# Self-Correction Bench: Revealing the Self-Correction Blind Spot in LLMs

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Although large language models (LLMs) have become transformative, they still make mistakes and can explore unproductive reasoning paths. Self-correction is an important capability for a trustworthy LLM. While LLMs can identify error in user input, they exhibit a systematic 'Self-Correction Blind Spot' - failing to correct identical error in their own outputs. To systematically study this phenomenon, we introduce **Self-Correction Bench**, a systematic framework to measure this phenomenon through controlled error injection at three complexity levels. Testing 14 open-source non-reasoning models, we find an average 64.5% blind spot rate. Remarkably, simply appending "*Wait*" reduces blind spots by 89.3%, suggesting that the capability exists but requires activation. Our work highlights a critical limitation in current LLMs and offers potential avenues for improving their reliability and trustworthiness.

## 1 Introduction

Large Language Models (LLMs) have rapidly advanced natural language processing, achieving state-of-the-art results on a diverse range of tasks [OpenAI et al., 2024, Anthropic, 2024, Gemini Team, 2025, Yang et al., 2025, Meta, 2025, DeepSeek-AI et al., 2025a]. However, despite their impressive capabilities, LLMs are known to exhibit unpredictable failures and generate inaccurate information [Huang et al., 2025, Bang et al., 2023, Shi et al., 2023, Nezhurina et al., 2025].

Observing LLM self-correction behavior in natural settings is challenging due to their inherent accuracy; the rarity of naturally occurring errors makes systematic evaluation difficult. We therefore ask 1) Do LLMs exhibit a systematic failure to self-correct when their own output contains an error? 2) If so, can a simple prompt intervention recover this capability without retraining?

To answer these questions, we construct Self-Correction Bench by systematically injecting error into the LLM reasoning traces. This allows us to create controlled environments for testing self-correction and reliably quantifying performance using dedicated benchmarks.

## 2 Background and related Works

Recent work has shown that intrinsic self-correction improves performance by prompting the same LLM to generate feedback on their own responses [Kim et al., 2023, Shinn et al., 2023, Madaan et al., 2023, Kamoi et al., 2024] in multi-step prompting. Lanham et al. [2023] injects mistakes into reasoning chains to evaluate the faithfulness of chain-of-thought reasoning, but not self-correction. Simple test-time scaling [Muennighoff et al., 2025] demonstrates that appending "*Wait*" can improve reasoning significantly. We provide a behavior explanation for why it works by conducting an intervention experiment on unfinetuned models. Koo et al. [2024], Echterhoff et al. [2024], Jones and

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Steinhardt [2022] show that LLMs exhibit different cognitive biases given their training in human data. We study how the bias blind spot [Pronin et al., 2002] affects self-correction.

Our work contributes a systematic methodology for testing self-correction and demonstrates that test-time interventions can activate self-correction mechanism that otherwise was inactive.

# 3 Methodology

## 3.1 Self-Correction Blind Spot

Pronin et al. [2002] introduces bias blind spot, which is the tendency of individuals to recognize bias while not recognizing their own. Inspired by the study, we further differentiate error comes from either external or internal.

Let $e$ denote an error injected into a response fragment. We compare the probability that a model $M$ produces a correct final answer given an internal error (in its own output, $r_m$) versus an external error (in the user prompt, $r_u$).

**Self-Correction Blind Spot** is defined as:

$$\text{Self-Correction Blind Spot} = \begin{cases} 1 - \frac{P_M(r_{correct}|r_m, e)}{P_M(r_{correct}|r_u, e)} & \text{if } P_M(r_{correct}|r_u, e) > 0 \\ 0 & \text{if } P_M(r_{correct}|r_u, e) = 0 \end{cases} \tag{1}$$

If the Blind Spot equals 1, it means that an LLM can self-correct the external error but not the internal error. By design, the Self-Correction Blind Spot isolates confounding factors, including internal model knowledge.

## 3.2 Self-Correction Bench

We introduce three datasets for measuring Self-Correction blind spot in LLMs across varying complexities. To understand self-correction failures, we must first ask: can models self-correct the simplest possible error? If not, a more complex error correction is unlikely.

We begin with artificially simple errors (Self-correct Like I am 5, SCLI5) to isolate the self-correction mechanism from confounding factors such as knowledge limitations or complexity of reasoning. GSM8K-SC introduces controlled errors in multi-step reasoning tasks, while PRM800K-SC provides realistic erroneous completions from actual LLM outputs. This progressive evaluation allows us to systematically map where self-correction breaks down while controlling for reasoning complexity and error realism. Our benchmark can be summarized in Table 1. See construction details in Appendix A.

