# Uncovering Latent Chain of Thought Vectors in Language Models

**Anonymous ACL submission**

## Abstract

Being able to reliably control the behaviors of Language Models (LMs) is becoming an increasingly pressing issue as they grow more influential and trusted. For this, we investigate the technique of steering vectors: an inference-time intervention on LMs using a 'steering vector' derived from a specific task. We apply them to steer LMs toward performing Chain of Thought (CoT) Reasoning without the need to prompt through natural language. We demonstrate this approach on Llama3 8b Instruct and Mistral 7b v0.2 Instruct, and obtain competitive, if not superior, qualitative and quantitative results relative to CoT-prompted performances on a series of reasoning benchmarks (GSM8k, MMLU, AGI Eval, ARC AI2). We find that this approach consistently steers responses toward CoT reasoning, while requiring less compute and memory compared to traditional methods like model finetuning.

## 1 Introduction

The integration of LMs into various aspects of society has accelerated rapidly in recent years (Zhao et al., 2023). However, these models still harbor significant limitations, particularly in generating accurate information and solving complex reasoning problems (Huang et al., 2023; Mondorf and Plank, 2024). In response, considerable work has been done in making LMs more precise. One effective approach highlighted in the literature involves prompting or steering models toward CoT Reasoning. This method enables LMs to decompose complex problems into smaller, more manageable sub-problems and detect errors made in the problem-solving process (Wei et al., 2022; Nye et al., 2022).

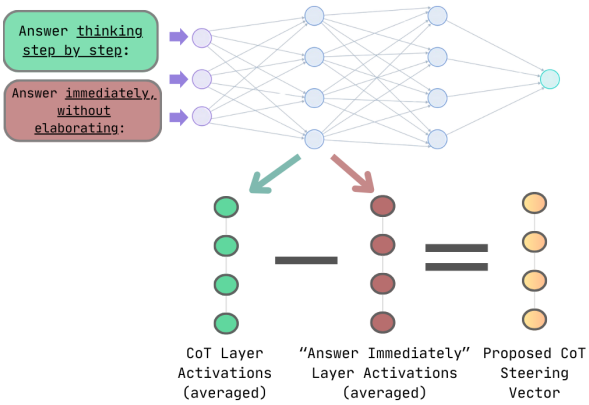


Figure 1: Overview of Steering Vector creation workflow. We use two contrasting natural language prompts alongside questions from reasoning benchmarks to extract "Think Step by Step" and "Answer Immediately" layer residual stream activations from the LM. We then average-pool these activations across token position and across a corpus of activations from other reasoning questions and subtract the two vectors to get a proposed "CoT Steering Vector" to inject for steered model generation.

Current prominent approaches to steering models towards CoT reasoning include supervised fine-tuning (Devlin et al., 2019), reinforcement learning from human feedback (Ziegler et al., 2020), and prompt engineering (Radford et al., 2019). However, these methods can be limited in effectiveness and require substantial compute and memory (Casper et al., 2023; Chen et al., 2024). By building upon existing methods of using steering vectors to perturb the latent activation space of LMs, this work introduces an approach of steering LMs towards CoT that optimizes for computational resources as well as reliability.

We outline the process of creating steering vectors and evaluate the effectiveness of steering on a variety of reasoning benchmarks (GSM8k, MMLU, ARC-AGI Reasoning Challenge, AGI Eval) (Cobbe et al., 2021; Hendrycks et al., 2021; Chollet, 2019; Zhong et al., 2023). Our experiments, conducted on two LMs (Llama3 8b Instruct

and Mistral 7b v0.2 Instruct) (Dubey et al., 2024; Jiang et al., 2023), demonstrate that this approach is not only effective, but also competitive with regards to performance on reasoning benchmarks when compared to models using natural language prompting for CoT.

## 2 Related Works

The advancement of LMs necessitate precise control over their responses. As such, the field of LM steering is being explored extensively. We delve into current prominent methodologies being employed to steer LM generated responses.

