Weak-to-Strong In-Context Optimization of Language Model Reasoning

Keshav Ramji[*][†]
IBM Research AI
keshav.ramji@ibm.com

Alok N. Shah[*]
University of Pennsylvania
alokshah@sas.upenn.edu

Vedant Gaur[*]
University of Pennsylvania
vedantg@upenn.edu

Khush Gupta[*]
University of Pennsylvania
khushg@upenn.edu

Abstract

Large language models (LLMs) have demonstrated remarkable in-context learning capabilities, leveraging demonstrations to adeptly perform a task. Recent works have shown that such models can perform optimization over a response scoring function, evaluating the quality of suboptimal generations and applying them as exemplars to produce a better response. In this work, we seek to further explore this phenomenon and determine whether strong LLMs can optimize their reasoning paths by leveraging differentiated copies of a weak model. Central to our approach is the use of filler tokens interleaved after each step in the reasoning chain. We then define reasoning optimality, our implicit objective function, in terms of the "efficiency" as measured by the number of steps. At inference time, three copies of the weak model fine-tuned on synthetic data with varying degrees of efficiency are used to generate responses for in-context optimization with the strong model. We evaluate this method on the MMLU benchmark with Gemma-2 2B-it weak learners and Llama-3.1-405B-Instruct as the strong model, and demonstrate that our approach improves performance in a cheap manner.

1 Introduction

Modern foundation models are capable of performing challenging reasoning tasks, with phenomena such as in-context learning being a primary driver (Wei et al., 2023, 2022). This includes the ability to refine or optimize generations (Yang et al., 2024a; Madaan et al., 2023; Shinn et al., 2023), interact with external tools in an agentic manner (Schick et al., 2023; Wang et al., 2024), and perform reasoning by leveraging test-time compute to sample at a greater scale (Snell et al., 2024; Wu et al., 2024). In particular, the ability to optimize generations at inference time with respect to an implicitly defined reward function holds promise in improving model reasoning in a cheap manner (Yang et al., 2024a). However, as model capabilities may continue to grow towards superhuman levels that humans cannot reliably judge or oversee, it is important to develop methods that can affect strong model performance, while leveraging smaller models at the human-controllable scale (Burns et al., 2024; Yang et al., 2024b). This requires us to study how small models may be adapted so they may be used to indirectly guide larger models towards desirable behaviors, including further boosting strong model capabilities.

[*] Equal contribution.
[†] Work done while at the University of Pennsylvania.

ATTRIB: 2nd Workshop on Attributing Model Behavior at Scale at NeurIPS 2024.

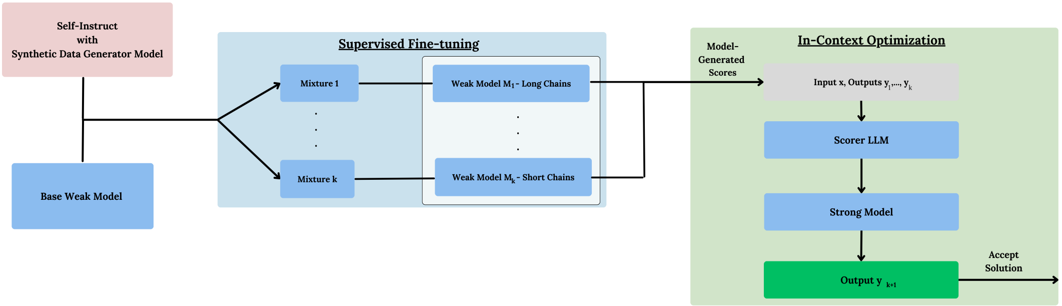

Figure 1: An illustration of our approach. We first use a modification on Self-Instruct to generate synthetic reasoning data with filler tokens, and fine-tune k copies of the weak model on the resulting data mixtures. Then, the generations are scored by an separate LLM, and the outputs and judged rewards are passed to the strong model. The strong model uses these as suboptimal generations to inductively learn the function to optimize with respect to and generate an improved response.

