Improving Math Problem Solving with Long Context LLMs

Rajeev Veeraraghavan, Anish Kiran Kulkarni, Ajay Mittur
rveerara@andrew.cmu.edu, anishkik@andrew.cmu.edu, amittur@andrew.cmu.edu
10-423/623 Generative AI Course Project

November 26, 2024

1 Introduction

Recent advancements in large language models (LLMs) have significantly increased their context sizes, enabling more extensive in-context learning capabilities. This development offers a unique opportunity to enhance performance on complex tasks by utilizing many-shot prompts, where a greater number of examples can be incorporated into the context. Motivated by recent research Agarwal et al. [2024], which demonstrates the benefits of many-shot in-context learning for solving mathematical problems, we aim to explore and extend these findings to open-source LLMs, particularly focusing on their performance on challenging datasets like MATH Hendrycks et al. [2021] and GSM8K Cobbe et al. [2021a].

Our study seeks to verify whether the performance gains reported for proprietary models can be replicated and generalized to open-source LLMs. Specifically, we evaluate the effectiveness of many-shot prompting in both supervised and unsupervised settings, comparing its performance against traditional few-shot approaches. Additionally, we investigate the potential of using synthetic data to augment many-shot prompts and assess its impact on model accuracy.

To achieve these goals, we reimplement the evaluation methodology used in prior work and conduct experiments using the Llama family of models, which support large context windows up to 128k tokens. We hypothesize that many-shot prompting will yield significant accuracy improvements over few-shot prompting on the MATH dataset and that the benefits will partially transfer to the GSM8K dataset, despite differences in task distributions. Furthermore, we expect to observe that unsupervised many-shot prompting, which excludes answers in the context, can still achieve notable performance gains.

Our approach leverages efficient methods like prefix prompt caching and structured output frameworks to minimize computational overhead and streamline the evaluation process. By systematically varying the number and type of examples in the prompts, we aim to identify optimal configurations for maximizing accuracy. Preliminary results suggest trends consistent with prior findings, indicating that longer context sizes and better example selection are key drivers of performance. Ultimately, our research contributes to a deeper understanding of how to utilize extended context lengths in LLMs to enhance task-specific outcomes effectively.

2 Dataset & Task

We will use the MATH Cobbe et al. [2021b] and GSM8K Cobbe et al. [2021a] datasets to evaluate whether many-shot prompting improves performance over few-shot prompting. We measure accuracy with exact match of LLM generated answers to ground truth solutions. The answers may be produced in latex, so, we also parse these before performing an exact math. To be specific, we use the evaluation function used in the Minerva paper Lewkowycz et al. [2022]. The source code is linked here. This is also the closest evaluation function we found to what Meta used while evaluating Llama 3.1 Dubey et al. [2024].

2.1 MATH

The MATH dataset Hendrycks et al. [2021] contains 12,500 high school-level math problems from competitive math events. Each problem requires machine learning models to produce a final answer, such as $\frac{2}{3}$, in a normalized form. This normalization makes answers unique, allowing exact match metrics instead of heuristic metrics. Problems in the dataset are categorized by difficulty on a scale of 1 to 5 and span seven mathematical domains like number theory, algebra, and probability. The main goal of the dataset is to evaluate a model's ability to tackle complex mathematical problems and provide concise solutions.

Further, we partition the MATH dataset into two a smaller subset that has answers which are fully numeric - integers or floats. We do this for 2 reasons. First, it allows us to evaluate results on this subset

without parsing latex outputs or using non exact match metrics. Second, we can evaluate whether few and many-shot prompting with this subset of "numerical answer" questions improves performance on problems in GSM8K over selecting example shots from the entire dataset. This is because the questions in both MATH and GSM8K come from different distributions - and using the numerical subset - sans latex - allows us to evaluate out-of-data distrbution performance on a fine grained level. Note that this is something we only do for this milestone and will use latex parsing and perform more comprehensive evaluation for the final milestone as described earlier.

2.2 GSM8K

GSM8K Cobbe et al. [2021a] consists of 8500 high quality grade school math problems created by human problem writers. The dataset is split into 7500 training problems and 1000 test problems. Each problem takes between 2 and 8 steps to solve. The human written solutions are in natural sentence form and perform a sequence of basic math operations in each step. The final step contains the answer. Unlike the MATH dataset, the problems are not classified by topics. However, the answers are exact. The exact match accuracy metric described earlier is employed to evaluate performance on this dataset as well.

3 Related Work

3.1 Using the long context for many shot in context learning

In many shot in context learning Agarwal et al. [2024] the authors analyze the benefit of the long context available in Gemini 1.5 Pro LLM. They explore two settings - reinforced in context learning and unsupervised learning. In reinforced in context learning, the authors model generated chain of thoughts in examples provided in the prompt and in unsupervised in context learning they remove reasoning and only include domain specific examples in the prompt. The authors observe and report improvement in performance for various tasks using many shot in context learning. Improvements in accuracy on GSM8K Cobbe et al. [2021a] are from 84% to 93% and MATH Hendrycks et al. [2021] from 50% to 58.1%.