### 2.1 Model Fine-tuning

Fine-tuning models remains a cornerstone in adapting LMs to specific tasks or preferences. These approaches involve using labeled data to adapt parameters of pre-trained LMs towards preferred outputs. Commonly used practices for altering model parameters normally include Supervised Fine-tuning (Devlin et al., 2019), Reinforcement Learning from Human Feedback (Ziegler et al., 2020) , and Direct Preference Optimization (Rafailov et al., 2024). However, these methods often require substantial computational resources and intricate tuning to ensure effectiveness, limiting their scalability and practicality in deployment (Casper et al., 2023).

### 2.2 Prompt Engineering

Prompt Engineering seeks to harness the intrinsic capabilities of LMs by crafting optimized prompts that guide models towards desired reasoning paths and outputs. Key techniques encompassed by this field include Few-Shot Prompting, which provides example solutions to guide the model (Brown et al., 2020), and Automated Prompt Generation, which uses LMs themselves to refine user queries (Zhou et al., 2023). Another effective technique in the field of Prompt Engineering includes CoT prompting, which encourages models to explain their thought process by thinking step-by-step, improving performance on complex reasoning tasks (Wei et al., 2023). In this paper, we focus steering the model towards CoT formatted responses.

### 2.3 Logit Filtering

Logit filtering involves manipulating the output distribution of token or sequence predictions in LMs during model generation. Relevant work that has been done includes prefix-tuning, which influences output distributions of logits by attaching a learned "prefix" matrix to input embeddings in order to influence model behaviors (Li and Liang, 2021).

### 2.4 Activation Engineering

Recent work has been done suggesting that the internal activations of LMs carry representational meaning. (Azaria and Mitchell, 2023). This promise has resulted in the creation of the field of Activation Engineering, in which researchers have sought to perturb the model's activation space to steer model behavior. Applications of this approach for model steering first originated with Plug-and-Play LMs as outlined by (Dathathri et al., 2020), in which a classifier was trained and used to steer layer activations closer towards desired activation structures that successfully reduced toxicity and strengthened sentiment control in model responses.

This approach was later followed by Subramani et al. (2022), who first formally used a learned "steering vector" to alter the activation space of a frozen LM. This approach yielded strong control over model generation, achieving near-perfect model completion for certain sentences. Similar work was later done by Turner et al. (2023) and Liu et al. (2024), who used natural language prompting instead of Subramani et al. (2022)'s gradient learned approach to extract latent steering vectors for similar goals of sentiment control and toxicity reduction.

Other work in the field of Activation Engineering carries more of a flair towards interpretability (Michel et al., 2019; Olsson et al., 2022). Notable work includes the use of Activation Engineering as a probe for truthfulness pioneered by Marks and Tegmark (2024), introducing a new technique of average-pooling layer activations in the creation of a more accurate probe.

## 3 Methodology & Experiment Set-Up

### 3.1 Steering Vector Creation

To obtain a steering vector from a LM, we extract layer activations from model forward passes over a corpus of logic and reasoning questions drawn from various reasoning benchmarks alongside contrasting natural language prompts (e.g., "Answer the following question thinking step by step..." versus "Answer the following question immediately, without any elaboration..."). Using extracted layer activations from our specified LM, we then average

pool them across token position and then across all 300 questions, thus obtaining a "Think Step by Step" and "Answer Immediately" vector. To obtain our final steering vector, we subtract the two vectors. More information about specific prompts used can be found in Appendix A.3.

