Team Elicitors Project Report

Jakob Bjorner
jbjorner3@gatech.edu

Loránd Cheng
lorand@gatech.edu

Milad Heydari
miladgheydari@gatech.edu

Atharva Mete
amete7@gatech.edu

April 8, 2025

Abstract: Generating step-by-step rationales, or "chain-of-thought" reasoning, enhances the performance of language models (LMs) on complex tasks such as mathematical problem-solving and commonsense question-answering. This process has been extensively studied in frameworks like Self-Taught Reasoner (STaR) and its generalization, Quiet-STaR, which emphasize improving rationale generation through Reinforcement Learning on sampled generations. In this work, we propose novel methods to advance task-relevant rationale generation within these paradigms. For the STaR framework, we utilize iterative reinforcement through Direct Preference Optimization (DPO) over generated rationales, allowing fine-tuning of LMs to produce more accurate and coherent reasoning. In the Quiet-STaR setting, we introduce meta-prompt tokens at the start of rationales, designed to introduce diversity in generated reasoning paths while maintaining relevance to the task. Our methods address key challenges such as optimizing rationale quality and encouraging broader exploration of reasoning strategies. Experimental results demonstrate that DPO does not significantly improve over the STaR baseline task of MMLU. Our meta prompt experiments led us to evaluation results which cause us to question the original results of Quiet-STaR and hint at future directions for improvement. [1] [2] [3] [4] [5] [6]

1 Introduction

Human decision-making often relies on extended chains of thought, enabling nuanced understanding and reasoning. Similarly, explicit intermediate reasoning, or "rationales," have been shown to significantly improve the performance of large language models (LLMs) across diverse tasks such as mathematical reasoning, commonsense question-answering, and code evaluation. Initial methods [12] employed specific structured prompting techniques to elicit steps of reasoning that would allow the language model to bootstrap its own generated context to solve more complex tasks. In contrast, [14] uses the SFT objective on carefully filtered question-answer pairs to improve reasoning capabilities through a "ratinalization" process. Methods like [14] and [11] have focused on solving individual tasks or predefined sets of tasks. These works rely on carefully curated datasets to provide either specific reasoning tasks or in some cases, the reasoning itself. Extending STaR to a general setting, [15] trains an LM to generate reasoning that helps it infer future text from a large internet text corpus, using REINFORCE to improve the thought generation policy.

In this work, we focus on the generation of high-quality, task-relevant rationales, where the quality of a rationale is defined by its ability to increase the likelihood of producing a correct answer for a given task. A similar case can be made for mathematical problem-solving tasks. [13] found that on math problems, LLMs build and update a computational graph with each forward pass and

[1] WandB for Star experiments https://wandb.ai/jakobbbjorner/star?nw=nwuserjakobbbjorner
[2] WandB for quiet-star experiments https://wandb.ai/jakobbbjorner/quiet-star-open-web-math?nw=nwuserjakobbbjorner
[3] Github for quiet-star results https://github.com/lorandcheng/quiet-star/tree/jakob
[4] Github for star results https://github.com/jakob-bjorner/star
[5] https://wandb.ai/yixiong_hao-georgia-institute-of-technology
[6] Summary Video Link: https://drive.google.com/file/d/1yjFLnmKIf90WRJmla4HSIztIn0YK5niq/view

make mistakes by prematurely trying to compute a problem parameter that can't yet be computed. Encouraging models to systematically reason about diverse problem parameters (while training with an RL objective) could mitigate these errors as the model learns to accurately identify the next parameter to compute.

Our project originally began with the goal of specifically improving model reasoning in low-resource languages. Our initial experiments showed that, when models were prompted in English for reasoning based tasks on MMLU, they greatly outperformed compared to when prompted in a low-resource language (in this case Yoruba). We also found that including prompts that asked the model to translate its reasoning to English before then returning the answer resulted in slight but noticeable average improvement which motivated our work of believe there are better 'reasoning' modes which we could trigger with meta prompting or DPO.