3.2 Analyzing the benefit of longer context overall

In In-Context Learning with Long-Context Models: An In-Depth Exploration Bertsch et al. [2024] the authors analyze in context learning for LLMs with larger context size. They identify that performance continues to increase with a large number of examples in the prompt. They identify that example retrieval provides diminishing returns as the length of the context increases. They also find long context in context learning less sensitive to the order of examples as compared to short context. They conclude that overall understanding of in context learning especially in view of the increasing context size remains incomplete and requires further work for more hypothesis validation.

3.3 Analyzing RAG performance using long context (1)

In Long-Context LLMs Meet RAG: Overcoming Challenges for Long Inputs in RAG Jin et al. [2024] the authors evaluate the impact and benefits of long context on performance of retrieval augmented generation. The authors observe that having more data in the prompt improves quality initially but it reduces as the as the retrieval size keeps increasing. The authors observe that the irrelevant passage negatively affects retrieval augmented generation performance and proppose optimization approaches with and without training to improve performance.

3.4 Analyzing RAG performance using long context (2)

In Long Context RAG Performance of Large Language Models Leng et al. [2024], the authors identify the effect of longer context on 20 major LLMs. They observe that retrieving more documents can improve performance but the improvement is generally lost as the context increases above 64K tokens.

3.5 Analyzing impact of information location in input for long context

In Lost in the Middle: How Language Models Use Long Contexts Liu et al. [2023] the authors try to analyze how LLMs tend to use the longer context typically availble with recent LLMs. The authors analyze performance using two tasks - multi document question answering and key value retrieval. They observe that the performance on the task is very dependent on the location of the relevant information. The observe that performance is highert when relevant information is at the beginning of the input and performance reduces as the relevant information is further towards the middle of the input. The performance again improves as the relevant information is towards the end of the input.

3.6 Does long context help in-context learning?

In Long-context LLMs Struggle with Long In-context Learning Li et al. [2024], the authors introduce a new benchmark for in context learning called Long-ICLBench to analyze the benefit of long context to in context learning for LLMs. They observe that the longer context helps with easy tasks but on difficult tasks almost all LLMs fail. They also observe that the LLMs tend to be more attentive to examples towards the end of the prompt.

3.7 Is long context necessary?

In Are Long-LLMs A Necessity For Long-Context Tasks? Qian et al. [2024], the authors provide a framework called LC Boost for using short context LLMs to solve long context problems. Based on the results observed wit small context models using this framework, the authors argue that long context is not necessary and short context LLMs can also solve a lot of long context problems using this framework.

3.8 Does in context learning help with instruction following?

In Is in-context learning sufficient for instruction following in LLMs? Zhao et al. [2024], the authors try to compare the performance of in context learning with instruction finetuning and try to analyze if in context learning can help an LLM in alignment. The authors provide an insight that providing high quality examples in the context can help the model performance increase towards that of an instruction fine tuned model.

4 Approach

We re-implement from scratch the work presented in Agarwal et al. [2024] - where supervised and unsupervised many-shot prompting improved performance over supervised few-shot in-context learning on MATH and GSM8K datasets in Gemini LLM models. Instead of Gemini, we evaluate the effect of many shot prompting on open source as well as small language models - especially the LLama3 family of LLMs. In Agarwal et al. [2024], the many-shot prompts' tokens' lengths were of the order of 100k. This means that we can try to replicate their results on models with maximum context window lengths of 128k tokens, which Llama supports. The baseline results of Gemini on MATH are in table 1.

4.1 Baseline Approach

We re-evaluate baselines on our own and compare if they match reported results. Since we are measuring improvement over zero shot and few shot ICL on MATH, our baseline evaluation is also with zero-shot and few-shot ICL on MATH test dataset. For few-shot prompting, we include 2-5 examples with both questions and answers in the prompt. We evaluate these baselines using exact match accuracy metrics on both the full MATH test dataset and the numerical subset that we derived.

4.2 Main Methods

We will implement the manyshot methods proposed in Agarwal et al. [2024] and include some of our own variations.

First, we will evaluate whether unsupervised many-shot prompting - where the prompt includes up to 500 questions, but no answers - outperforms regular few-shot prompting with both questions and answers. We will then evaluate whether synthetically generated answers improve accuracy, irrespective of their quality.

The results that we plan to replicate are the consistent improvements over few-shot prompting achieved by both supervised (re-inforced) and unsupervised many-shot prompting (for an average of a 7.9% improvement) on MATH. Even if our average improvement is not the same, we expect to see a similar trend where many-shot prompting (unsupervised and re-inforced) beats regular in-context-learning over a range of a number of many-shot prompts. As an extension, we also plan to verify the finding that the many-shot prompts obtained from MATH transfer to GSM8K and improve performance slightly even though GSM8K's question distribution is different.

For efficiency, we will use existing frameworks (such as DsPY Khattab et al. [2023]) to obtain structured outputs and abstract away the parsing of generated text. We also take advantage of VLLM **?** which implement prefix prompt caching to save computation and cost when adding many-shot examples. To ensure efficient prefix prompt caching, we will only add new examples (shots) as we increase the number of shots in our many-shot ablations.

