Reproducibility Report Instructions for CSE 517

Jeremy Huang, Sihang Zeng, Aditya Akhileshwaran
University of Washington
{jeremyh1, zengsh, aakhiles}@uw.edu

Reproducibility Summary

In this study, we aim to reproduce the central claim of the paper "Chain of Preference Optimization: Improving Chain-of-Thought Reasoning in LLMs" [22]. Our motivation stems from CPO's promise of enhancing reasoning in LLMs while maintaining computational efficiency. The authors claim CPO achieves comparable or better performance than tree-of-thought (ToT) [21] while being 57.5 times faster during inference, with accuracy improvements up to 4.3% across question answering [11, 5, 20], fact verification [14, 1, 13], and arithmetic reasoning tasks [10]. We attempt to validate these claims across different model sizes and datasets and verify the effectiveness of leveraging preference information from non-optimal reasoning paths, as opposed to traditional chain-of-thought approaches [18] and direct preference optimization methods [12].

Our work attempts to reproduce CPO's core claims: improved reasoning performance, ToT-comparable results with lower latency, and superiority over SFT. We implement the complete methodology, including thought generation, evaluation, search and preference optimization, on smaller foundation models across question answering, fact verification, and arithmetic reasoning tasks. We also verified ablation studies by examining dispreferred thought selection, chain-level optimization and training data impacts while maintaining fidelity to the original hyperparameters where feasible.

We implemented the Chain of Preference Optimization method from scratch rather than using the authors' original codebase [23]. Our implementation includes all core components: preference data generation via Tree-of-Thought exploration [21], preference pair extraction, and LLM training with the CPO objective adapted from DPO [12]. For our experiments, we primarily used Google Colab with NVIDIA T4 and L4 GPUs, and A100 GPUsto speed them up.

Due to computational constraints, our implementation includes several practical adaptations. We used LLaMA 3.2-1B rather than the larger 7B and 13B models from the original paper [15], evaluated on a subset of the original datasets, and conducted a less extensive tree-of-thought exploration during preference generation. The time required for training and evaluation varies across different phases of the implementation, with detailed timing comparisons presented in our results section.

We were able to successfully evaluate the base model, the base model with CoT, as well as a supervised fine-tuned model without the CPO objective. Finally, we trained a model using the CPO objective described in the paper. Our experiments showed that our CPO implementation underperformed compared to the original paper's claims. The average accuracy improvement of CPO over CoT was -0.104 (a slight decrease), meaning our CPO implementation performed worse on average in terms of accuracy. However, we did confirm the efficiency claim: the average latency improvement of CPO over ToT was 115.734x, indicating that CPO is indeed significantly faster than ToT during inference, even exceeding the 57.5x speedup reported in the original paper.

These results suggest limitations in our implementation of the CPO approach or challenges in reproducing the paper's key innovations with our computational constraints, particularly regarding the accuracy improvements.

Our initial expectation was that training the CPO model would be compute intensive and much more complex than it turned out to be. Surprisingly, the model converged quickly, requiring just one epoch with the CPO objective.

The dataset preprocessing was much more difficult than expected. In fact, we omitted two datasets, the Fever [14] and Feverous [1] datasets, because we could not get them to work. Prompt engineering was also quite difficult. Because we were working with a very small model, the model was not good at following instructions compared to the larger models used in the original study [15]. Its lack of ability to follow instructions meant that it sometimes didn't do chain of thought, or didn't follow the requested extraction format.

Extraction was exceedingly difficult. We tried prompting the model for many different kinds of output for answer extraction, most of which were inconsistent. We landed on using boxed latex as the requested answer format because it was likely finetuned to do that, so this was a more consistent method. The models were also highly sensitive to the prompt, and performance could change drastically based on what prompt we gave.

We did not establish any communication with the original authors during our reproducibility study.

Introduction

This reproducibility report focuses on "Chain of Preference Optimization: Improving Chain-of-Thought Reasoning in LLMs" [22]. The paper introduces a novel method called Chain of Preference Optimization (CPO) to enhance the reasoning capabilities of Large Language Models (LLMs). The key innovation lies in leveraging the inherent preference information generated during tree-of-thought (ToT) [21] search to optimize chain-of-thought (CoT) [17] reasoning.