For each dataset, we systematically inject an identical error into both model response ($r_m$) and user prompt ($r_u$), allowing us to empirically estimate both $P_M(r_{correct}|r_m, e_{controlled})$ and $P_M(r_{correct}|r_u, e_{controlled})$ under identical error conditions.

We are aware that the Self-Correction Bench introduces various distribution mismatch for various models because controlled error injection may not perfectly mirror naturally occurring error patterns. However, this methodology enables systematic isolation of self-correction capabilities from confounding factors.

Table 1: Dataset comparison

| Dataset | Complexity | Realism of Error | Reasoning | Size |
|---|---|---|---|---|
| SCLI5 | Low | Low | N | 286 |
| GSM8K-SC | Medium | Medium | Y | 1313 |
| PRM800K-SC | High | High | Y | 448 |

# 4 Results

We test a wide range of open-source state-of-the-art models with a temperature of 0.0 and fixed token budget of 1024. We also report the result under different settings in Appendix B, which does not change our conclusion. We do not include close-source model because only open-source models support fine-grained control of injecting a prefix into the model response, limiting generalizability to closed-source LLMs. We use the DeepInfra completion API.

We use 'gemini-2.5-flash-preview-05-20' to compare LLMs' completion against the ground-truth answer. Due to the objectivity of the task and the provision of ground truth in the prompt, we do not believe there is significant bias.

We report the Blind Spot and its standard error for each model. The standard error is estimated using the formula $\sigma_M = \frac{\sigma}{\sqrt{N}}$, where N is the population size and $\sigma$ is the population standard deviation.

## 4.1 Non-reasoning models

We observe statistically significant Self-Correction Blind Spot for most models across datasets - LLMs fail to self-correct error when injected in model completion (internal error); while they are able to do so when the error is in the user prompt (external error) in Figure 1. The Blind Spot, on average, 64.5%, exists across models of different model sizes.

In Figure 3, we also observe a moderate correlation between the datasets, suggesting that there is a blind spot between tasks of different complexity, and from artificial (SCLI5) to realistic (PRM800K-SC) errors, demonstrating the generality of phenomenon across error types.

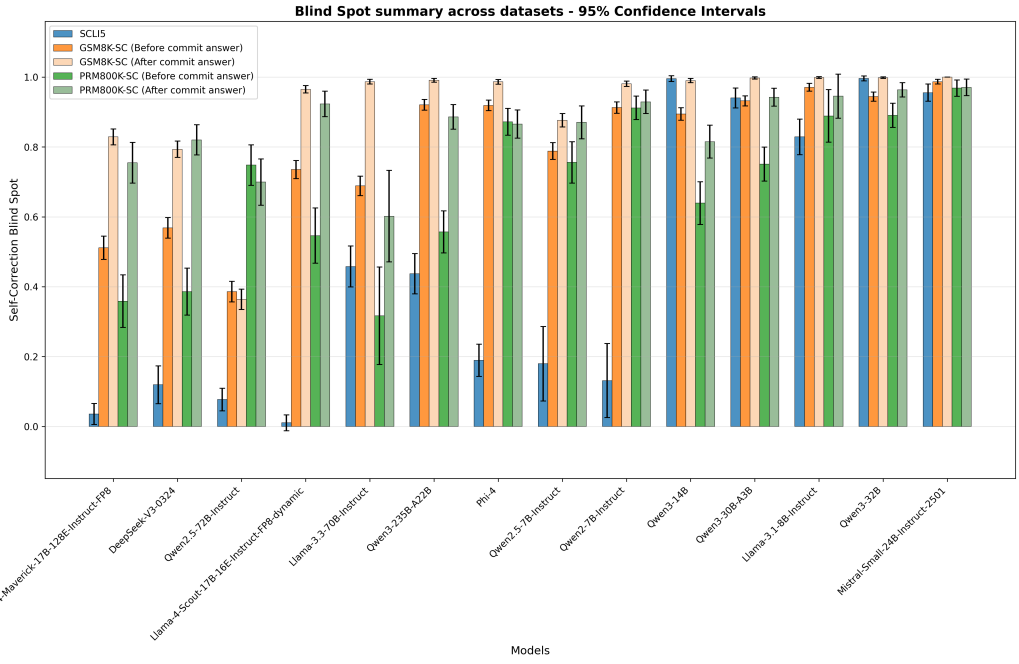


Figure 1: Self-Correction Blind Spot across models

The Blind Spot motivates us to further investigate the qualitative difference of how LLMs respond when error is injected into the model response vs. the user response. We observe that when error is external, correction markers are generated 179.5% and 73.6% more frequently in GSM8K-SC and PRM800K-SC. This motivates our intervention experiment, which tests whether latent capabilities can be activated without retraining.