In this work, we seek to determine whether the reasoning traces of weaker language models can be used to optimize the reasoning of a frontier LM. We train three *differentiated* copies of a weak model, Gemma 2 2B (Team et al., 2024) with unique sets of synthetic data, incorporating filler tokens (Goyal et al., 2024; Pfau et al., 2024; Zelikman et al., 2024) placed at the end of each reasoning step, generated by Self-Instruct (Wang et al., 2023). We find these models to perform differently from one another on the MMLU benchmark (Hendrycks et al., 2021a), in order of *reasoning efficiency*: that is, training on data with fewer steps (and therefore, the number of filler tokens) improves performance. Furthermore, then using the generations of these weak model copies in performing in-context optimization as in OPRO (Yang et al., 2024a) boosts the performance of Llama-3.1-405B-Instruct (Dubey et al., 2024) on MMLU by 0.6%. This means that the generations of Llama-3.1-405B-Instruct are being optimized, without sampling and scoring initial response attempts from said strong model (as is done in the OPRO method). As such, the improvements of this procedure can be directly attributed to reasoning chains *generated by weak models*, suggesting that this is a viable, cheap means of boosting performance at inference time, and yielding a promising scope for future work.

2 Methods

2.1 Preliminaries

LLMs as Optimizers. Given the ubiquity of optimization problems, OPRO (Yang et al., 2024a) seeks to solve this by leveraging the language model as the optimizer directly. The method involves sampling a few responses to be used as in-context exemplars for the model of prior attempts, along with scores of these responses to contrast their relative quality. Then, prompting the model in-context to produce a response which yields an improved score does indeed succeed in doing so, as demonstrated on both more classic optimization problems (e.g. traveling salesman problem) and prompt optimization, identifying more effective prompts.

More formally, we can define $D_x = \{y_i, r_i\}$ for $i \in [k]$ to be the set of k in-context demonstrations and their scores (y_i and r_i, respectively), given an input x. The reward function which governs the scores can be defined as $R(y, f(x, D_x))$; we assume R to be a well-defined response quality function, implying that the scores it assigns correspond to a monotonically increasing function correlated with human judgements. Take $\mathcal{H}_S$ to be the hypothesis space of the strong model; then, the optimization problem we propose to solve can be stated as:

$$\max_{f \in \mathcal{H}_S} E_{(x, D_x)}[R(y, f(x, D_x))] \tag{1}$$

In this work, we note that the problem of improving LM reasoning capabilities is also an optimization problem, and propose one dimension that may be optimized over, *reasoning efficiency*.

Weak-to-Strong Generalization. To enable scalable oversight of frontier models which will develop superhuman capabilities towards the pursuit of artificial general intelligence (AGI), weak-to-strong generalization (Burns et al., 2024) fine-tunes GPT-4 on feedback labels generated by GPT-2, finding that this bridges the substantial gap between the two models. They demonstrate several methods that improve performance, including progressive weak-to-strong fine-tuning; strong models also appear to overfit to the weak labels, making it challenging to recover strong model performance.

2.2 Weak-to-Strong In-Context Optimization

While OPRO uses the same model to generate initial solutions to use as in-context exemplars, we separately use various models for each of the three stages of sampling, scoring, and in-context learning. To enable a weak-to-strong approach at inference time, we use weaker models for sampling responses, and a strong model to learn how to apply them to optimize. In particular, we use k different weak models; the design of our weak learners is discussed in Section 2.3.

Formally, define $\mathcal{H}_{W_i}$ for $i \in [k]$ to be the hypothesis class of each of the k weak models, each of which will generate a demonstration. That is, take $D_{x,i} = \{f(x), r_i\}, f \in \mathcal{H}_{W_i}$, and $D_x = \bigcup_{i \in [k]} D_{x,i}$. Then, our optimization function remains as (1), applying this method of obtaining D_x instead.

While the weak-to-strong generalization work introduces the performance gap recovered (PGR) metric, we instead analyze the performance change relative to the strong model's performance; since there are no parameter updates, this is more appropriate for the in-context learning setting.