We ended up not be able to pursue to low-resource language idea in full because we had to limit ourselves to English language only models and could not use large, multilingual models due to compute and compatibility limitations. Despite that, the slightly improved reasoning when prompted to think in English served as our basis for wanting to explore if similar improvements were possible on general English based reasoning tasks that require special types (mathmatical, ect) of reasoning.

Motivated by these initial findings, our work focuses on developing methods to generate better, task-relevant rationales for LLMs in two key frameworks: the Self-Taught Reasoner (STaR) and its generalization, Quiet-STaR. Our primary objectives are to improve rationale quality and diversity while maintaining alignment with task requirements. For STaR, we propose an iterative approach using Direct Preference Optimization (DPO) to refine rationale generation. In Quiet-STaR, we introduce meta-prompt tokens at the start of rationales to encourage diverse and contextually appropriate reasoning.

2 Related Work

While LLMs pretrained on a text generation objective have demonstrated an impressive level of fluency, these objectives do not explicitly teach language models to follow logical reasoning steps in their output. Considerable effort has gone into studying methods for of improving the logical reasoning of language models, including both explicit and implicit ("quiet") methods of harnessing rationales.

2.1 Training Language Models to Reason

One direction to train LMs to reason or improve their reasoning is training the LM on mined reasoning traces or reasoning-like data [1], [11]. This shows some improvements but is difficult to scale due to the need for manual annotation, which is expensive and sensitive to the capability of the annotators. Another direction for teaching reasoning relies on a language model's own generated reasoning, which can be seen as building on a large body of literature on self-play [8], [6].

STaR: Self-Taught Reasoner iteratively bootstraps a model's capacity for generating rationales (step-by-step explanations) by using a small set of example rationales to generate rationales for a larger dataset. The model then refines itself by fine-tuning on the correctly generated rationales sampled from its own output, including those generated by "rationalization," where for an incorrect response the model is prompted with the correct answer to generate the rationale. Follow-ups to STaR includes [10] which incorporates "process-based" supervision to filter out incorrect reasoning traces, as well as [4] that trains a verifier to guide the thought generation.

Quiet-STaR: Quiet-STaR [15] is motivated by the lack of flexibility in the training data required for methods like STaR, which define the reward of a rationale as the probability of getting the correct answer. This relies on the ability to determine correct answers, or a Q/A structured dataset. Instead, Quiet-STaR defines the reward of a rationale as the improvement in log-likelihood of the next few tokens in any text data. The language model is trained with reinforcement learning (REINFORCE) to generate internal "thoughts" or rationales at each token in a sequence that are most helpful in predicting the next series of tokens. The model learns to generate these rationales in parallel, using learnable tokens to mark thought boundaries and a mixing head to combine predictions with and without thoughts. Unlike explicit rationale methods like Chain-of-Thought reasoning, Quiet-STaR's thoughts are implicit and not directly visible in the model output. In addition, no manual prompt tuning is necessary, since the training objective is entirely captured by next-token prediction and RL.

2.2 Math Reasoning

LLMs have demonstrated impressive capabilities in mathematical problem-solving, particularly on grade-school-level questions. The GSM8K [2] benchmark has been widely used to assess LLMs' performance on such tasks, with models showing substantial improvements in recent years. In [13] the authors create a synthetic dataset of math problems to control for data contamination and explore the models' reasoning processes. They use probing techniques to examine the models' internal states and discovered that LLMs usually make three kinds of mistakes: incorrectly identifying necessary parameters, computing parameters prematurely (before all their dependent parameters have been calculated), and failing to recognize parameter dependencies. Following these observations, we took the suggested prompts in the paper as our meta-prompts in the hope of reducing these mistakes in the language model's reasoning.

2.3 Preference Optimization

Direct preference optimization (DPO) [7] provides a more straightforward approach to aligning language models with human preferences compared to traditional methods that require complex reward modeling and multiple training stages. DPO directly optimizes the model's policy against a reference model using a simple binary cross-entropy objective. The method leverages a constrained optimization problem that maps preference data directly into the model's training objective, enabling more efficient and stable alignment of language models with human-defined preferences.