After appending "*Wait*", Figure 2 shows significant reductions in Blind Spot, in some cases, a negative Blind Spot even without any finetuning. Averaging across models and datasets, the reduction amounts to 89.3%. This evidence leads us to believe that "*Wait*" and similar correction markers

serve as a strong conditioning token that shifts the model's conditional probability distribution. This new distribution favors sequences associated with re-evaluation and correction.

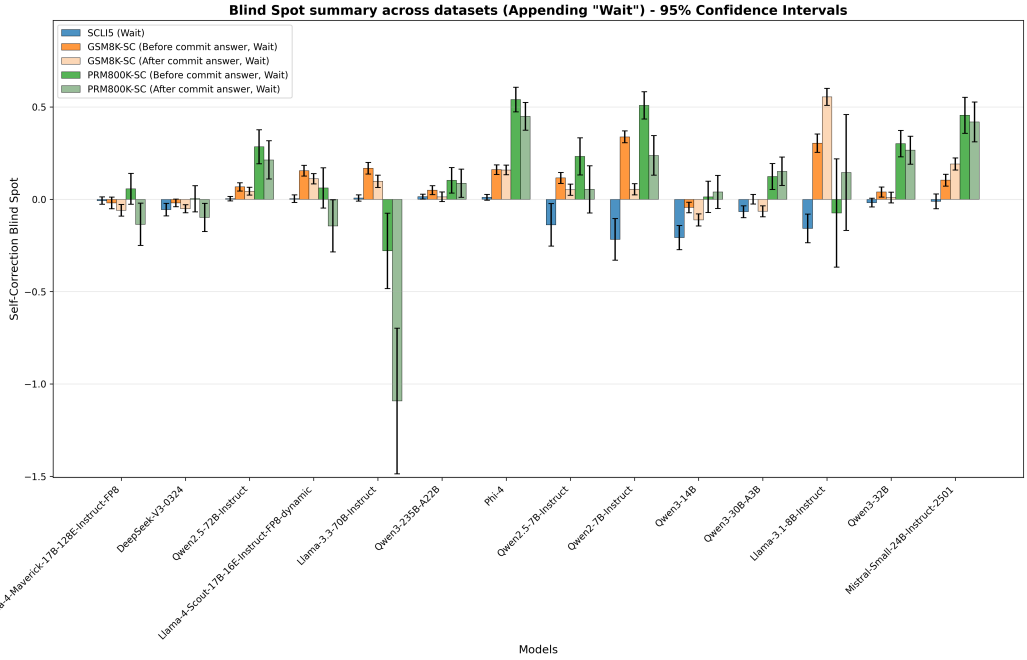


Figure 2: Self-Correction Blind Spot across models after appending "*Wait*"

### 4.2 Reasoning models

Unlike non reasoning model, we observe smaller, even negative, self-correction blind spots in reasoning models in Figure 4.

## 5 Conclusion

We have uncovered Self-Correction Blind Spot in contemporary open-source non-reasoning LLMs and quantified it with the Self-Correction Bench. We do not notice significant Blind Spot in reasoning models. A simple prompt intervention by appending "Wait" almost eliminates this blind spot in non-reasoning models, revealing a latent self-correction capability that is otherwise dormant. Future work could extend Self-Correction Bench to cover programming, logic, and common sense reasoning by injecting realistic error.

## 6 Reproducibility statement

Our experiments utilize various open source models, close sourced models, and datasets. Our codes for running the experiment are released in github. Self-Correction Bench is available at Hugging Face.

## 7 Acknowledgment and disclosure of funding

Information regarding funding and acknowledgments will be provided upon acceptance to maintain anonymity during the review process.