2.3 Designing Weak Learners

To simplify the design of the weak learners, we take a fixed base model and fine-tune k copies of this model on different mixtures. We use a modification on the Self-Instruct method (Wang et al., 2023), first generating instructions, then classifying whether it is a reasoning problem or not, and then generating samples for each instruction. Notably, we introduce filler tokens at the end of each step in the reasoning chain, on the basis of prior works which have suggested that these tokens store context of the steps that precede it, improving computational width when fine-tuning on data with them (Goyal et al., 2024; Zelikman et al., 2024). By changing the sample generation stage to use in-context exemplars of reasoning chains with a "\t" token interleaved between steps, and prompting the model to produce k outputs for each input, where outputs $1, 2, \ldots, k$ use fewer steps, we yield k different synthetic data mixtures from Gemma-2 9B-it to train the weak learner models [3].

3 Experiments

3.1 Evaluation Setup

We use Gemma-2 2B-it as the weak learner; this is a strong performing small language model (SLM), and is already instruction-tuned. We rely on instruction-following weak learners to ensure that that the weak response quality is not poor, but rather, mediocre, leaving room for improvement. We first evaluate the three fine-tuned copies of Gemma-2 2B-it on MMLU (Hendrycks et al., 2021a), with the results reported in Table 1. Then, we perform in-context optimization using this weak learners with Llama-3.1-405B-Instruct as the strong model and Llama-3.1-70B-Instruct as the scoring model. We prompt the scorer model to provide a score to each generation from 1 to 100 (where 100 is the best).

3.2 Results

We first report the results of evaluating the three weak models (Gemma-2 2B-it copies) on MMLU, each having been trained on 5,528 samples of generated synthetic data, in Table 1; we denote the model trained on "output 1" (longest reasoning chains) as Weak-Long, the one trained on "output 2" (medium-length reasoning chain) as Weak-Medium, and the one trained on "output 3" (shortest chains) as Weak-Short. Our results confirm the hypothesis that reasoning efficiency does correlate with downstream reasoning benchmark performance (presumably, provided that the synthetic data samples generated are indeed correct), at least for MMLU.

[3] We note that one could use different models altogether for an ensemble-style approach; however, attribution then requires an understanding of the drivers of performance for each model.

Model	Gemma-2 2B-it	Weak-Long	Weak-Medium	Weak-Short
MMLU (5-shot)	52.13%	52.42%	52.48%	52.61%

Table 1: MMLU results on the three weak models fine-tuned from Gemma-2 2B-it; we find that models trained on shorter reasoning chains by prompting the model during Self-Instruct to provide three responses of contrasting lengths yield better performance.

Next, we evaluated the weak-to-strong in-context optimization approach, with the Llama-3.1-405B strong model as noted in Section 3.1, on the MMLU, GPQA, and MATH datasets (Hendrycks et al., 2021a; Rein et al., 2024; Hendrycks et al., 2021b). The results are contained in Table 2.

Model	Llama-3.1-405B (Instruct)	Llama-3.1-405B (Instruct + ICO)	Diff.
MMLU (5-shot)	87.24%	87.71%	+0.47%
GPQA (0-shot, CoT)	51.11%	50.89%	-0.22%
MATH (0-shot, CoT)	73.76%	74.04%	+0.28%

Table 2: Evaluations of the strong model without and with in-context optimization (denoted ICO).

While GPQA performance decreased slightly, we find that both MMLU and MATH improve with this method, despite the weak models being significantly worse at these tasks than the strong model. Recall that the strong model does not have any generations sampled from its own policy unlike OPRO: this furthermore highlights this interesting result, showing that models can improve at inference time even with very weak supervision, as long as the in-context exemplars for optimization are contrasted.