5 Experiments

Our goal is to improve performance on MATH using many shot in-context examples with Llama 3.1. Hence, we evaluate the performance of Llama 3.1 8B Instruct Dubey et al. [2024] with zero shot chain of thought prompting on the MATH dataset as our baseline. We

Model	Prompt	Dataset	Subset	Accuracy (%)
Llama 3.1 8B Instruct Dubey et al. [2024]	Zero Shot CoT	MATH	All	51.9%
Llama 3.1 8B Instruct (Our Eval)	Zero Shot CoT	MATH	All	47.6%
Llama 3.1 8B Instruct (Our Eval)	Zero Shot CoT	MATH	Numeric	44.46%
Gemini Ultra Agarwal et al. [2024]	250 Shot CoT	MATH	MATH500	58.8%

Table 1: Baseline results

Model	Prompt	Source	Type	Dataset	Subset	Accuracy (%)
Llama 3.1 8B Instruct	3 Shot CoT	Synthetic	Supervised	MATH	All	31.8%
Llama 3.1 8B Instruct	5 Shot CoT	MATH Test	Unsupervised	MATH	Numeric	35.19%

Table 2: Initial experiment resutls

Model	Prompt	Source	Type	Dataset	Subset	Accuracy (%)
Llama 3.1 8B Instruct	-	Synthetic/RAG	(Un)Supervised	-	-	-

Table 3: Skeleton table for Many shot experiments with synthetic filtered and unfiltered data

also evaluate the same model on the subset of MATH dataset which have numeric answers. These results are provided in Table 1.

It is evident from the Table 1 that we were unable to get the exact same accuracy as Meta. This can be attributed to a few reasons. First, Meta did not disclose the exact prompt they used to evaluate their model. Second, Meta used an LLM as a judge on top of exact match to determine the accuracy. We did not do this because of compute constraints.

We further evaluate the effect of adding examples in to the model in a few shot chain of thought prompting setup. We experiment with the amount, type and source of examples. We also experiment about how the performance of the model change depending on whether the in context learning examples are provided with corresponding answers or not i.e. supervised and unsupervised in context learning. Result of these experiments are provided in Table 2.

The initial experiments were done in bare minimum settings with minimal prompt engineering and data filtering, leading to results that are not competitive yet. However, moving forward, we will continue experimenting along these areas as described. Another angle that we will craft experiments around are whether the quality of synthetic examples created given a problem are important. The results of which will be filled in the table 3. We will experiment with different number of N-Shots, filtering and not filtering the generated synthetic data, and using supervised and unsupervised ICL prompts on different subsets (classes of problems) of the MATH dataset.

6 Plan

Going ahead, we would like to experiment about how many shot in context learning affects the performance of the model on solving math examples. We would like to identify if the performance gains described in Agarwal et al. [2024] also work for smaller and open source models. Further, we will work on improving the synthetic data generation and scale it to many shots at test time. At the same time we are also working on the reasoning and prediction pipeline on the dataset in parallel. So, we are on track to reach our next big milestone of a complete pipeline and experiments in the next few weeks.

Our weekly plan and responsibilities until the final presentation is as follows:

- Nov 25 - Nov 29: Improve Synthetic Generation of Math Problems (Ajay). Concurrently, setup and organize experiment logging and plotting (Rajeev, Anish).
- Nov 30 - Dec 7: Run experiments and across the settings described in section 5 (Ajay, Rajeev, Anish). Perform synthetic data filtering for the filtered data experiment (Ajay, Anish).
- Dec 7 - Dec 11: Collect results, plot graphs and conduct a systematic analysis (Ajay, Rajeev, Anish). Refactor Code (Rajeev).
- Dec 11: Submit Poster and Report (All of us).

References

Rishabh Agarwal, Avi Singh, Lei M. Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang,

Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. Many-shot in-context learning, 2024. URL https://arxiv.org/abs/2404.11018.

Amanda Bertsch, Maor Ivgi, Uri Alon, Jonathan Berant, Matthew R. Gormley, and Graham Neubig. In-context learning with long-context models: An in-depth exploration, 2024. URL https://arxiv.org/abs/2405.00200.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021a.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021b. URL https://arxiv.org/abs/2110.14168.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi,

Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O. Arik. Long-context llms meet rag: Overcoming challenges for long inputs in rag, 2024. URL https://arxiv.org/abs/2410.05983.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. Dspy: Compiling declarative language model calls into self-improving pipelines, 2023. URL https://arxiv.org/abs/2310.03714.

Quinn Leng, Jacob Portes, Sam Havens, Matei Zaharia, and Michael Carbin. Long context rag performance

of large language models, 2024. URL https://arxiv.org/abs/2411.03538.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning, 2024. URL https://arxiv.org/abs/2404.02060.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023. URL https://arxiv.org/abs/2307.03172.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Yujia Zhou, Xu Chen, and Zhicheng Dou. Are long-llms a necessity for long-context tasks?, 2024. URL https://arxiv.org/abs/2405.15318.

Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Is in-context learning sufficient for instruction following in llms?, 2024. URL https://arxiv.org/abs/2405.19874.