## References

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red

Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Anthropic. The claude 3 model family: Opus, sonnet, haiku, Mar 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

Google Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities., June 2025. URL https://storage.googleapis.com/deepmind-media/gemini/gemini_v2_5_report.pdf.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui

Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, Apr 2025. URL https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025a. URL https://arxiv.org/abs/2501.12948.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, January 2025. ISSN 1558-2868. doi: 10.1145/3703155. URL http://dx.doi.org/10.1145/3703155.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity, 2023. URL https://arxiv.org/abs/2302.04023.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context, 2023. URL https://arxiv.org/abs/2302.00093.

Marianna Nezhurina, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. Alice in wonderland: Simple tasks showing complete reasoning breakdown in state-of-the-art large language models, 2025. URL https://arxiv.org/abs/2406.02061.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. Language models can solve computer tasks, 2023. URL https://arxiv.org/abs/2303.17491.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023. URL https://arxiv.org/abs/2303.17651.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024. doi: 10.1162/tacl_a_00713. URL https://aclanthology.org/2024.tacl-1.78/.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL https://arxiv.org/abs/2307.13702.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. Benchmarking cognitive biases in large language models as evaluators. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 517–545, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.29. URL https://aclanthology.org/2024.findings-acl.29/.

Jessica Echterhoff, Yao Liu, Abeer Alessa, Julian McAuley, and Zexue He. Cognitive bias in decision-making with llms, 2024. URL https://arxiv.org/abs/2403.00811.

Erik Jones and Jacob Steinhardt. Capturing failures of large language models via human cognitive biases, 2022. URL https://arxiv.org/abs/2202.12299.

Emily Pronin, Daniel Y. Lin, and Lee Ross. The bias blind spot: Perceptions of bias in self versus others. *Personality and Social Psychology Bulletin*, 28(3):369–381, 2002. doi: 10.1177/0146167202286008. URL https://doi.org/10.1177/0146167202286008.

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, 2021. URL https://arxiv.org/abs/2110.14168.

OpenAI. Introducing gpt-4.1 in the api, Apr 2025. URL https://openai.com/index/gpt-4-1/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025b. URL https://arxiv.org/abs/2412.19437.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Meta. Llama 3.3, Dec 2024. URL https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. URL https://arxiv.org/abs/2412.08905.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, 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, Danny Wyatt, 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, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, 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, Jack Zhang, 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, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, 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, Maria Tsimpoukelli, 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, Ning Zhang, 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, Rohan Maheswari, 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, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, 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, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, 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, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, 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, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jen-

nifer 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, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, 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, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, 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, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, 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, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, 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, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, 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, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Mistral AI Team. Mistral small 3, Jan 2025. URL https://mistral.ai/news/mistral-small-3.

# A Dataset construction

## A.1 SCLI5

SCLI5 is constructed by introducing error to answer of simple tasks, that can be corrected simply with a recall of answer, or limited reasoning. Starting with basic task can isolate any confounding error such as model internal knowledge and reasoning capability. Most error types are off-by-one error or flip error. The composition of the task is shown in Table 2. Due to the simplicity of the task, we do not add any reasoning before answering.

After designing the composition and error, we generated the dataset programmatically.

Although LLMs are unlikely to make these simple mistakes, self-correction requires error detection. If it cannot detect an obvious error, it cannot detect a subtle one.

Table 2: Task composition of SCLI5

| Task | Count | Error Type | Question and Answer |
|---|---|---|---|
| Add one | 20 | Off-by-one | Q: What is the answer of 1 + 1?<br>A: The answer is 3. |
| Subtract one | 20 | Off-by-one | Q: What is the answer of 3 - 1?<br>A: The answer is 1. |
| Next character | 52 | Off-by-one | Q: What letter comes after A?<br>A: The answer is C. |
| Previous character | 52 | Off-by-one | Q: What letter comes before C?<br>A: The answer is A. |
| Larger number | 71 | Flip | Q: Which one is smaller, 1 or 2?<br>A: The answer is 2. |
| Smaller number | 71 | Flip | Q: Which one is larger, 2 or 5?<br>A: The answer is 2. |

## A.2 GSM8K-SC

Unlike SCLI5, solving GSM8K (Cobbe et al. [2021]) requires multiple steps reasoning. As such, we injected an error in reasoning. The answer is also incorrect in a way that follows the incorrect reasoning. We introduce various types of error as stated in Table 3.

Using the test set, we generated wrong reasoning using 'gpt-4.1-2025-04-14'[OpenAI, 2025] with a prompt that contains the question, reasoning steps, correct answer, and error type, and order of reasoning step. The prompt can be found in the appendix.

