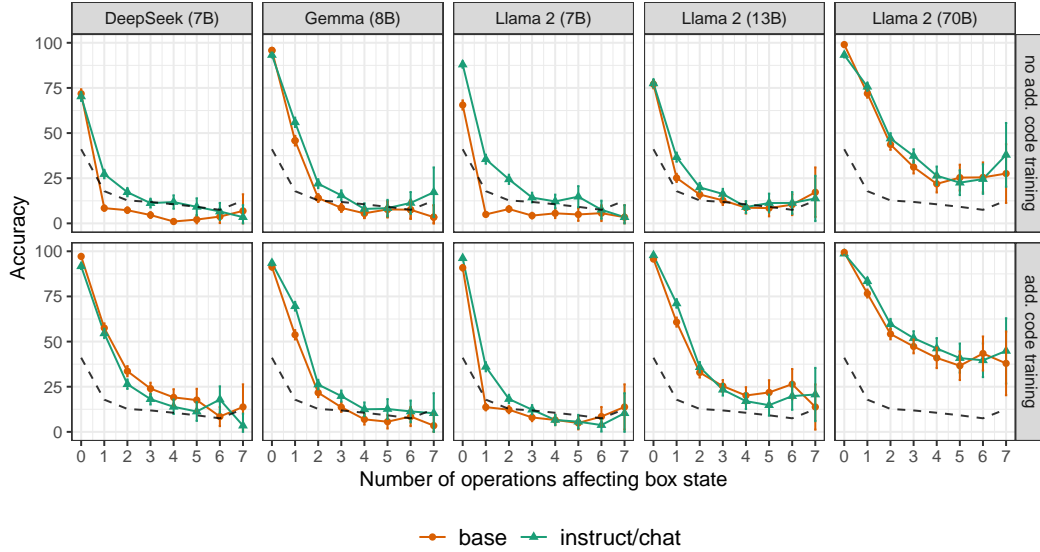


Figure 3: Entity tracking results for alignment-tuned DeepSeek, Gemma and Llama 2 models. The top panels show models without additional code training, whereas the bottom panels show models that have been trained on additional amounts of code before alignment tuning. Error bars indicate 95% confidence intervals, and the black dashed lines show the performance of the random baseline.

The DeepSeek models showed similar trends to the general observation made above: alignment tuning of the base model led to gains, whereas alignment tuning of the code model did not. Gemma 8B and CodeGemma 8B models did benefit from alignment tuning, similarly to Llama 2 7B and CodeLlama 7B models, although the gains were smaller.

Overall, alignment tuning affects base and code models differently, where the gains for base models tend to be greater. The benefit of alignment tuning for base models seems to be inversely correlated with scale: smaller base models benefit more from alignment tuning.

4 Conclusion and Future Work

We explored the effect of code, math, and alignment tuning on LLMs’ capacity to track entities in natural language text. Our main findings are threefold:

1. Additional code training leads to consistent improvements across model families and sizes.
2. Additional math training does not yield consistent improvements, and the performance gains are at best marginal.
3. Alignment tuning leads to different patterns of improvement depending on whether it was applied to base models or code models. Base models consistently benefit from alignment tuning, and smaller models see more improvement. The benefit for code models is more mixed, but the best performance is achieved through combining code and instruction tuning.

Our work thus adds to a growing body of literature that suggests that pretraining on code improves LLM performance on reasoning tasks, including commonsense reasoning (Madaan et al., 2022), chain-of-thought reasoning (Wei et al., 2022b), mathematical problems (Razeghi et al., 2024), and entity tracking tasks (Muennighoff et al., 2023). *Why might this be the case?* Kim & Schuster (2023) argued that keeping track of the states of variables is important for producing correct code, and hypothesized that this kind of procedural input may provide a

stronger training signal than pure natural language text. We consider investigating how code training imbues models with entity tracking and other reasoning abilities an important direction for future research.

Limitations While the pairs of models we compared are “minimal pairs”, several possible confounds remain in our interpretation. For example, we interpreted parts of the results in Experiment 2 as marginal benefit of math compared to code, but the OpenMathInstruct dataset (1.5 GB) is two orders of magnitude smaller in terms of the number of tokens compared to Code Llama’s code data (500 B tokens), so the size of the additional training data could be a confound. The additional math training data of DeepSeek-Math is more comparable (120B tokens), but we do not have a model that is only continually trained on math data; DeepSeek-Math is trained on both code and math. Furthermore, the math data vary along several other important dimensions: OpenMathInstruct, and FLoat models use synthetic data, whereas others use naturally occurring data. FLoat and OpenMathMistral are tuned on math data in instruction format, whereas the training data of DeepSeek-Math and Llemma are not. Unfortunately, we cannot fully tease apart the effect of the format of the math data (instruct vs. non-instruct) here because the format co-varies with whether code was additionally in the training mixture: all models that were additionally trained with non-instruction-formatted math data were continuously trained from models that were trained on code already. Lastly, for the experiments investigating the effect of alignment tuning, we considered models that were alignment tuned through a range of different methods and types of data, and some of the diverging findings in this experiment may be attributed to these differences. We plan to address these existing limitations through controlled training experiments in future investigations.