In an effort to isolate as much as possible for CoT in the steering vector and avoid overfitting steering vectors to a specific topic, question type, or level of difficulty, we use a broad and roughly even split of questions from 3 different benchmarks encompassing a multitude of different topics. Below is the corpus of training data used for this paper's steering vector (297 questions total):

1. Big Bench Lite (Srivastava et al., 2023): 3 questions from each category (72 total)

2. MMLU: 2 questions from each category in dev split (125 total)

3. GSM8k: 100 questions from training dataset

### 3.2 Steering Vector Injection

To utilize the steering vector to steer the model at inference time, we add a PyTorch (Paszke et al., 2019) hook at the specified layer the vector was extracted from, and add the vector in after scaling it alongside an injection coefficient (both the specified layer and injection coefficient are found through grid search).

We experiment with injecting the steering vector in 2 different token positions: one being a single injection of the steering vector with a high injection coefficient after the end of the user input, and the other being a continuous injection of the steering vector with a much lower injection coefficient after each new token generated by the model.

## 4 Results

Our findings, shown in Table 1, demonstrate that steering vector injected systems perform competitively against CoT-prompted systems for both models examined. This approach underscores the potential of steering vectors as an effective method for guiding LMs toward Chain-of-Thought reasoning without compromising overall model performance. Further, qualitative examples (as seen in Figure 2 and Figure 3) illustrate the practical application of steering vector systems compared to baseline model responses for both Llama3 8b Instruct and Mistral 7b v0.2 Instruct.

**Question (GSM8k)**

A trader buys some bags of wheat from a farmer at a rate of $20 per bag. If it costs $2 to transport each bag from the farm to the warehouse, and the trader made a total profit of $400 after selling all the bags at a rate of $30 each, how many bags did he sell?

**Baseline Output (Llama3 8b Instruct)**

The trader bought the bags at 20 dollars each. Then he sold them at 30 dollars each. So he made 30 - 20 = 10 dollars per bag. He made a total profit of 400 dollars. So he sold 400 / 10 = 40 bags. The answer is 40.

**Steered Output (Llama3 8b Instruct)**

(1) Let's calculate the total cost of buying the bags of wheat. The cost of buying each bag is $20, and the total cost is 20x, where x is the number of bags. The cost of transporting each bag is $2, and the total cost of transportation is 2x. So the total cost of buying and transporting the bags is 20x + 2x = 22x.
(2) The trader sells each bag at a rate of $30, and the total revenue is 30x.
(3) The trader made a total profit of $400, so the total revenue is equal to the total cost plus the profit. 30x = 22x + 400.
(4) Subtract 22x from both sides to get 8x = 400.
(5) Divide both sides by 8 to get x = 50.
The trader sold 50 bags of wheat.

Figure 2: Qualitative Example of Steered vs. Unsteered Response for GSM8k

We also find that different steering vector injection positions yield better results on different models. Llama3 8b Instruct performed substantially better on benchmark evaluations with a single steering vector injection at the end of the input prompt alongside a high injection coefficient ($c = 20$) compared to steering vector injection at each new token generation step, while Mistral 7b v0.2 Instruct performed far better steering vector injection alongside a low injection coefficient ($c = 1$) at each step.

Table 1: Experimental Results

| Model | GSM8K | MMLU | ARC AI2-C | AGI Eval[1] |
|---|---|---|---|---|
| Llama3 8b Instruct: CoT[2] | 73.90 | **65.60** | 80.46 | 59.09 |
| Llama3 8b Instruct: Steering Vector | **79.15** | 64.20 | **81.23** | **61.40** |
| Mistral 7b v0.2 Instruct: CoT | **50.72** | 48.95 | 60.75 | 40.00 |
| Mistral 7b v0.2 Instruct: Steering Vector | 48.20 | **52.30** | **62.70** | **42.30** |

[1] Used Only SAT Math Section
[2] Zero shot prompting used for both steered and CoT prompted systems. For exact prompting details, please see Appendix A.3.
Note: Due to compute constraints, the results reported are from a single experimental run for each model and task.

We note that for both models, injecting a steering vector with a low injection coefficient yields strong steering towards CoT structured responses, but subsequent performance on reasoning benchmarks isn't guaranteed, as seen with Llama3 8b Instruct. More information about hyper-parameter search for optimal $l$ and $c$ values can be found in Appendix B.