3 Methods

In this section, we detail the two primary approaches we pursued with the goal of improving the rationales generated by language models.

First, motivated by the recent success of DPO and RLHF for language model reward tuning, we sought to leverage correct and incorrect answers to questions sampled from the model and reinforce cases that lead to the correct answer with DPO - doing so allows us to use negative examples in such a way that is more stable than direct unlearning methods. Furthermore, beyond DPO on fixed preference datasets, our preference data is sampled from the model directly, so we are only reinforcing already existing behaviors reducing the off policy gap sometimes common with DPO. In addition, we tried to use another model to compare two possible rationals generated from the model that lead to the same answer, so we can still determine the better one and reinforce this behaviors.

Second, we observed that while Quiet-STaR is an elegant method of improving rationale generation on unstructured text datasets, if the initial rationales that are sampled are too far from the desired rationales, there is no mechanism to improve exploration to discover the better reasoning paths. Combined with the insight from [13] that LLMs tend to make systematic errors in their logical reasoning, we introduce the idea of a meta-prompt for thought generation that more easily allows the model to explore useful logical reasoning paths that are difficult to discover naturally.

3.1 Iterative Direct Preference Optimization for Model Reasoning trained under STaR setting

Using a LLama 7b-instruct model that generates step-by-step reasoning, we evaluated the model on an MMLU (Massive Multitask Language Understanding) dataset [3] which uses multiple-choice questions to assess knowledge and reasoning performance across areas like STEM, Math, humanities, and social sciences.

An example of a very basic question and the model's output are shown below:

```
Which of these is a fish?
```

```
A) sea horse
B) sea cow
C) sea snake
D) sea lion
------------------------------
To determine which of these is a fish, let's examine each option:
A) Sea horse: Sea horses are marine animals that belong to the family Hippocampus. They are fish, but not
what most people think of as "fish" because they are part of a specific, distinct group.
B) Sea cow: Sea cows, also known as manatees, are large, slow-moving herbivores that live in shallow, warm
freshwater and marine waters. They are mammals, not fish.
C) Sea snake: While they live in the ocean and are adapted to an aquatic environment, sea snakes are
actually marine reptiles.
D) Sea lion: Sea lions are pinnipeds, a type of fin-footed, semi-aquatic marine mammal. They are large
mammals  and not typically considered fish.

Considering these descriptions, only sea horses (A) are something that may actually be considered fish.
 Answer: A
```

In an attempt to improve the model's performance (accuracy % on MC questions) on this task via improving its reasoning abilities such that it leads to better answers, we performed the following experiment:

1. For every question within our training split of the MMLU dataset (11,000 questions), using the LLama 7b-instruct model with temperature 1, we sampled 3 different outputs and saved these to a database/csv. We also recorded the original question as well as the correct answer and the model's output answer.
2. Using our sampled dataset, we performed multiple steps of offline training using Direct Policy Optimization [7]. This should train our model to generate with higher probability the reasoning that we train it to prefer, and we aim to elicit/prefer more logical and correct reasoning that leads to better results with our approach. Since we do not have access to human labelers to pick the best reasoning/answer like traditional RLHF, we aimed to pick preferred answers based on the assumptions:
   A) If two different reasoning outputs on the same question lead to different answers and one is correct and one is wrong, we prefer the reasoning that led to the correct answer.
   B) If two different reasoning outputs on the same question lead to the same answer (or both incorrect answers), we used another Llama instruct model to determine which reasoning is better. This is to avoid incorrect/simple reasoning that leads to the correct answer.
3. Repeat steps 1 and 2 for multiple iterations as is done in Star [14].

```python
def evaluate_reasoning(question, reasoning1, reasoning2):
    prompt = f"""Which of the following reasoning approaches is more logical in
    answering the given question?
The better reasoning should:
1. Follow a clear and logical sequence of steps
2. Use valid assumptions and premises
3. Draw conclusions that directly follow from the provided evidence
Question:
{question}

Reasoning 1:
'''
{reasoning1}
'''

Reasoning 2:
'''
{reasoning2}
'''

Respond with either "1" or "2" to indicate which reasoning is more logical"""
    choice = genOutput(prompt)
    return choice
```

Figure 1: Eval prompt

To prevent the ordering of the answers from impacting which answer is preferred by the Llama model, we ran the above twice with the ordering of the input switched as well (which is reasoning 1 and which is reasoning 2).