Acknowledgments

We thank Abdul Rafay for running preliminary experiments as part of his master’s thesis. We also thank Cookie.

References

- Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An Open Language Model for Mathematics. In *ICLR*, 2024.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code. *arXiv:2107.03374*, 2021.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanu-malayan Sankaranarayanan Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon

-
- Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling Language Modeling with Pathways. *JMLR*, 2023.
- Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, and Peter Clark. Everything Happens for a Reason: Discovering the Purpose of Actions in Procedural Text. In *EMNLP-IJCNLP*, 2019.
- DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. DeepSeek LLM: Scaling Open-Source Language Models with Longtermism, 2024.
- Yao Fu, Hao Peng, and Tushar Khot. How does GPT Obtain its Ability? Tracing Emergent Abilities of Language Models to their Sources. *Yao Fu’s Notion*, Dec 2022. URL <https://yaoфу.notion.site/How-does-GPT-Obtain-its-Ability-Tracing-Emergent-Abilities-of-Language-Models-to-their-Sources-b9a57ac0fcf74f30a1ab9e3e36fa1dc1>.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqi, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Gura, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, Hyunjong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayanan Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer,

Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodgkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Vilella, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlias, Arpi Vezzer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin

Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kepa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejas Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrz, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjit Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzdankowski, Fiona Macintosh, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang,

Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnappalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Urias, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christina Koh, Soheil Hassas Yeganeh, Siim Pöder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzasczcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidl, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauer, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Lohrer, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan

Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshv, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesch Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirsenschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. Gemini: A Family of Highly Capable Multimodal Models. *arXiv:2312.11805*, 2024.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikula, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L. Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open Models Based on Gemini Research and Technology. *arXiv:2403.08295*, 2024.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David

-
- Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. OLMo: Accelerating the Science of Language Models. *arXiv:2402.00838*, 2024.
- Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. DeepSeek-Coder: When the Large Language Model Meets Programming – The Rise of Code Intelligence. *arXiv:2401.14196*, 2024.
- Najoung Kim and Sebastian Schuster. Entity Tracking in Language Models. In *ACL*, 2023.
- Tiedong Liu and Bryan Kian Hsiang Low. Goat: Fine-tuned LLaMA Outperforms GPT-4 on Arithmetic Tasks. *arXiv:2305.14201*, 2023.
- Yingwei Ma, Yue Liu, Yue Yu, Yuanliang Zhang, Yu Jiang, Changjian Wang, and Shanshan Li. At Which Training Stage Does Code Data Help LLMs Reasoning? In *ICLR*, 2024.
- Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. Language Models of Code are Few-Shot Commonsense Learners. In *EMNLP*, 2022.
- Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling Data-Constrained Language Models. In *NeurIPS*, 2023.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.
- Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-Tuning Enhances Existing Mechanisms: A Case Study on Entity Tracking. In *ICLR*, 2024.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *NeurIPS*, 2023.
- Yasaman Razeghi, Hamish Ivison, Sameer Singh, and Yanai Elazar. Backtracking Mathematical Reasoning of Language Models to the Pretraining Data. In *The Second Tiny Papers Track at ICLR 2024*, 2024.
- Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code Llama: Open Foundation Models for Code, 2024.
- Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. Neural Theory-of-Mind? On the Limits of Social Intelligence in Large LMs. In *EMNLP*, 2022.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv:2402.03300*, 2024.
- Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. OpenMathInstruct-1: A 1.8 Million Math Instruction Tuning Dataset. *arXiv:2402.10176*, 2024.

-
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv:2302.13971*, 2023a.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv:2307.09288*, 2023b.
- Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned Language Models are Zero-Shot Learners. In *ICLR*, 2022a.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=_VjQlMeSB_J.
- Jason Weston, Antoine Bordes, Sumit Chopra, and Tomás Mikolov. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. In *ICLR*, 2016.
- Brandon T. Willard and Rémi Louf. Efficient Guided Generation for Large Language Models. *ArXiv*, abs/2307.09702, 2023.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-Art Natural Language Processing. In Qun Liu and David Schlangen (eds.), *EMNLP: System Demonstrations*, 2020.
- Ke Yang, Jiateng Liu, John Wu, Chaoqi Yang, Yi R. Fung, Sha Li, Zixuan Huang, Xu Cao, Xingyao Wang, Yiquan Wang, Heng Ji, and Chengxiang Zhai. If LLM Is the Wizard, Then Code Is the Wand: A Survey on How Code Empowers Large Language Models to Serve as Intelligent Agents. *arXiv:2401.00812*, 2024.