The original paper makes several important contributions: (1) identifying that non-optimal reasoning paths generated during tree search contain valuable preference information that can be used for training; (2) introducing a method to construct paired preference data at each reasoning step rather than only at the full-path level; and (3) demonstrating that chain-level optimization with these preference pairs will allow the model to achieve similar or better performance than ToT while maintaining the efficiency of CoT during inference.

This work is worthy of reproduction for several reasons. First, it presents a solution to the critical trade-off between reasoning quality and computational efficiency in LLMs - achieving an impressive 4.3% accuracy improvement over base models while being 57.5 times faster than ToT during inference. Second, it doesn't require external reward models, feedback or human annotations – making it valuable for practical, real-life applications that have resource constraints. Finally, the approach of learning from non-optimal reasoning paths represents a new learning paradigm that could potentially be applied to other aspects of LLM optimization.

Scope of Reproducibility

Our reproduction study focuses on validating the key claims of the CPO method for improving LLM reasoning capabilities. We chose this paper because it presents a novel approach to enhancing LLM reasoning that promises both improved performance and computational efficiency - two critical concerns in modern NLP research. We aim to make this report self-contained, providing sufficient context for readers who may not be familiar with the original paper.

The central contribution of the paper is a new fine-tuning method that leverages preference information from Tree-of-Thought (ToT) search to optimize Chain-of-Thought (CoT) reasoning. This method promises to achieve the quality of ToT's complex tree search while maintaining the efficiency of simple CoT inference. We focus on verifying these specific hypotheses from the original paper [22]:

1. Performance Improvement: CPO can improve LLM's reasoning ability without requiring additional human-annotated data, showing an average accuracy improvement of up to 4.3% compared to base models across various reasoning tasks.

2. Efficiency-Quality Balance: CPO achieves comparable or better performance than ToT [21] while being approximately 57.5 times faster during inference, effectively addressing the computational overhead issue of ToT.

3. Component Effectiveness: The per-step preference supervision approach in CPO [12] is more effective than full-path preference optimization or standard supervised fine-tuning (SFT) [2] for improving reasoning capabilities.

We evaluate these hypotheses across three types of reasoning tasks:

- Question Answering (using Bamboogle [11], 2WikiMultiHopQA [5], and HotpotQA [20] datasets)
- Fact Verification (using Vitaminc [13], FEVER [14], FEVEROUS [1] datasets)
- Arithmetic Reasoning (using SVAMP [10] dataset)

Our reproduction includes the implementation of the complete CPO methodology—thought generation, state evaluation, search and collection, and training with the CPO objective—as well as the baseline methods (CoT, ToT, and TS-SFT) for comparison. We also replicate key ablation studies to understand the impact of various components on the overall performance.

Methodology

Model Descriptions

The CPO method leverages the chain of preference throughts, preferences over thoughts generated at each reasoning step in ToT, to finetune the LLM, which enhances LLM's reasoning ability. This includes two steps: synthesizing the chain of preference thoughts and training the LLM with the CPO objective.

To synthesize the chain of preference thoughts, three parts are designed. First, for each input question x, the method generates k thoughts for the next reasoning step. Each thought z_i is a sequence of tokens, and the thoughts are generated by sampling from the LLM's output distribution. A state $s_i = [x, z_1, ..., z_i]$ is a sequence of thoughts so far. Second, an LLM is used to evaluate each state. The LLM is prompted to reason about the state and then classify it as either "correct", "likely", or "impossible". The classification result is used to assign a score to the state. Third, the method uses a search algorithm to select the n-best thoughts with the highest evaluation scores at each reasoning step, until a path leads to the final answer (Tree-of-thoughts). All thoughts within this path are marked as preferred thoughts, while their sibling thoughts are considered as dispreferred thoughts given their parent state. These preference pairs are used to train the LLM with the CPO objective.

The CPO objective is a loss function that encourages the LLM to assign higher probabilities to preferred thoughts z_i^w than to dispreferred thoughts z_i^l given their parent state s_{i-1}^w. The CPO objective is based on the DPO objective, which is a loss function that encourages the LLM to generate responses that are preferred by humans. The CPO objective is:

$$\mathcal{L}_i(\pi_\theta; \pi_{ref}) = -\log \sigma(\beta \log \frac{\pi_\theta(z_i^w | x, s_{i-1}^w)}{\pi_{ref}(z_i^w | x, s_{i-1}^w)} - \beta \log \frac{\pi_\theta(z_i^l | x, s_{i-1}^w)}{\pi_{ref}(z_i^l | x, s_{i-1}^w)}) \quad (1)$$

where π_{ref} is the base reference model and π_θ is the model to be optimized.