4 Discussion

In this work, we introduced an approach to boost the reasoning capabilities of strong models purely in-context, using generations from weak models and their reward scores assigned by a language model judge. Surprisingly, we find that this indeed improves the strong model's performance, despite not optimizing at any point with respect to the strong model's generations. Furthermore, the weak models are validated as being differentiated on the basis of their synthetic data injected with filler tokens, yielding different performances on MMLU. This being said, we recognize a few limitations and areas for future work to extend our exploration:

More reliable judges. Our approach of studying reasoning efficiency through the number of steps (and correspondingly, the number of filler tokens) provides a unique lens to quantifying the quality of model reasoning. That being said, while we find this to correlate with downstream performance improvement, there are notions of reward in reasoning (based on correctness) defined at both the sample-level and at the CoT step level (Lightman et al., 2023; Ma et al., 2023). Another interesting notion introduced recently is that of a human-aware loss function (HALO), corresponding to system 2 reasoning with methodical, deliberate thinking (Ethayarajh et al., 2024); this work shows that inductive biases may be "selected" based on choice of HALO. We look to study in the future how performing in-context optimization over HALO reward functions improves human reasoning.

Preference optimization from paired reasoning generations. In order to further align both weak and strong models for reasoning by contrasting the quality of reasoning chains, we can rank the candidate generations (the three weak model responses and the strong model response) based on their scores by the scorer LLM. Based on this, one could either perform preference optimization with DPO (Rafailov et al., 2023) either with a Plackett-Luce objective or by forming paired data based on the ranks. We note that this approach is akin to rejection sampling (Touvron et al., 2023; Bai et al., 2022), given the best response is often significantly better (coming from the strong model). That being said, this provides a unique lens to explore distillation (training a Gemma-2 2B-it model on these synthetic preferences) and true weak-to-strong learning by preference fine-tuning the strong model on offline reasoning preferences.

Iterative in-context optimization. Works such as Self-Refine and Reflexion (Madaan et al., 2023; Shinn et al., 2023) perform in-context refinement for several iterations, though Self-Refine notes diminishing returns with this procedure. Our work only involves a single iteration; we do not resample generations, nor augment the policy which generates them, at any point in the algorithm's execution. Exploring our algorithm for multi-iteration could reveal interesting findings about how much test-time improvement is possible from weak supervision.

Acknowledgements

We would like to thank Xinyun Chen for helpful insights and suggestions on optimization with LLMs.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022. URL https://arxiv.org/abs/2212.08073.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4971–5012. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/burns24b.html.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12634–12651. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ethayarajh24a.html.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens, 2024. URL https://arxiv.org/abs/2310.02226.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

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, 2021b. URL https://arxiv.org/abs/2103.03874.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Qianli Ma, Haotian Zhou, Tingkai Liu, Jianbo Yuan, Pengfei Liu, Yang You, and Hongxia Yang. Let's reward step by step: Step-level reward model as the navigators for reasoning, 2023. URL https://arxiv.org/abs/2310.10080.

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. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf.

Jacob Pfau, William Merrill, and Samuel R. Bowman. Let's think dot by dot: Hidden computation in transformer language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=NikbrdtYvG.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

Keshav Ramji, Young-Suk Lee, Ramón Fernandez Astudillo, Md Arafat Sultan, Tahira Naseem, Asim Munawar, Radu Florian, and Salim Roukos. Self-refinement of language models from external proxy metrics feedback, 2024. URL https://arxiv.org/abs/2403.00827.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Yacmpz84TH.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision, 2023. URL https://arxiv.org/abs/2305.03047.

Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinhong Zhou, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Salmon: Self-alignment with instructable reward models, 2024. URL https://arxiv.org/abs/2310.05910.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric

Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

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. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=ehfRiF0R3a.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=yzkSU5zdwD. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. An empirical analysis of compute-optimal inference for problem-solving with language models, 2024. URL https://arxiv.org/abs/2408.00724.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=Bb4VGOWELI.

Yuqing Yang, Yan Ma, and Pengfei Liu. Weak-to-strong reasoning, 2024b. URL https://arxiv.org/abs/2407.13647.

Eric Zelikman, Georges Raif Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah Goodman. Quiet-STar: Language models can teach themselves to think before speaking. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=oRXPiSOGH9.

Tianjun Zhang, Aman Madaan, Luyu Gao, Steven Zheng, Swaroop Mishra, Yiming Yang, Niket Tandon, and Uri Alon. In-context principle learning from mistakes, 2024. URL https://arxiv.org/abs/2402.05403.