Model	Size	Hugging Face Identifier	Chat format
Llama 2	7B	meta-llama/Llama-2-7b-hf	–
	13B	meta-llama/Llama-2-13b-hf	–
	70B	meta-llama/Llama-2-70b-hf	–
Llama 2-Chat	7B	meta-llama/Llama-2-7b-chat-hf	✓
	13B	meta-llama/Llama-2-13b-chat-hf	✓
	70B	meta-llama/Llama-2-70b-chat-hf	✓
Code Llama	7B	codellama/CodeLlama-7b-hf	–
	13B	codellama/CodeLlama-13b-hf	–
	34B	codellama/CodeLlama-34b-hf	–
	70B	codellama/CodeLlama-70b-hf	–
Code Llama-Instruct	7B	codellama/CodeLlama-7b-Instruct-hf	✗
	13B	codellama/CodeLlama-13b-Instruct-hf	✗
	34B	codellama/CodeLlama-34b-Instruct-hf	✗
	70B	codellama/CodeLlama-70b-Instruct-hf	✗
DeepSeek	7B	deepseek-ai/deepseek-llm-7b-base	–
DeepSeek-Chat	7B	deepseek-ai/deepseek-llm-7b-chat	✓
DeepSeek-Coder	7B	deepseek-ai/deepseek-coder-7b-base-v1.5	–
DeepSeek-Coder-Instruct	7B	deepseek-ai/deepseek-coder-7b-instruct-v1.5	✗
DeepSeek-Math	7B	deepseek-ai/deepseek-math-7b-base	–
Gemma	8B	google/gemma-7b	–
Gemma-Instruct	8B	google/gemma-1.1-7b-it	✗
CodeGemma	8B	google/codegemma-7b	–
CodeGemma-Instruct	8B	google/codegemma-1.1-7b-it	✗
Llemma	7B	EleutherAI/llemma_7b	–
	34B	EleutherAI/llemma_34b	–
Llama	7B	huggyllama/llama-7b	–
FLoat	7B	nikhil07prakash/float-7b	–
Mistral	7B	mistralai/Mistral-7B-v0.1	–
OpenMathMistral	7B	nvidia/OpenMath-Mistral-7B-v0.1	–

Table 4: Details of all the models evaluated in the paper. The rightmost column indicates whether the chat format was used for prompting the model.

A Appendix

A.1 Model Details

We used the transformers library (Wolf et al., 2020) by Hugging Face for all our experiments. Table 4 presents all the models along with their Hugging Face identifiers.

For alignment-tuned models, we experimented with prompting the model with and without chat formatting (see Table 5 and Table 2 for the different formats). Based on results over a held-out development set, we selected the best-performing prompt format. We find that except for Llama 2-Chat models and DeepSeek-Chat, all other alignment-tuned models performed better with the non-chat format.

A.2 Constrained Decoding

```
Statement: Box 0 contains( [a-zA-Z]+)*, Box 1 contains( [a-zA-Z]+)*, Box
2 contains( [a-zA-Z]+)*, Box 3 contains( [a-zA-Z]+)*, Box 4 contains(
[a-zA-Z]+)*, Box 5 contains( [a-zA-Z]+)*, Box 6 contains( [a-zA-Z]+)*.
```

Figure 4: Regular expression used for constrained decoding of entity states.

2-shot prompt

<s>[INST] «SYS» Given the description after "Description:", write a true statement about all boxes and their contents to the description after "Statement:". «/SYS»

Description: Box 0 contains the car, Box 1 contains the cross, Box 2 contains the bag and the machine, Box 3 contains the paper and the string, Box 4 contains the bill, Box 5 contains the apple and the cash and the glass, Box 6 contains the bottle and the map. [/INST] Statement: Box 0 contains the car, Box 1 contains the cross, Box 2 contains the bag and the machine, Box 3 contains the paper and the string, Box 4 contains the bill, Box 5 contains the apple and the cash and the glass, Box 6 contains the bottle and the map. </s>

<s>[INST] Description: Box 0 contains the car, Box 1 contains the cross, Box 2 contains the bag and the machine, Box 3 contains the paper and the string, Box 4 contains the bill, Box 5 contains the apple and the cash and the glass, Box 6 contains the bottle and the map. Remove the car from Box 0. Remove the paper and the string from Box 3. Put the plane into Box 0. Move the map from Box 6 to Box 2. Remove the bill from Box 4. Put the coat into Box 3. [/INST] Statement: Box 0 contains the plane, Box 1 contains the cross, Box 2 contains the bag and the machine and the map, Box 3 contains the coat, Box 4 contains nothing, Box 5 contains the apple and the cash and the glass, Box 6 contains the bottle. </s>

<s>[INST] Description: description

Table 5: Chat-formatted prompt with 2-shot in-context demonstrations.