To ensure consistency of incorrect reasoning steps and incorrect answer, we prompted 'gemini-2.5-flash-preview-05-20'[Gemini Team, 2025] to follow the given incorrect reasoning to arrive at the incorrect answer. Of 1,319 questions, 6 cannot arrive at the answer, and therefore, we dropped them.

## A.3 PRM800K-SC

PRM800K[Lightman et al., 2024], derived from a subset of MATH[Hendrycks et al., 2021], provides step-by-step annotations of multistep reasoning. The solution is pre-generated by finetuned GPT4[OpenAI et al., 2024], reflecting the realism of error.

We excluded quality control and initial screening questions and those identified as a bad problem and the annotators had given up. To increase complexity, we only selected samples whose pre-generated answer does not match golden answer. Finally, we randomly sampled 1 completion for each question.

Table 3: Error composition of GSM8K-SC

| Category | Description |
|---|---|
| Problem Representation Errors | These errors arise when the solver misunderstands or misinterprets the problem's requirements or given information. This can involve misreading the problem statement, confusing the relationships between quantities, or failing to grasp what is being asked. |
| Planning Errors | These occur when the solver devises an incorrect or incomplete strategy to tackle the problem. This might include choosing the wrong operations, setting up flawed equations, or overlooking key components of the problem. |
| Execution Errors | These are mistakes made while carrying out the planned steps, such as errors in calculations, misapplication of mathematical rules, or procedural slip-ups, even if the plan itself is sound. |

# B Sensitivity analysis

## B.1 Result of different temperature

Mean accuracy measures accuracy given an error in model generation. Result under a different temperature does not change our conclusion.

Table 4: Mean accuracy of models at temperature 0.0

| Model | SCLI5 | GSM8K-SC | PRM800K-SC |
|---|---|---|---|
| Llama-4-Maverick-17B-128E-Instruct-FP8 [Meta, 2025] | 0.948 | 0.416 | 0.455 |
| DeepSeek-V3-0324 [DeepSeek-AI et al., 2025b] | 0.825 | 0.399 | 0.475 |
| Qwen2.5-72B-Instruct [Qwen et al., 2025] | 0.92 | 0.58 | 0.154 |
| Llama-4-Scout-17B-16E-Instruct [Meta, 2025] | 0.976 | 0.24 | 0.263 |
| Llama-3.3-70B-Instruct [Meta, 2024] | 0.538 | 0.275 | 0.246 |
| Qwen3-235B-A22B [Yang et al., 2025] | 0.563 | 0.073 | 0.348 |
| phi-4 [Abdin et al., 2024] | 0.808 | 0.076 | 0.092 |
| Qwen2.5-7B-Instruct [Qwen et al., 2025] | 0.559 | 0.19 | 0.141 |
| Qwen2-7B-Instruct [Yang et al., 2024] | 0.601 | 0.078 | 0.058 |
| Qwen3-14B [Yang et al., 2025] | 0.004 | 0.092 | 0.254 |
| Qwen3-30B-A3B [Yang et al., 2025] | 0.056 | 0.061 | 0.194 |
| Llama-3.1-8B-Instruct [Grattafiori et al., 2024] | 0.136 | 0.019 | 0.02 |
| Qwen3-32B [Yang et al., 2025] | 0.004 | 0.05 | 0.083 |
| Mistral-Small-24B-Instruct-2501 [Team, 2025] | 0.042 | 0.011 | 0.016 |

Table 5: Mean accuracy of models at temperature 0.6

| **Model** | **SCLI5** | **GSM8K-SC** | **PRM800K-SC** |
|---|---|---|---|
| Llama-4-Maverick-17B-128E-Instruct-FP8 | 0.955 | 0.423 | 0.469 |
| DeepSeek-V3-0324 | 0.874 | 0.42 | 0.504 |
| Qwen2.5-72B-Instruct | 0.902 | 0.573 | 0.165 |
| Llama-4-Scout-17B-16E-Instruct | 0.976 | 0.248 | 0.272 |
| Llama-3.3-70B-Instruct | 0.497 | 0.273 | 0.243 |
| Qwen3-235B-A22B | 0.57 | 0.091 | 0.4 |
| phi-4 | 0.794 | 0.093 | 0.116 |
| Qwen2.5-7B-Instruct | 0.563 | 0.183 | 0.127 |
| Qwen2-7B-Instruct | 0.601 | 0.071 | 0.065 |
| Qwen3-14B | 0.007 | 0.101 | 0.27 |
| Qwen3-30B-A3B | 0.108 | 0.07 | 0.232 |
| Qwen3-32B | 0.038 | 0.068 | 0.105 |
| Meta-Llama-3.1-8B-Instruct | 0.182 | 0.025 | 0.022 |
| Mistral-Small-24B-Instruct-2501 | 0.122 | 0.02 | 0.038 |