## 5 Conclusion & Future Work

Our findings demonstrate the efficacy of steering vector as a potential method for steering LM responses towards CoT reasoning. Our experimental results show that models steered with our vector system not only achieve competitive performance to CoT reasoning benchmarks, but in several cases, surpass models prompted with conventional natural language CoT. This suggests that steering vectors could potentially serve as an efficient and intuitive tool for enhancing reasoning capabilities of LMs.

Future research on the application of steering vectors for reasoning tasks should focus on enhancing the steering vector workflow in several key areas. First, diversifying and expanding the training datasets used for steering vector creation can help to reduce noise. Second, applying steering vector solutions to a broader range of models, including base models, should be explored. Finally, investigating the integration of steering vector solutions with specialized prompting techniques could further advance their effectiveness.

**Question (GSM8k)**

There are four schools competing at a basketball tournament. Each school has sent a girls' basketball team and a boys' basketball team and each team has 5 players each. Each school has also sent a coach for each team. In total, how many people have all of the schools sent?

**Steered Output (Mistral 7b v0.2 Instruct)**

Each school has sent the following number of people to the basketball tournament:
1. Girls' basketball team: 5 players + 1 coach = 6 people
2. Boys' basketball team: 5 players + 1 coach = 6 people
Since there are 4 schools, the total number of people sent to the tournament is:
Total number of people = Number of schools * (Number of people in a girls' team + Number of people in a boys' team)
Total number of people = 4 * (6 + 6)
Total number of people = 4 * 12
Total number of people = 48
So, all of the schools have sent a total of 48 people to the basketball tournament.

**Baseline Output (Mistral 7b v0.2 Instruct)**

Each school has sent the following number of people:
1. Girls' basketball team: 5 players + 1 coach = 6 people
2. Boys' basketball team: 5 players + 1 coach = 6 people
Since there are 4 schools, the total number of people sent is:
4 schools * (6 people/school) = 4*6=24 people



Figure 3: Qualitative Example - Steered vs. Baseline Response on GSM8k for Mistral 7b v0.2 Instruct

## 6 Limitations

The limitations of this work primarily stem into two main parts: (1) our evaluation of the steering vector does not rigorously isolate for CoT reasoning, and as such could still include much noise from other concepts; (2) CoT steered systems can at times output illogical reasoning steps.

Regarding (1), concerns may arise with regards to whether our vector is a faithful representation of CoT reasoning within the model. While steering vector solutions will likely always have a source of noise from the training data it was derived from, potential work can be done to reduce this noise by increasing the scale and range of the training data used to train the steering vector. Another path forward to isolating for CoT in LMs could be to investigate causal ablation techniques with steering vector information to identify potential CoT related causal structures in LMs (Wu et al., 2023).

Regarding (2), we note in some test examples that CoT steered systems can generate incorrect or irrelevant CoT steps in the problem solving process. In order to meaningfully address this concern and ensure it isn't due to the size of the LMs we test on, work can be done on evaluating this system on larger systems, as well as in designing systems (potentially a smarter AI agent) to evaluate each step in the reasoning process of steered vs. CoT promped systems to see if any meaningful differential exists in the quality of CoT in steered vs. prompted systems. Nonetheless, the competitive test scores we're able to achieve on benchmarks in comparison to CoT performance give us confidence that our approach holds promise in its ability to capture even complex representations like reasoning.

## 7 Ethics & Risks Statement

Our work on steering vectors for CoT in LMs is an important advancement in the field of LMs to show that this structure is representationally encoded in LM activation layers. However, considerable risk and ethical implications still should be addressed. This study takes more of a steering rather than interpretability perspective on steering vectors, and as a result, considerable noise could also be encoded in steering vectors. It is important for robustness work to be done to understand potential biases, hallucinations, or erroneous behaviors that might be encoded in these vectors.