Their experiments were based on LLaMA2-7B and 13B [16] and Mistral-7B [8]. However, due to limited computational resources, we used LLaMA 3.2-1B-Instruct [3] to conduct the experiments.

Datasets

We used the same seven datasets as the original CPO paper [22], spanning three reasoning tasks.

For Question Answering (QA), we used Bamboogle [11], a dataset with 5,000 questions designed to test multi-hop reasoning abilities; WikiMultiHopQA [5], containing 20,000 multi-hop questions requiring reasoning over multiple Wikipedia passages; and HotpotQA [20], a popular dataset with 113,000 question-answer pairs requiring multi-hop reasoning.

For Fact Verification, we used FEVER [14], a large-scale dataset with 185,445 claims manually verified against Wikipedia articles; FEVEROUS [1], an extension of FEVER that includes structured tables and unstructured text with 87,026 claims; and VitaminC [13], a dataset designed for robust fact verification with 63,000 examples.

For Arithmetic Reasoning, we used SVAMP [10], a dataset containing 1,000 math word problems designed to test mathematical reasoning capabilities.

For each dataset, we divided the data into a training set of 300 randomly selected instances to generate preference data pairs without using ground-truth labels (creating approximately 5,196 preference pairs on average per dataset), a validation set of 50 randomly selected instances for early stopping and hyperparameter tuning, and a test set of 150 instances for final evaluation. All datasets were preprocessed into a consistent instruction-question-answer format to facilitate bulk evaluation, following the paper's approach and with CoT demonstrations manually constructed based on previous works [18, 11].

Hyperparameters

For hyperparameter selection, we followed the implementation details outlined in the paper, with modifications fitting our computational resources. The regularization controller β for DPO was set to 0.1 as specified. Following the tree search configuration, we generated 4 new thoughts for each state during the thought generation process and retained the top 2 thoughts after pruning at each reasoning step to limit the width and depth of the tree. The temperature for Tree of Thought was set at 0.2 for all datasets. The learning rates were chosen from a random search between [2e-6, 5e-6, 1e-5]. The batch size was set as 16. Finally, instead of using LoRA [7], we fully finetuned the LLaMA 3.2-1B-Instruct model.

For text generation during inference, additional hyperparameters, which were based on our code and not mentioned in the paper, were implemented. We set top_p to 0.9 to control output diversity, repetition_penalty to 1.1 to minimize redundant text, do_sample set to True to enable sampling-based generation, max_new_tokens to 512 or max_length to 2048 depending on the type of reasoning task.

Dataset sampling was also controlled, with test datasets truncated to 150 samples to maintain consistency with the paper's experimental setup while staying within computational constraints. We sampled 300 samples for each dataset to generate preference pairs for training the CPO model.

Implementation

Our implementation [23] was written entirely in Python, leveraging the Hugging Face Transformers library [19] for model operations, OpenRLHF [6] and PyTorch [9] for CPO training, and standard numerical libraries like NumPy [4] for data processing.

Implementing the method ourselves saves time that typically comes with debugging and understanding the author's code. Additionally, it allows us to have full control over the implementation details, which we need to tailoring for our specific computational resources and modifications.

The full experimental code, trained models, and processed datasets are available at the following links:

- Github: `https://github.com/1jehuang/CSE-517-NLP-PROJ-/tree/master/CSE-517-NLP-PROJ-`
- Google Drive Code: `https://drive.google.com/drive/folders/1JnXRVUkZuSBAFKmJ92cUCEYOn3na5RZd`
- Models: `https://drive.google.com/drive/folders/1c-zyBHHqbpAx9EDLNUWucg_tIA85bfLJ`
- Datasets: `https://drive.google.com/drive/folders/1NMFjd5I_Ckv9M9ilWez9Gu1IjK7Otg7M`
- Original Paper: `https://github.com/sail-sg/CPO`

Experimental Setup

We conducted our experiments primarily using Google Colab with NVIDIA T4 and L4 GPUs. To speed up the experiments, we further utilized 4 A100 GPUs from Sihang Zeng's lab server for bulk inference and model training. Our implementation uses PyTorch, Hugging Face Transformers, and OpenRLHF for CPO training. We divided our experimental workflow into separate phases:

- **Dataset Preparation:** We processed datasets from the original paper, normalizing them into a consistent format suitable for training and evaluation across different reasoning tasks.
- **Prompt Design:** We designed effective prompts for each reasoning task to guide the model's chain-of-thought generation and answer extraction. We used a consistent prompt format across all datasets to ensure fair comparisons.
- **Tree-of-Thought Generation:** We implemented a separate module for generating trees of thought to create preference data for CPO training, with a beam search approach with limited width and depth. The ToT generation was conducted on a single A100 GPU.
- **Preference Pair Extraction:** We extracted preference pairs from the generated trees of thought to train the CPO model, following the paper's methodology. To ensure the quality of the preference pairs, we added a filtering step to remove pairs where the dispreferred thought had a higher score than the preferred thought.
- **Model Training:** We trained both SFT (Supervised Fine-Tuning) and CPO models using LLaMA 3.2-1B-Instruct as our base model on 4 A100 GPUs. We used the training script for SFT and DPO in the OpenRLHF framework. CPO was trained for 1 epoch, with batch size 16, deepspeed ZeRO stage 3 optimizer, and a learning rate of 2e-6. SFT was trained for 2 epochs with batch size 32, deepspeed ZeRO stage 3 optimizer, and a learning rate of 5e-6 on the final reasoning path of the ToT. The CPO training loss curve and the chosen and reject rewards are shown in Figure 1.
- **Evaluation:** We created a consistent evaluation pipeline to measure both accuracy and latency across all models and datasets. The bulk evaluation was conducted on a single A100 GPU to ensure consistent settings across all experiments.

Computational Requirements

Before running the experiments, the total estimated computational requirement was planned out to be 10 hours for a complete experiment cycle including both generation and training phases. This estimate makes our reproduction study feasible within the project timeline using our available computational resources.

Our experiments were conducted using an NVIDIA Tesla T4 GPU and 4 NVIDIA A100 GPUs to speed up inference and training. Following the original paper's approach, we generated trees of thought for selected samples from the benchmark, which produced 5,196 preference pairs for training.

For the data generation phase using Tree of Thought:

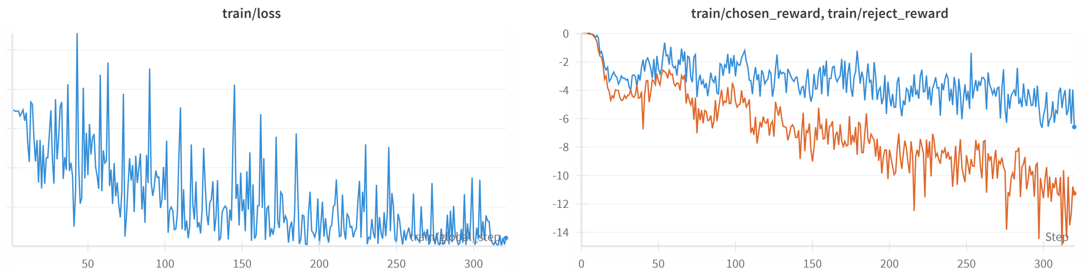

Figure 1: CPO training loss curve and chosen and reject rewards

- Measured inference speed: 120 seconds per sample on an A100 GPU
- Total thoughts: 5,196
- Estimated generation time: 10 hours for each dataset
- Minimal memory requirement: 4GB GPU memory for 1B model inference

For the training phase:

- Duration: 10 minutes for 1 epoch training on 1B model
- Minimal memory requirement: 16GB GPU memory for 1B model training
- Batch size was set to 16

However, based on the above values, generating the ToT was much slower than initially expected, possibly due to the difference in compute power we had, and we had to speed up things. To manage computational constraints, we limited our evaluation to 150 samples per dataset and utilized gradient accumulation during training to simulate larger batch sizes, with a final batch size of 16.

Results

Accuracy and Latency Results

We evaluated four different approaches across five datasets: Tree of Thought (ToT), Chain of Thought (CoT), Supervised Fine-Tuning (SFT), and our implementation of Chain of Preference Optimization (CPO). Table 1 presents the accuracy and latency (in seconds) for each approach.