3.2 Quiet-STaR with meta-thought tokens

As demonstrated by [13], LLMs often make systematic errors on logical reasoning problems by prematurely attempting to compute specific problem parameters. Quiet-STaR, while useful for reinforcing useful implicit rationales, is limited in its ability to overcome these systematic errors due to its limited exploration. To address this, we propose encouraging the model to consider diverse problem parameters and training it using a REINFORCE objective over such generated thoughts. This approach aims to guide the model not only to focus its reasoning on relevant problem parameters but also to effectively discern the most appropriate next parameter to consider.

To implement this, we extend the Quiet-STaR framework by incorporating meta-thought tokens into the rationale generation process. After each start-of-thought token, we insert tokenized meta-prompts to condition the model's rationale on a prompt that explicitly encourages exploration in its reasoning. Specifically, we use a list of 5 meta-thought prompts to promote diversity over problem parameters, taken from [13]:

1. identifying useful intermediate quantities
2. considering relevant problem parameters
3. exploring relationships between variables
4. next parameter to compute is
5. necessary variable to be known is

We appropriately modify the kv cache, positional embedding and the attention mask to account for the expanded implicit rationale. We also try another set of experiments where we replace the original start-of-thought token with the first token in a particular meta-prompt.

4 Experimental Results

4.1 DPO for Model Reasoning

Sadly, our trained model did not show significant improvements over the baseline Llama-Instruct 7b model on the MMLU accuracy and maintained a similar accuracy to the baseline (60-70%).

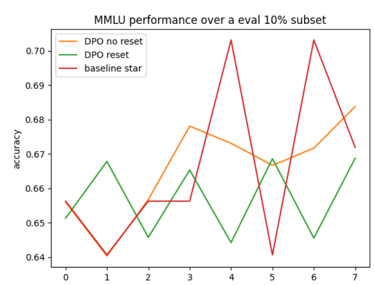

Figure 2: Star in MMLU setting at its best improves by 5% over baseline performance from the LLama 3.1 8B instruction tuned model. Our variation of Star where we instead used DPO has little improvement over the baseline performance. And upon seeing the cyclic behavior at every Star iteration, we came up with an alternative implementation, which doesn't reset the policy on every iteration to llama 7b instruct weights, but rather maintains the policy for successive iterations relying on the regularization provided by the reference policy in DPO. Even expanding this to further iterations, our DPO + reset and baseline star seem to both cap out at 70%

Upon investigation, we believe this is for two main reasons:

1. When using an 'evaluator' language model to pick between two rationals (figure 2), we found that, if we switch the order of the inputs/reasoning, it switches which chain of thought/reasoning the model thinks is better, so it is clearly being impacted by the order and not very good at consistently determining what is better based only on the content. Switching the order often flips which one the model things is better, so when considering both orderings, we end up with a 50-50 win rate of which prompt is better and thus do not get any useful signal. Therefore, the chains of thought on responses that lead to the same answer are not different enough to be able to serve as a signal for DPO and we were not able to use it in training. We do see consistent picking of the better chain of thought when one answer is wrong and one is correct, but this could also be trivially picked by simply picking chains of thought that lead to the correct answer.

If we manually put in dumb reasoning that leads to the correct answer: "The answer is A because A is my favorite letter," we do see the model consistently pick the other rationale even though both lead to the correct answer ven when the order of input is switched. However, we never see a case where the rationales are different enough to trigger this from the rationales sampled from our model, thus we conclude this is a data and not a code issue.

Note: This investigation was done in Google Colab seperate from our main code base

Therefore, conclude that our model is not sampling/outputting rationales that are different enough, when leading to the same answer, to produce a useful signal for DPO that would allow us to differentiate good rationales, even with high temperatures. Manual, human inspection of the rationales backs up this hypothesis as many of the chains of thought are fairly similar and not clearly better than others.