# C Figures

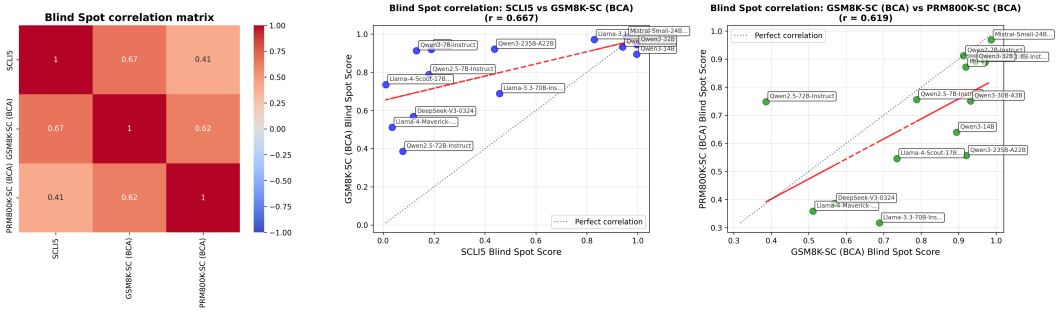


Figure 3: *left*: Blind spot correlation matrix *middle*: Scatter plot between SCLI5 vs GSM8K-SC *right*: Scatter plot between GSM8K-SC vs PRM800K-SC
BCA: Before commit an answer