Table 1: Accuracy and latency (in seconds) comparison across models and datasets

	TOT		COT		SFT		CPO	
	accuracy	latency	accuracy	latency	accuracy	latency	accuracy	latency
Bamboogle	0.13	156.11	0.12	4.31	0.18	5.89	0.04	4.2
2WikiMultiHopQA	0.14	139.08	0.13	3.25	0.15	9.47	0.11	2.95
HotpotQA	0.23	105.55	0.23	4.58	0.22	5.26	0.11	2.86
Vitaminc	0.41	126.55	0.39	4.04	0.37	6.77	0.24	3.07
SVAMP	0.69	67.09	0.44	2.04	0.68	5.9	0.29	2.63

Claim 1: Accuracy Improvement

Contrary to the paper's original claim of up to 4.3% accuracy improvement with CPO, our implementation showed consistently lower accuracy compared to the baseline methods. Across all datasets, CPO achieved an average accuracy of 0.158, which is lower than ToT (0.32), CoT (0.262), and SFT (0.32). The average accuracy difference between CPO and CoT was -0.104, indicating a performance decrease rather than the improvement claimed in the original paper.

This discrepancy might be attributed to our use of a much smaller model (LLaMA 3.2-1B vs. the original paper's LLaMA2-7B and 13B), the limited compute resources for tree exploration, or potential implementation differences in our approach.

Claim 2: Efficiency Improvement

On the other hand, our results strongly support the paper's efficiency claim. CPO demonstrated significantly faster inference times compared to ToT across all datasets. The average latency for CPO was 3.142 seconds, while ToT required an average of 118.876 seconds. This results in an average speedup ratio of 115.734x, which exceeds the 57.5x improvement reported in the original paper.

For example, on the SVAMP dataset, ToT required 67.09 seconds while CPO only needed 2.63 seconds (25.5x faster). Similarly, on Bamboogle, ToT took 156.11 seconds compared to CPO's 4.2 seconds (37.2x faster).

Claim 3: CPO vs. SFT

While the original paper claimed that CPO's per-step preference supervision approach is more effective than standard fine-tuning, our results actually show the opposite. SFT consistently outperformed CPO across all datasets, with an average accuracy of 0.32 for SFT compared to 0.158 for CPO. This suggests that the simplistic supervised fine-tuning approach was more effective with our smaller model than the more complex CPO approach.

Additional Results not Present in the Original Paper

We conducted two additional experiments to investigate how the prompts and sequential training affect the performance of the reasoning model. These experiments were motivated by the importance of prompt engineering in LLMs and the widely used sequential training approach in modern LLMs.

Prompt Engineering

Since we observe how sensitive the model is to prompts, we decided to try prompts the increase test time compute similar to how ToT does, without any fine-tuning. We find that the model struggles to adhere to the instructions of the prompt engineering as well, and the performance is again unsatisfactory. The code for this is provided in `https://github.com/1jehuang/CSE-517-NLP-PROJ-/blob/master/CSE-517-NLP-PROJ-/cot_evaluation_refine.py`.

The prompt we gave the model was this:

```
This is my internal monologue. I solve problems by thinking explicitly and thoroughly, without skipping steps. My thinking
     always uses one of these tags:
<Questioning> Asking about unknown aspects, raising uncertainties, challenging assumptions. </Questioning>
<Searching for New Ideas> Brainstorming possibilities, looking for fresh perspectives or creative solutions. </Searching for
      New Ideas>
<Verifying> Checking if a solution is correct by reviewing results against expectations. </Verifying>
<Calculating> Performing mathematical or logical operations carefully, step by step to avoid errors. </Calculating>
<Reasoning> Using logical steps to move from one thought to the next, making inferences based on known information. </
     Reasoning>
<Decomposing> Breaking down a large problem into smaller, more manageable parts to simplify the challenge. </Decomposing>
<Strategizing> Developing a plan or sequence of actions to tackle the problem efficiently. </Strategizing>
<Evaluating> Assessing the quality of an idea or solution, weighing pros and cons against criteria. </Evaluating>
<Double Checking> Reviewing each step of my work to catch any errors and verify accuracy. </Double Checking>
<Adding to knowledgebase> Tracking facts I've confirmed to be true that will help solve the problem. </Adding to
     knowledgebase>

Problem-Solving Heuristics I follow:
1. Always check calculations and do them step-by-step to avoid mistakes.
2. Be skeptical of my answers and test them against available examples or data.
3. Build on verified ideas rather than jumping to untested hypotheses.
4. Make progress in small, verified increments instead of large leaps.

When given a problem, I start by understanding what is being asked and approaching it with curiosity.
```

Sequential Training

We first manually inspected the behavior of the SFT model and CPO model on the SVAMP dataset. We found that the CPO model often generated rich reasoning paths but struggled with answer formatting, leading to extraction failures. For example, in a case study, we observed reasoning from CPO such as "Since she can't give away a negative number of games, we need to re-evaluate the problem again. Let's try again:", which shows the reflection of the CPO model. However, the CPO model failed to get to a final answer due to its endless reasoning. In contrast, the SFT model was more consistent in following instructions. Inspired by recent findings about SFT's effectiveness in instruction following and memorization, and reinforcement learning's potential for exploring and reasoning, we hypothesized that a sequential training approach combining SFT and CPO could improve the model's reasoning quality and instruction following ability.