## References


Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying. *Preprint*, arXiv:2304.13734.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Stephen Casper, Xander Davies, Claudia Shi, T. Gilbert, J'er'emy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, P. Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Ségerie, Micah Carroll, Andi Peng, P. Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, J. Pfau, Dmitrii Krasheninnikov, Xin Chen, L. Langosco, Peter Hase, Erdem Biyik, A. Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *ArXiv*, abs/2307.15217.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2024. Longlora: Efficient fine-tuning of long-context large language models. *ICLR*.

François Chollet. 2019. On the measure of intelligence. *Preprint*, arXiv:1911.01547.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

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. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. *Preprint*, arXiv:1912.02164.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

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. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *Preprint*, arXiv:2311.05232.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597, Online. Association for Computational Linguistics.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2024. In-context vectors: Making in context learning more effective and controllable through latent space steering. *Preprint*, arXiv:2311.06668.

Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *Preprint*, arXiv:2310.06824.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Preprint*, arXiv:1905.10650.

Philipp Mondorf and Barbara Plank. 2024. Beyond accuracy: Evaluating the reasoning behavior of large language models – a survey. *Preprint*, arXiv:2404.01869.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2022. Show your work: Scratchpads for intermediate computation with language models.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Preprint*, arXiv:2209.11895.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power,

Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo

Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Preprint*, arXiv:2206.04615.

Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. 2022. Extracting latent steering vectors from pretrained language models. *ArXiv*, abs/2205.05124.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Zhengxuan Wu, Atticus Geiger, Christopher Potts, and Noah D. Goodman. 2023. Interpretability at scale: Identifying causal mechanisms in alpaca. *ArXiv*, abs/2305.08809.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Ji rong Wen. 2023. A survey of large language models. *ArXiv*, abs/2303.18223.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *Preprint*, arXiv:2304.06364.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. *Preprint*, arXiv:2211.01910.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *Preprint*, arXiv:1909.08593.

# A Dataset, Model, Prompting Details

## A.1 Dataset Information

***GSM8k*** (Cobbe et al., 2021): The GSM8k dataset contains 8,500 linguistically diverse, multi-step grade school math word problems, covering topics up to early algebra levels. The training dataset consists of 7,500 entries and the testing dataset consists of 1,000 entries. We train our steering vector on 100 examples from the training dataset, from entries 4,500 to 4,600. We then evaluate our steered system on the entire test dataset.

***MMLU*** (Hendrycks et al., 2021): The MMLU dataset comprises of a series of approximately 16,000 multiple choice questions to test the knowledge and reasoning capabilities of LLMs. It is composed of a test set of 14,079 questions, a validation set of 1,540 questions, and roughly 285 questions from the "dev" split (a split of sample questions used for few-shot prompting). It contains questions from a variety of subjects, from mathematics and physics to law and computer science. It is split roughly into 57 subjects across various fields, with difficulty levels varying from elementary to professional level questions.

For steering vector training, we use the the first 2 questions from the "dev" split of the MMLU questions to sample a few questions from a broad range of topics. This totals out to around 125 questions used. For evaluation, we use a random seed and shuffle the "test" dataset and take the first 1,000 questions for testing.

***BIG-Bench*** (Srivastava et al., 2023): The BIG-Bench dataset contains more than 200 language-related tasks covering skills from common sense reasoning to mathematics. In this paper, we focus on BIG-Bench Lite, a more compact subset of BIG-Bench comprising of 24 carefully selected tasks.

For the training data of our steering vector, we use the first 3 questions from each task in BIG-Bench Lite, for a total of 72 questions.

***ARC AI2*** (Clark et al., 2018): The ARC AI2 is a comprehensive dataset of grade-school level science questions designed to evaluate a LM Question Answering and Reasoning capabilities. It contains a total of 7,787 multiple choice science questions, split into a "Challenge" set of 2,590 "hard" questions, and an "easy" set of 5,197 questions.