2. To start with a model that already generated a chain of thought/reasoning such that we could fine-tune it, we had to start with a fairly advanced baseline model. The Llama instruct model was already achieving 70% accuracy on the MMLU and may have already been trained on a similar type of reasoning, improving DPO/training or Supervised Fine-Tuning, making it hard for us to gain any competitive edge with further training. The fact that we did not regress the performance (and did very slightly improve) supports this claim.

4.2 Quiet-STaR with meta-thought tokens

In the Quiet-STaR setting, we primarily studied two directions of improvement:

1. Implementing meta-thought prompts to improve rationale generation
2. Reducing the variance of the REINFORCE algorithm that Quiet-STaR uses.

Our evaluation for these questions follow Quiet-STaR and use GSM8K [2] and CommonSenseQA [9] as the evaluation benchmarks for logical reasoning.

4.2.1 Meta-Thought Prompting

In the initial set of experiments, we recreated the baseline and used 1 percent of the evaluation set from the original Quiet-STaR paper to establish whether introducing meta-prompts could help with rationale generation. We evaluated only on a relatively small subset of the dataset due to compute requirements (full evaluation requires a full node of 8 H100s for two hours for 100 steps). The number of training points used was kept the same compared to quiet-STAR (1000 example phrases from the OpenWebMath dataset), and the training dataset used is Open Web Math [5].

In Quiet-STaR, the start of thought token is initialized with "—", indicating a natural pause in text that would allow for a cohesive thought. This start of thought token is then learned throughout the training process using an embedding scaling of 100 on the gradients. By studying the effects of these components, we sought to understand how sensitive Quiet-STaR is to the thinking tokens and whether scaling the gradients for this component was important. Overall, we found that it was quite sensitive and the 100-step evaluation presented in the paper is somewhat misleading.

Our first attempt at implementing our meta-prompt idea led to interesting but counter-intuitive results. The first implementation we tried corresponded to changing the start of thought (SoT) token embedding and reducing the embedding gradient scaling added in the original paper from 100 to 1.

In Figure 3, we compare the baseline with (1) an alternate (single) SoT token with reduced embedding scaling, (2) randomly sampled meta thoughts from the predefined list, and (3) the best performing single prompt. The first thing to note is that in the original paper, the Quiet-STaR authors only include the first 100 training steps, which masks the decline in performance as training progresses further. This could imply that the benefits from this method saturate quickly and long-term benefits are not to be expected. From the comparison with our implemented methods, we observe a few interesting results. First, the Quiet-STaR method is sensitive to the initialization of the SoT token, as shown in the GSM8K performance comparison. The best token we found, sampled from the start of our meta-thoughts, is ”considering”. After 100 steps, this prompt results in almost 50% improvement on GSM8K, but exhibits instability in training. Second, the full meta-prompts show much slower improvement, though the best performing prompt (”necessary variable to be known is”) exhibits an upward trend throughout even past where other methods have plateaued or collapsed. We attribute the difference between the randomly selected and a single full meta-thought prompt to the instability in training when the prompts are constantly being swapped out at every step. In the full meta-prompt runs, we set the embedding scaling to 100 like the original approach.

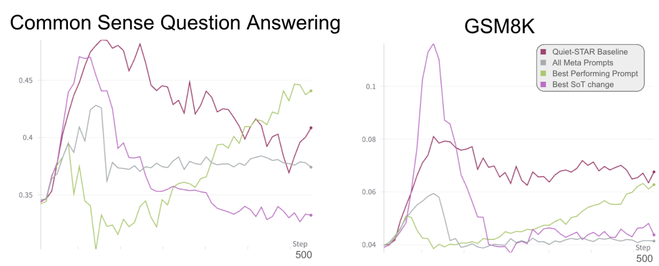

Figure 3: Comparison of baseline with alternate SoT token and full meta-prompts, selected randomly and best single prompt.