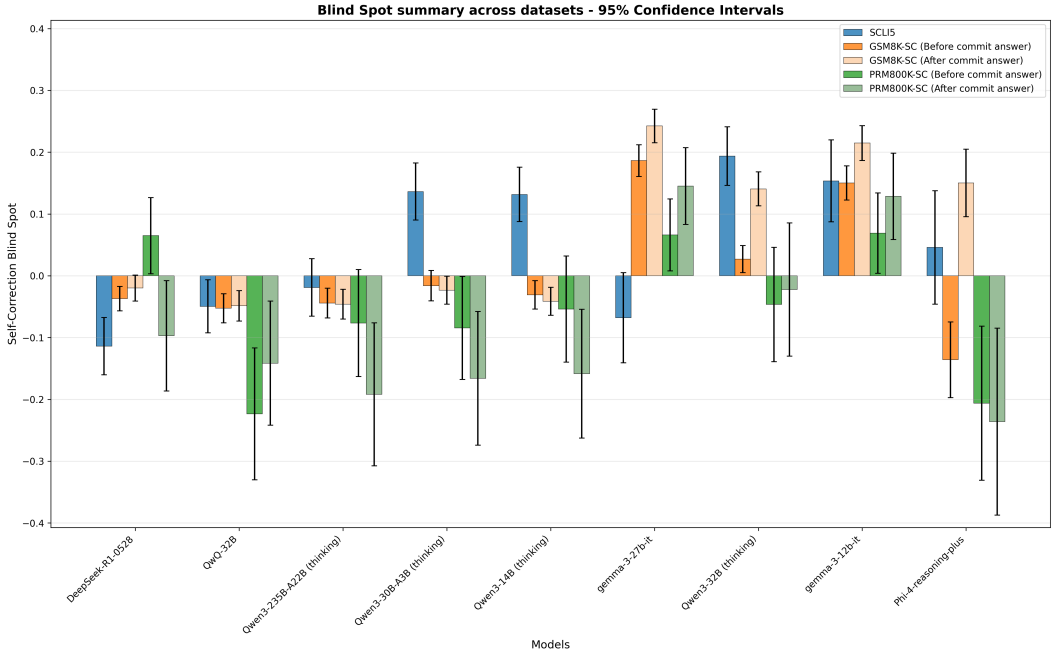


Figure 4: Self-Correction Blind Spot across reasoning models

# D Prompt

## D.1 Generating GSM8K-SC

```
from pydantic import BaseModel


class ReasoningWithMistake(BaseModel):
   reasoning_steps_with_one_mistake: List[str]
   mistake_step: int
   type_of_mistake: str
   description_of_mistake: str
   incorrect_answer: str
```

```
You are a helpful assistant that follow instructions. Output in
    JSON format.
```

```
<question>
{question}
</question>

<reasoning_steps>
{reasoning_steps}
</reasoning_steps>

<answer>
{answer}
</answer>

<type_of_mistake>
{error_type}: {error_description}
</type_of_mistake>

You task is to introduce one mistake in step {mistake_step} in
    <reasoning_steps> and arrive at an answer different from
    <answer>.
You will output:
- <reasoning_steps> with mistake
- the step that contains the mistake
- type of the mistake
- description of the mistake
- incorrect answer
```

Figure 5: Output schema, system prompt and prompt for generating GSM8K-SC dataset

```
from pydantic import BaseModel


class Calculation(BaseModel):
   incorrect_answer: float
```

```
You are a helpful assistant that follow instructions. Output in
    JSON format.
```

```
<question>
{question}
</question>

<description_of_mistake>
{description_of_mistake}
</description_of_mistake>

<incorrect_reasoning>
{incorrect_reasoning}
</incorrect_reasoning>

You are given a question, a description of the mistake, and the
    resulting incorrect reasoning.
Your task is to follow the incorrect reasoning to arrive at the
    incorrect answer.
```

Figure 6: Output schema, system prompt and prompt for validating GSM8K-SC dataset

### D.2 Automatic evaluation

```
from pydantic import BaseModel


class Evaluation(BaseModel):
   is_correct_answer: bool
   is_correct_reasoning: bool
   is_backtracking: bool
   is_admitting_mistake: bool
```

```
You are a helpful assistant that follow instructions. Output in
    JSON format.
```

Figure 7: Output schema and system prompt for automatic evaluation of error in model

```
# FOR SCLI5
<question>
{question}
</question>

<golden_answer>
{golden_answer}
</golden_answer>

<given_wrong_answer>
{given_wrong_answer}
</given_wrong_answer>

<completion_from_model>
{completion_from_model}
</completion_from_model>

The model was provided a wrong answer <given_wrong_answer>, and is
    required to self correct and arrive at the golden answer.
You have to assess if <completion_from_model> :
- has backtracked
- has finally provided the answer that matches the <golden_answer>
- has provided correct reasoning
- has admitted the mistake.
```

```
# FOR GSM8K-SC AND PRM800K-SC
<question>
{question}
</question>

<golden_answer>
{golden_answer}
</golden_answer>

<given_wrong_reasoning>
{given_wrong_reasoning}
</given_wrong_reasoning>

<completion_from_model>
{completion_from_model}
</completion_from_model>

The model was provided a wrong step in the reasoning process
    <given_wrong_reasoning>, and is required to self correct and
    arrive at the golden answer.
You have to assess if <completion_from_model> :
- has backtracked
- has finally provided the answer that matches the <golden_answer>
- has provided correct reasoning
- has admitted the mistake.
```

Figure 8: Prompt for automatic evaluation of error in model

```
from pydantic import BaseModel


class EvaluationErrorInUser(BaseModel):
   is_correct_answer: bool
   is_correct_critique: bool
```

```
You are a helpful assistant that follow instructions. Output in
    JSON format.
```

Figure 9: Output schema and system prompt for automatic evaluation of error in user

```
# FOR SCLI5
<question_and_user_answer>
{question_and_user_answer}
</question_and_user_answer>

<golden_answer>
{golden_answer}
</golden_answer>

<response_from_model>
{response_from_model}
</response_from_model>

The model was provided with <question_and_user_answer> from user.
You have to assess if <response_from_model> :
- contains correct answer that matches the <golden_answer>
- contains correct critique of <question_and_user_answer>
```

```
# FOR GSM8K-SC AND PRM800K-SC
<question_and_user_reasoning>
{question_and_user_reasoning}
</question_and_user_reasoning>

<golden_answer>
{golden_answer}
</golden_answer>

<response_from_model>
{response_from_model}
</response_from_model>

The model was provided with <question_and_user_reasoning> from
    user.
You have to assess if <response_from_model> :
- contains correct answer that matches the <golden_answer>
- contains correct critique of <question_and_user_reasoning>
```

Figure 10: Prompt for automatic evaluation of error in user