We use the ARC AI2 Challenge set to evaluate our steered system on questions that it entirely hasn't been trained on. We evaluate on the "test" split of the ARC AI2 Challenge set, which contains roughly 1,170 questions.

***AGI Eval*** (Zhong et al., 2023): The AGI Eval dataset comprises of 20 tasks derived from official admission and qualification exams, from the American SAT to the Chinese Gaokao. It includes 18 multiple choice question tasks and 2 cloze tasks.

We use the AGI Eval's SAT Math Section (roughly 220 questions) to evaluate our steered system on questions that it hasn't been trained on.

## A.2 Models

***Llama3 8b Instruct*** (Dubey et al., 2024): The Llama 3 family of models is an open source collection of LMs released by Meta. We use the meta-llama/Meta-Llama-3-8B-Instruct checkpoint on Huggingface[1] as one of our target models for steering, as we find it yields more effective results to steering injections.

***Mistral 7b v0.2 Instruct*** (Jiang et al., 2023): Mistral 7b Instruct v0.2 is an instruction-tuned LM released by Mistral. We use the mistralai/Mistral-7B-Instruct-v0.2 checkpoint on Huggingface[2] as our other target model for steering.

***GPT 4o mini***: GPT 4o mini is one of OpenAI's flagship small LMs that has been instruction tuned to offer high performance at a significantly lower cost than its larger counterparts like GPT 4. We use OpenAI's API service to use GPT 4o mini as an automated answer scraper for model evaluations.

## A.3 Prompting Implementation Details

### A.3.1 Steering Vector Creation:

**Llama3 8b Instruct**, Think Step by Step:

1. System Prompt: `You are a helpful AI Assistant who answers questions step by step.`
2. User Prompt: `{question}`

**Llama3 8b Instruct**, Answer Immediately:

[1] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[2] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

1. System Prompt: `You are an AI Assistant who answers questions immediately without elaboration.`
2. User Prompt: `{question}`

**Mistral 7b Instruct v0.2**, Think Step by Step:

1. System Prompt: `You are a helpful AI Assistant who answers questions step by step.`
2. User Prompt: `{question}`

**Mistral 7b Instruct v0.2**, Answer Immediately:

1. System Prompt: `You are an AI Assistant who answers questions immediately without elaboration.`
2. User Prompt: `{question}`

### A.3.2 CoT Prompted Model Control

These prompts were used to serve as the "Control" experiments to compare with steered model performance.

**Llama3 8b Instruct**

1. System Prompt: N/A
2. User Prompt: Answer the following question thinking step by step: {question}

**Mistral 7b Instruct v0.2**

1. System Prompt: N/A
2. User Prompt: Answer the following question thinking step by step: {question}

### A.3.3 Steered Model Prompts

These prompts were used alongside steering vector injection.

**Llama3 8b Instruct**

1. System Prompt: N/A
2. User Prompt: {question}

**Mistral 7b Instruct v0.2**

1. System Prompt: N/A
2. User Prompt: {question}

### A.3.4 GPT 4o Mini Answer Extraction Prompts

These prompts were given to GPT 4o mini for extracting answers from models during evaluation over datasets.

**GSM8k:**

```
You are a precise answer extractor. Given a response to a test problem (which includes the test problem in the beginning), your task is to extract ONLY the final numerical answer. Do not include any units, explanations, or additional text. DO NOT SOLVE THE PROBLEM YOURSELF. If there are multiple numbers in the response, identify and return only the final answer. If no clear numerical answer is found, return -1758.

Response to analyze: {model_response}

Extract the final numerical answer from the above response. Your output should be ONLY the number, or None if no clear numerical answer is found.
```

