Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *arXiv preprint arXiv:2302.03668*, 2023.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Yanggang Wang, Haiyu Li, and Zhilin Yang. Gps: Genetic prompt search for efficient few-shot learning. *arXiv preprint arXiv:2210.17041*, 2022.

Weizhe Yuan, Kyunghyun Cho, and Jason Weston. System-level natural language feedback. *arXiv preprint arXiv:2306.13588*, 2023.

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E Gonzalez. Tempera: Test-time prompt editing via reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pp. 12697–12706. PMLR, 2021.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022a.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*, 2022b.

A SOME FAILURE CASES

Although LLMs show the power of optimizing basic math problems (Section 3) and prompts (Section 4), we see some limitations across all optimizer LLMs that may impede their power of solving more challenging problems. These limitations include:

- **Hallucinating the values that need to come from math calculation**: The optimizer LLMs often output contents like “the function value at (5, 3) is 15” despite that the true value is not 15. The model will get it right if external tools that can reliably calculate the value are triggered. When and how to trigger such tool use cases remains an interesting topic (see e.g., (Schick et al., 2023; Cai et al., 2023)).
- **Generating solutions already appeared in context even if we tell it to "Give me a new (w, b) pair that is different from all pairs above"**: the optimizer LLMs do not 100% reliably follow this instruction even if its own outputs often include sentences like “I will provide a new pair that is different”, making the output self-contradictory. The output is almost guaranteed to be different from in-context old solutions when the model output contains a comparison of the new pair and all old pairs, though. Thus (implicitly) triggering such behaviors may be a solution. How to implement this feature without harming the instruction following performance of other parts remains an interesting topic to study.
- **In black-box math optimization, getting stuck at a point that is neither global nor local optimal**: This often occurs in two linear regression cases: (a) The in-context exemplars all share the same w or b that is different from w_{true} or b_{true}. This case is more likely to be avoided when a larger number of past solutions are included in the meta-prompt; (b) one or several of the best previous solutions in the meta-prompt have ws and bs in quantitatively opposite directions from the global optima w_{true} and b_{true}: for example, the ws are all smaller than w_{true} while the bs are all larger than b_{true}. Since the optimizer model often proposes to only increase w or decrease b when the past solutions in meta-prompt share w or b, the optimization will get stuck if either increasing w or decreasing b would increase the objective value. This issue is mitigated by sampling multiple new solutions (thus more exploration) at each step.
- **Hard to navigate a bumpy loss landscape**: Like other optimizers, it is harder for the optimizer LLM to optimize black-box functions when the loss landscape gets more complicated. For example, when minimizing the Rosenbrock function $f(x, y) = (a-x)^2+b(y-x^2)^2$ with $a = 20$ (whose global optimal point is $x = 20$, $y = 400$) with 5 starting points in $[10, 20] \times [10, 20]$, the optimization often gets stuck at around $(0, 0)$. This is because the optimizer LLM sees a decrease of objective value when it drastically decreases both x and y to 0. Then starting from $(0, 0)$, the optimizer LLM is hard to further navigate x and y along the narrow valley in the loss landscape towards $(20, 400)$ (Figure 13).

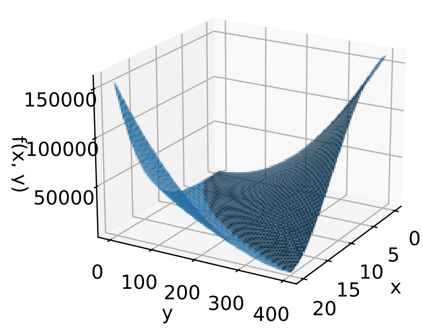

Figure 13: A visualization of the landscape of the Rosenbrock function $f(x, y) = (a-x)^2+b(y-x^2)^2$ with $a = 20$ and $b = 1$. The global optima is at $x = 20$, $y = 400$ with function value 0. The function value at $x = 0$, $y = 0$ is 400. The landscape has a narrow valley between $(0, 0)$ and $(20, 400)$.

B Prompting Formats for Scorer LLM

Figure 14, 15, and 16 show examples of the Q_begin, Q_end, and A_begin prompting formats when the “QA” pattern is present. The “QA” pattern is eliminated when prompting instruction-tuned scorer models like `text-bison` with the Q_begin and Q_end formats (Figure 17 and 18).

```
Q: {instruction}
Janet’s ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers’ market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers’ market?

A:
```

Figure 14: The Q_begin prompting format on a GSM8K test exemplar with the "QA" pattern.

```
Q: Janet’s ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers’ market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers’ market?
{instruction}

A:
```

Figure 15: The Q_end prompting format on a GSM8K test exemplar with the "QA" pattern.

```
Q: Janet’s ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers’ market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers’ market?

A: {instruction}
```

Figure 16: The A_begin prompting format on a GSM8K test exemplar.

```
{instruction}
Janet’s ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers’ market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers’ market?
```

Figure 17: The Q_begin prompting format on a GSM8K test exemplar without the "QA" pattern.

```
Janet’s ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers’ market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers’ market?
{instruction}
```

Figure 18: The Q_end prompting format on a GSM8K test exemplar without the "QA" pattern.