A Synthetic Reasoning Data Generation

As noted, we generated synthetic reasoning data, injected with filler tokens interleaved between sentences; this is akin to the introductions of "<|startofthought|>" and "<|endofthought|>" tokens in (Zelikman et al., 2024). To apply the Self-Instruct algorithm (Wang et al., 2023), rather than identifying classification tasks, we instead use the LLM as a classifier to determine which problems (model-generated instructions) are *"reasoning problems"* that could yield multi-step / chain-of-thought rationales. Then, for each *reasoning-inducing instruction*, we generate 5 sample instances (input-output pairs), such that for each input, we generate 3 outputs. This is included in the instruction to the model, and demonstrated by few-shot exemplars across a wide variety of domain. The resulting generations are then parsed and partitioned into three mixtures (one for each output number) for our weak models to be fine-tuned on.

A.1 Reasoning Classification Exemplars

Can the following task be regarded as a reasoning task?
Task: Given my personality and the job, tell me if I would be suitable. Is it reasoning? Yes
Task: Convert the following text to uppercase letters. Is it reasoning? No
Task: Give me an example of a time when you had to use your sense of humor. Is it reasoning? Yes
Task: Provide the numerical value of pi up to 10 decimal places. Is it reasoning? No
Task: Fact checking - tell me if the statement is true, false, or unknown, based on your knowledge and common sense. Is it reasoning? Yes
Task: Return the SSN number for the person. Is it reasoning? No
Task: Detect if the Reddit thread contains hate speech. Is it reasoning? Yes
Task: Analyze the sentences below to identify biases. Is it reasoning? Yes
Task: Retrieve the official website URL of the World Health Organization. Is it reasoning? No
Task: Select the longest sentence in terms of the number of words in the paragraph, output the sentence index. Is it reasoning? Yes
Task: Find out the toxic word or phrase in the sentence. Is it reasoning? Yes
Task: You are provided with a news article, and you need to identify all the categories that this article belongs to. Possible categories include: Music, Sports, Politics, Tech, Finance, Basketball, Soccer, Tennis, Entertainment, Digital Game, World News. Output its categories one by one, seperated by comma. Is it reasoning? Yes
Task: State the boiling point of water at sea level in Kelvin. Is it reasoning? No
Task: Select the oldest person from the list. Is it reasoning? Yes
Task: Find the four smallest perfect numbers. Is it reasoning? Yes
Task: Does the information in the document supports the claim? You can answer "Support" or "Unsupport". Is it reasoning? Yes
Task: Return the default port number for HTTP. Is it reasoning? No
Task: Create a detailed budget for the given hypothetical trip. Is it reasoning? Yes
Task: Provide the chemical symbol for gold, carbon, and oxygen. Is it reasoning? No
Task: Explain the following idiom to me, and try to give me some examples. Is it reasoning? Yes
Task: Is there anything I can eat for a breakfast that doesn't include eggs, yet includes protein, and has roughly 700-1000 calories? Is it reasoning? Yes
Task: Decide whether the syllogism is logically sound. Is it reasoning? Yes
Task: Provide the RGB hexcode for the color navy blue. Is it reasoning? No
Task: How can individuals and organizations reduce unconscious bias? Is it reasoning? Yes
Task: What are some things you can do to de-stress? Is it reasoning? Yes
Task: Find out the largest one from a set of numbers. Output the number directly. Is it reasoning? Yes
Task: Provide the capital city of Canada. Is it reasoning? No
Task: Replace the <mask> token in the text with proper words that are consistent with the context. You can use multiple words for each <mask> token. Is it reasoning? Yes
Task: Write a program to compute the sum of integers from k to n. Is it reasoning? Yes

A.2 Prompt for Synthetic Sample Generation with Filler Tokens

"'Come up with examples for the following reasoning tasks. Try to generate multiple examples when possible. If the task doesn't require additional input, you can generate the output directly. For a given example, produce a filler token '\t' after each step in your response. When you have come up with an example input (when one is required), or determined that an input is not required, give 3 possible outputs, with output 2 requiring fewer steps to solve than output 1, and output 3 requiring fewer steps to solve than outputs 1 and 2, while still arriving at the same final answer.