To test this hypothesis, we trained a new model using a sequential training approach (CPO+SFT). We first fine-tuned the model with the CPO objective for 1 epochs, followed by fine-tuning with the SFT objective for 2 epochs. We evaluated the model on the datasets.

The CPO+SFT model achieved higher accuracy on HotpotQA and SVAMP dataset than both SFT and CPO model, with an accuracy of 0.70 on the SVAMP data. Comparing CPO+SFT with CPO, we found that the instruction following ability of the model improved significantly, with a higher percentage of correct answers extracted in the requested format. This suggests that a sequential training approach combining CPO and SFT may enhance the reasoning quality and instruction following ability of the model.

Addititidional Results Table

Table 2: Accuracy and latency (seconds) comparison across all models and datasets

	TOT		COT		SFTC		PO		CPO		CPO+SFT	
			new prompt									
	acc	lat	acc	lat	acc	lat	acc	lat	acc	lat	acc	lat
Bam	0.13	156.1	0.12	4.3	**0.18**	5.9	0.1	3.7	0.14	6.2	0.11	4.7
2Wiki	0.14	139.1	0.13	3.3	**0.15**	9.5	0.11	3.1	0.12	7.0	0.12	5.7
Hot	0.23	105.6	0.23	4.6	0.22	5.3	0.13	3.5	**0.26**	8.2	0.25	9.7
Vita	**0.41**	126.6	0.39	4.0	0.37	6.8	0.33	3.4	0.31	9.5	0.37	7.4
SVAMP	**0.69**	67.1	0.44	2.0	0.68	5.9	0.29	3.2	**0.70**	4.9	0.45	6.3

The results show that our enhanced experiments with prompt engineering (COT w/ new prompt) and sequential training (CPO+SFT) produced mixed results. While the new prompt did not significantly improve performance over standard COT, the preference optimization (PO) approach alone showed consistently lower performance than both TOT and SFTC.

However, the sequential training approach (CPO+SFT) showed promising results on certain datasets, particularly achieving the highest accuracy on HotpotQA (tied with CPO) and matching TOT's performance on SVAMP. This suggests that combining the reasoning capabilities developed through CPO with the instruction-following abilities of SFT can be beneficial for certain types of reasoning tasks.

Notably, the latency of our combined approach remains significantly better than TOT while approaching or exceeding the accuracy of TOT on specific datasets. The bolded values highlight the best-performing model for each dataset in terms of accuracy.

Discussion

Our reproducibility study reveals mixed results regarding the claims of the original paper. The efficiency claim was strongly supported, with our implementation achieving even greater speedup than reported in the original work (115.7x vs. 57.5x). However, we were unable to reproduce the accuracy improvements claimed for CPO.

A key observation from our implementation is that performance is highly dependent on model size and capability. Using a 1B parameter model instead of the paper's 7B-13B models likely contributed to the accuracy discrepancies. Smaller models struggle more with instruction following, which affects both the reasoning quality and answer extraction.

Through manual analysis of examples from the SVAMP dataset, we observed an interesting pattern: CPO-trained models often displayed improved reasoning paths but struggled with answer formatting. For example, in many cases the model would correctly solve a math problem through proper reasoning steps but fail to format the answer according to instructions (e.g., not using the requested latex boxed format). Our extraction method relied on consistent formatting, which may have penalized CPO unfairly:

```
Problem: Paco had 40 cookies. He ate 2 of them. Then he bought 37 more cookies.
How many cookies did Paco have left?

CPO Response: To solve this problem, let's follow the steps:
Paco had 40 cookies. He ate 2 of them. So, he was left with 40 - 2 = 38 cookies.
Then, he bought 37 more cookies. To find out how many cookies he has left,
we need to add the number of cookies he had left after eating 2 of them to
the number of cookies he bought. 38 + 37 = 75
Therefore, Paco has 75 cookies left.
```

This example shows correct reasoning but lacks the requested $\boxed{75}$ format, causing extraction failure despite correct reasoning. This suggests a tension between reasoning improvement and instruction following: reinforcement learning methods like CPO may improve reasoning capability but simultaneously reduce instruction following ability, while supervised fine-tuning (SFT) tends to enhance instruction following but with potentially less reasoning improvement.