In our experiments, we found that the models struggled to adhere to the output format specified via the few-shot prompt examples. Luckily, the expected output can be described precisely by the regular expression shown in Figure 4. We used the outlines library (Willard & Louf, 2023) which supports regex-based constrained decoding. We found a significant improvement with constrained decoding. For e.g., the performance of the Llama 2 70B model went up from 54.95 to 62.13 with constrained decoding.

A.3 Detailed Results

Table 6 presents the detailed results of all the models evaluated in this work. The results are categorized by the number of operations affecting the entity of interest.

Model	Performance split by number of operations								
	Overall (5012)	0 (1303)	1 (1410)	2 (1083)	3 (651)	4 (288)	5 (142)	6 (106)	7 (29)
Random	21.08	41.06	17.85	12.70	11.87	10.58	9.16	7.51	12.59
Llama 2-7B	21.31	65.54	4.96	7.94	4.30	5.56	4.93	5.66	3.45
Llama 2-7B Chat	41.28	87.95	35.53	24.38	14.29	12.15	14.79	7.55	3.45
Llama 2-13B	33.28	77.05	25.18	15.97	12.75	8.68	8.45	10.38	17.24
Llama 2-13B Chat	38.05	77.51	36.67	19.94	16.28	9.03	11.27	11.32	13.79
Llama-2 70B	62.13	99.00	71.91	43.67	31.18	21.88	25.35	25.47	27.59
Llama-2 70B Chat	63.43	93.25	75.67	47.00	37.33	26.39	22.54	24.53	37.93
Code Llama 7B	31.94	90.87	13.69	12.28	7.99	6.60	4.93	8.49	13.79
Code Llama 7B Instruct	41.34	96.16	36.03	18.19	12.29	6.60	5.63	3.77	10.34
Code Llama 13B	54.79	95.70	60.78	32.87	25.35	20.14	21.83	26.42	13.79
Code Llama 13B Instruct	58.14	97.77	71.13	35.83	23.35	17.01	14.79	19.81	20.69
Code Llama 34B	58.26	94.70	71.28	33.89	31.18	21.53	14.79	19.81	24.14
Code Llama 34B Instruct	61.47	95.09	77.09	39.98	31.03	22.57	19.01	20.75	20.69
Code Llama 70B	69.77	99.39	76.60	54.20	47.31	40.97	36.62	43.40	37.93
Code Llama 70B Instruct	73.66	98.70	83.40	59.65	51.92	46.18	40.85	39.62	44.83
Llemma 7B	34.12	84.65	26.81	13.20	8.29	4.51	6.34	9.43	0.00
Llemma 34B	60.18	95.24	75.60	36.10	31.64	21.53	14.79	21.70	20.69
DeepSeek 7B	23.46	71.83	8.44	7.29	4.61	1.04	2.11	3.77	6.90
DeepSeek 7B Chat	32.24	70.30	27.30	17.27	11.21	11.81	9.15	6.60	3.45
DeepSeek 7B Coder Base	53.67	97.16	57.52	33.61	23.96	19.10	17.61	8.49	13.79
DeepSeek 7B Coder Instruct	48.76	91.71	54.54	26.41	18.13	13.89	11.27	17.92	3.45
DeepSeek 7B Math Base	56.40	98.93	60.28	35.73	26.73	24.65	16.20	23.58	27.59
Gemma-7B	42.66	95.78	45.82	14.04	8.60	5.56	7.75	7.55	3.45
Gemma-7B Instruct	47.79	93.25	55.96	21.98	15.51	7.99	8.45	11.32	17.24
CodeGemma-7B	46.09	91.33	53.76	21.70	13.67	6.94	5.63	8.49	3.45
CodeGemma-7B Instruct	53.49	93.48	69.57	26.22	19.82	12.50	12.68	11.32	10.34
Mistral 7B	45.67	96.01	50.28	18.56	11.37	8.68	12.68	7.55	10.34
OpenMathMistral	38.25	90.94	34.40	14.22	8.76	5.90	4.93	5.66	20.69
LLama 7B	28.67	97.93	0.71	10.53	2.92	3.82	3.52	0.94	3.45
Float 7B Instruct	27.55	89.33	1.56	12.37	5.07	5.21	2.82	4.72	13.79

Table 6: Entity Tracking performance of models categorized by the number of operations affecting the entity of interest. The count of test set instances with the number of operations affecting an entity is indicated in parentheses below the corresponding column title.