Task: Suggest a better and more professional rephrasing of the following sentence. Example 1 Sentence: This house is surprisingly not constructed very well, and you probably need more money to fix it after you buy it. If you ask me, I would suggest you to consider other candidates. Output 1: The original sentences are informal and contain redundant phrases, making them less professional. \t Phrases like "surprisingly not constructed very well" can be tightened for clarity. \t Additionally, "If you ask me, I would suggest you to consider other candidates" is wordy and can be made more direct. \t Thus, a rephrased version of the sentence is: This house is poorly constructed and will likely require additional funds for repairs after purchase. I recommend considering other properties. \t Output 2: The original sentences are informal and contain redundant phrases, making them less professional. \t Phrases like "surprisingly not constructed very well" can be tightened for clarity. \t Additionally, "If you ask me, I would suggest you to consider other candidates" is wordy and can be made more direct. \t Thus, a rephrased version of the sentence is: This house is poorly constructed and will likely require additional funds for repairs after purchase. I recommend considering other properties. \t Output 3: Step 1: Streamline the sentence by eliminating unnecessary words and phrases. \t Step 2: Rephrase for clarity and professionalism. \t Rephrased sentence: This house is poorly constructed and will likely require additional funds for repairs after purchase. I recommend considering other properties. \t

Task: Write a Python program to compute the sum of integers from k to n. Output 1: This can be done by looping over the numbers from k to n, inclusive, and adding their value to a variable which stores the running sum. F̑irst, initialize a variable, sum, to 0. T̑hen, looping over the range(k, n+1) covers the range of values from k to n, inclusive. F̑or each value in that range, add it to sum, using the += operator. F̑inally, return the variable sum. T̑his is represented with Python syntax as follows: def sum(k, n): sum = 0 for i in range(k, n+1): sum += i return sum
Output 2: Step 1: Utilize Python's built-in sum() function combined with range() to calculate the sum. S̑tep 2: Return the computed sum directly. T̑his is represented with Python syntax as follows: def sum(k, n): return sum(range(k, n+1))
Output 3: Step 1: Apply the arithmetic series formula to compute the sum in a single step. S̑tep 2: Return the calculated sum. T̑his is represented with Python syntax as follows: def sum(k, n): return (n - k + 1) * (k + n) // 2
<more exemplars, each with 3 outputs>

Task:"'

B Implicit Objective Functions

Following from the work of OPRO (Yang et al., 2024a), we use another large language model to act as the scorer to assess the quality of generations. However, there are several means through which we could score quality, depending on the metrics and principles of interest to the user, which guides the model's refinement / optimization process (Sun et al., 2023, 2024; Ramji et al., 2024; Zhang et al., 2024). That is, one could choose to directly select metrics which assign real-valued scores to use in place of the Scorer LLM, or use a model trained on such lens which might have a better human-centered notion of preferred reasoning pathways. This leads to the notion of human-aware loss functions (HALOs), introduced in (Ethayarajh et al., 2024). The premise of HALOs is fundamentally built on prospect theory, on the notions of *human value* and utility; a HALO is defined as a function such that given the functional form for human value v, a reward function for the model r_θ, and a reference distribution Q, the function is linear is $v(r_\theta(x, y) - E_Q[r_\theta(x, y')])$. Notably, the work shows that direct preference optimization (DPO; Rafailov et al. 2023) is a HALO; we thus use Llama-3.1-70B-instruct as our Scorer LLM, as it has been trained with an iterative rejection sampling + DPO pipeline (Dubey et al., 2024).

An aspect we are then curious about is whether a model that has undergone preference optimization with a fixed HALO function f_{HALO} can assign rewards to new generations that approximate that HALO. If so, this would provide a new dimension to our work, as it is important to examine the intersection of preference (a human-centric notion) with reasoning (an in-demand application of foundation models). Doing so by designing a mathematically grounded loss formulation could unlock new doors to boosting reasoning with cheaper feedback signals, as opposed to leveraging the same model for sampling with test-time compute.