**MMLU:**

```
You are a precise answer extractor. Given a response to a multiple choice problem, your task is to extract ONLY the final answer choice the model makes (either A, B, C, or D). Do not include any units, explanations, or additional text. DO NOT SOLVE THE QUESTION AND PROVIDE THE ANSWER YOURSELF. If there are multiple answers in the response, identify and return only the final answer. If no clear answer choice is found, return 'None'.

Response to analyze: {model_response}

Extract the final answer choice from the above response. Your output should be ONLY the final multiple choice answer picked in the response (either A, B, C, or D), or None if no clear answer choice is found.
```

**Arc AI2 Challenge:**

```
You are a precise answer extractor. Given a response to a multiple choice problem, your task is to extract ONLY the final
```

answer choice the model makes (either A, B, C, or D). Do not include any units, explanations, or additional text. DO NOT SOLVE THE QUESTION AND PROVIDE THE ANSWER YOURSELF. If there are multiple answers in the response, identify and return only the final answer. If no clear answer choice is found, return 'None'.

Response to analyze: {model_response}

Extract the final answer choice from the above response. Your output should be ONLY the final multiple choice answer picked in the response (either A, B, C, or D), or None if no clear answer choice is found.

**AGI Eval:**
You are a precise answer extractor. Given a response to a multiple choice problem, your task is to extract ONLY the final answer choice the model makes (either A, B, C, or D). Do not include any units, explanations, or additional text. DO NOT SOLVE THE QUESTION AND PROVIDE THE ANSWER YOURSELF. If there are multiple answers in the response, identify and return only the final answer. If no clear answer choice is found, return 'None'.

Response to analyze: {model_response}

Extract the final answer choice from the above response. Your output should be ONLY the final multiple choice answer picked in the response (either A, B, C, or D), or None if no clear answer choice is found.

## B Hyperparameter Search

In steering vector injection systems, there are two hyperparameters we optimize for:

1. $l$, the layer of steering vector injection
2. $c$, the injection coefficient we scale the steering vector by

In order to search for optimal hyperparameters, we conduct a grid search for steered LM systems over the first 300 GSM8k test questions over different combinations of $l$ and $c$. From this grid search, we find the following optimal hyperparameter combinations for both models:

1. **Llama3 8b Instruct**: $l = 16$, $c = 20$ (using single steering vector injection at the beginning and end of the user's input prompt)
2. **Mistral 7b Instruct v0.2**: $l = 13$, $c = 1$ (using steering vector injections at each token generation step)

## C Licensing for Models and Datasets Used

Our use of artifacts is consistent with their intended research purposes and licensing. The steering vectors and methodologies developed in this research are intended for academic and research use only, permitted under the licensing contracts of Llama3 8b Instruct, Mistral 7b v0.2 Instruct, and the GPT 4o mini API service under their terms of use for research purposes. Our application of datasets is permittable under the licensing of the GSM8k, MMLU, BIG-Bench, ARC AI2, and AGI Eval, as all of them are used with the purposes of research use and no modifications were made to any dataset questions.

Regarding the use of potentially sensitive data that includes offensive content or uniquely identifies individuals, we primarily utilized publicly available datasets and pre-trained language models that have been widely used in the research community. These datasets include GSM8k, MMLU, BIG-Bench, ARC AI2, and AGI Eval. To the best of our knowledge, these datasets do not contain personally identifiable information about individuals and we did not collect or generate any new data that would require additional anonymization steps in our experimental process. All experiments were conducted using existing, publicly available resources designed for research purposes.

## D Use of AI in Work

AI (Chat GPT 4) was used in the creation of the prompts for the answer extractor for GPT 4o mini, but nowhere else.

## E Parameters For Packages

In this paper, we use the `transformers` library with `AutoModelForCausalLLM` to load Mistral 7b Instruct v0.2 and Llama3 8b Instruct, and use the `datasets` library to load all datasets used from

HuggingFace. If there are any other questions regarding packaging implementation, please contact one of the authors.

## F Computational Infrastructure

All experiments were run with Python 3.8.10, and conducted on Ubuntu 22.04 using NVIDIA A40 GPUs running with CUDA version 11.8.0.