We also conducted limited evaluations on a different, larger subset of the data (5%) on the best meta-thought from our initial experiments. We trained 3 seeds of both the baseline and the meta-thought version for 500 training steps, though we also doubled the batch size so that the same 500 steps from before account for double the training data. In addition, we doubled the sequence length from 128 to 256 to better match the baseline.

In figure 4, we observe similar trends as before in the first half of training: changing the start of thought prompt leads to a noticeable increase on GSM8K but proves to be neglible or even harmful on CSQA. Interestingly, compared to before, the meta-thought prompting does not collapse on GSM8K as it did earlier. We suspect this may have something to do with the increased sequence length but we cannot draw definitive conclusions without further experiments.

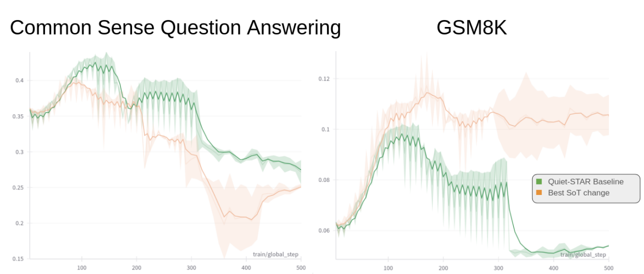

Figure 4: Comparing best meta thought with baseline on larger eval set. This was done to verify that our results would generalize beyond our small initial eval sample.

Observing results from 4, we see our results generalize beyond 1 percent evaluation set. Which reinforces the questions raised, that the start of thought token, and the embedding scaling chosen in the original quiet-STAR paper may not be conducive for effective thought exploration, and therefore may hinder the performance.

4.2.2 Variance Reduction

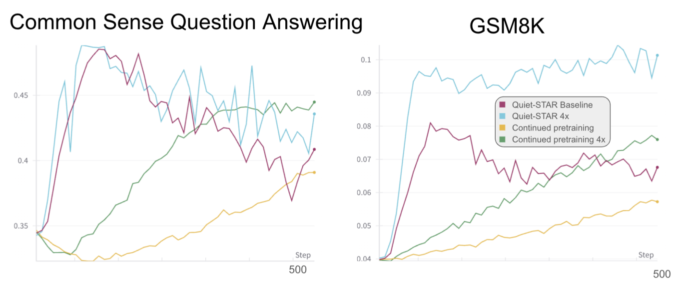

Figure 5: The 4x refers to the fact that the effective batch size was increased by 4 times. We conducted additional experiments to see how far this improvement would continue if left training, but the continued pre-training plateaus shortly after 500 steps, so we truncate here. Something important to note about this setting is that Continued pretraing doesn't improve the models performance if you increase the dataset size. We are unsure of the cause of this. (results are reported on 1 pct of eval set)

During discussions, our team thought that the variance reduction technique to stabilize reinforce was weak in the original quiet-STAR implementation. We considered alternatives to reinforce, but ultimately couldn't implement these ideas due to time limitations further exacerbated by the difficulty in working with the existing code base. What we were able to test was whether or not their implementation would benefit from reduced variance in their gradient estimate. To do this, we simply increased the batch size thought additional gradient accumulation steps. What we found was, not only that the quiet-star's method improved from this increase in effective batch size, but also that the continued pre-training baseline presented in quiet-STAR also massively benefits. It is important to note that the same amount of data is seen during training even when a higher effective batch size is used because only 1000 examples are used. Also, the step counts do not show the full picture. The wall clock training and inference time for eval is hidden, but for the quiet-STAR method, we observed a more than 14 x slow down when compared to the continuous pre training of equivalent effective batch size.

Another counter intuitive model property of the star models is that on the eval task, the COT performance of the 'base' model has increased. This is without the production of thoughts to nearly match that of the performance of the thought augmented model predictions. This needs to be investigated further through future work, and may be related to some form of distillation from the thought model to the base model occurring during training.

5 Discussion and Analysis

Our approach of improving rational generation through reinforcement using DPO rather than REINFORCE (without the use of negative examples) as is done in star [14] for the MMLU setting, was strong in the theoretical inspiration to leverage negative examples, but fell short, due to what we believe is a sample efficiency problem. Standard DPO requires significantly more examples (order of 60k) to generalize, which weren't available for MMLU which is limited to only 11k questions.

The approach we took to improve Quiet-STaR's [15] performance of reducing variance, and introducing meta prompts led to observations which support different directions for future investigation than our original intuitions would have suggested. The performance gain created by variance reduction observed in 5 over the baseline quiet-star method was interesting, but what was more shocking was the change seen in the continuous pre-training baseline when the same effective batch size increase was tested. This observation leads one to broadly question the results reported in Quiet-STaR. On top of this our attempts at finding effective meta prompts led us to discovering that changing the start of thought token hyper parameters caused improvement in eval results much higher than we would have anticipated given our initial thoughts that the single start of thought token wouldn't allow for as much generation diversity, and would still be prone to all the systematic errors found in [13].

From our experiments, the future directions should be to more thoroughly investigate the performance claims of Quiet-STaR, only in the light of this analysis can more experiments into changing the start of thought tokens or incorporating meta thoughts make sense.

6 Conclusion

Deep reinforcement learning for language models through traditional RLHF or RLAIF has been and will continue to be an important part of the post training pipeline used on commercially deployed language models. Our work builds on a body of works applying reinforcement learning beyond preference data, specifically we target settings where rational can be applied at solving problems. Through our experiments we raise critical concerns on the claims presented in Quiet-STaR, a seminal work in the field of reinforced rational generation. Orthogonal to this we find limitations in low data regimes for applying preference optimization style algorithms.

Acknowledgments

Yixiong Hao and Ayush Panda for their contributions in theoretical analysis and code developement.

References

[1] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei. Scaling instruction-finetuned language models, 2022. URL `https://arxiv.org/abs/2210.11416`.

[2] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems, 2021. URL `https://arxiv.org/abs/2110.14168`.

[3] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021. URL `https://arxiv.org/abs/2009.03300`.

[4] A. Hosseini, X. Yuan, N. Malkin, A. Courville, A. Sordoni, and R. Agarwal. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457v2*, 2024. URL `https://arxiv.org/abs/2402.06457v2`.

[5] K. Paster, M. D. Santos, Z. Azerbayev, and J. Ba. Openwebmath: An open dataset of high-quality mathematical web text. *arXiv preprint arXiv:2310.06786*, 2023.

[6] S. Polu and I. Sutskever. Generative language modeling for automated theorem proving, 2020. URL `https://arxiv.org/abs/2009.03393`.

[7] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023. URL `https://arxiv.org/abs/2305.18290`.

[8] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017. URL `https://arxiv.org/abs/1712.01815`.

[9] A. Talmor, J. Herzig, N. Lourie, and J. Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

[10] J. Uesato, N. Kushman, R. Kumar, F. Song, N. Siegel, L. Wang, A. Creswell, G. Irving, and I. Higgins. Solving math word problems with process- and outcome-based feedback, 2022. URL `https://arxiv.org/abs/2211.14275`.

[11] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners, 2022. URL `https://arxiv.org/abs/2109.01652`.

[12] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL `https://arxiv.org/abs/2201.11903`.

[13] T. Ye, Z. Xu, Y. Li, and Z. Allen-Zhu. Physics of language models: Part 2.1, grade-school math and the hidden reasoning process. *arXiv preprint arXiv:2407.20311v1*, 2024. URL `https://physics.allen-zhu.com/part-2-grade-school-math/part-2-1`.

[14] E. Zelikman, Y. Wu, J. Mu, and N. D. Goodman. Star: Self-taught reasoner bootstrapping reasoning with reasoning. *arXiv preprint arXiv:2203.14465v2*, 2022. URL `https://arxiv.org/abs/2203.14465v2`.

[15] E. Zelikman, G. Harik, Y. Shao, V. Jayasiri, N. Haber, and N. D. Goodman. Quiet-star: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629v2*, 2024. URL https://arxiv.org/abs/2403.09629.