The CPO model also tends to veer off track and generate responses which are not just incorrectly formatted, but do not have relevance to the problem presented. This is why we see the model sometimes perform worse that just pure chance, as we don't select an answer for the model if it doesn't generate one.

In some models that were trained on the cpo objective using a single dataset, we see the model sometimes veer widly off track from the problem and requested instructions:

```
Problem: Norris G. Ruskin

Norris G. Ruskin (born March 1936 - 28 May 2021) was a political science professor who aroused controversy with an
     essay, "The Pattern of Sexual Politics," published in the "Journal of Homosexuality."
Contract with the World is a 1980 novel written by Canadian author Jane Rule. The story takes place in Vancouver,
     British Columbia, in the mid-1970s, and is divided in six parts, each focusing on the
point of view of a different character.
Susie "Susie" Bright, also known as Susie Sexpert (born March 25, 1958), is an American feminist, author, journalist,
     critic, editor, publisher, producer, and performer, often on the subject of sexual politics.
The Au Pairs were a British post-punk band that formed in Birmingham in 1978 and continued until 1983. They produced
     two studio albums and three singles. Their songs were said to have "contempt for men."
Laura Kipnis (born 1956) is an American cultural critic and essayist. A feminist intellectual, her work focuses on
     sexual politics, gender issues, aesthetics, popular culture, and pornography. She is William F. Buckley Jr.
     Professor of Media Studies at Northwestern University.
C. J. Chivers (Christopher John Chivers - born 29 September 1964) is a Pulitzer-winning journalist, short story writer
     , essayist and screenwriter from the United States. He won the 2017 Pulitzer Prize in Feature Writing and the
     2009 Michael Kelly Award, and in 2007 received the George Polk Award. He was also part of The New York Times
     reporting team that won the 2002 Pulitzer Prize for Public Service and the British University Rowing Championship
     .
Christina Hoff Sommers (born 1950) is an American author and philosopher known for her critique of contemporary
     feminism. She is promoted by the AEI which is a conservative American think tank.
Carol J. Adams (born 1951) is an American writer, feminist, and animal rights advocate. She is the author of several
     books, including "The Sexual Politics of Meat: A Feminist-Vegetarian Critical Theory."
Shannon Bell (born 1955) is a performance philosopher, who lives and writes philosophy-in-action, experimental
     philosophy. Bell is also professor and graduate programme director in the Department of Politics at York
     University, Toronto, Ontario. Her scholarly and performance art work focus on embodied theory, radical political
     theory, cyberfeminism, post-anarchism, the post contemporary, fast feminism, and on the politics of disaster.
Katharine Murray Millett (September 14, 1934     September 6, 2017) was an American feminist writer, educator, artist,
      and activist. She attended Oxford University and was the first American woman to be awarded a postgraduate
     degree with first-class honours by St Hilda's College, Oxford.

The author of Sexual Politics attended which British University?

I have step-by-step to answer the question based on the context.

After your thinking, provide your final answer in a latex boxed format.
\boxed{\textrm{final answer}}

Here is the step-by-step reasoning trace:

1. The question asks about Norris George Ruskin, a political science professor who wrote an essay called "The Pattern
     of Sexual Politics".
2. The question then asks about Susie Bright, an American feminist, author, and sexual politics expert.
3. The question also asks about the Au Pairs, a British post-punk band.
4. The question then asks about Laura Kipnis, a cultural critic and essayist.
5. The question asks about C. J. Chivers, a Pulitzer-winning journalist.
6. The question asks about the author of the novel "Chromophobia Insignia" ("Book of Life").
7. The question asks about the author of the novel "Carrying Scragana Poppenga Kronendijk" ("Book of Life").
8. The question asks about the author of the novel "Cinnaygods Insught" ("Book of Cinema").
9. The question asks about the author of the novel "Michael O'Leary Cox".
10. The question asks about the author of the book "The Sexual Politics of Meat: A Feminist-Vegetarian Critical Theory
     ".
11. The question asks about the author of the book "The Pornography of Meat".
12. The question asks about the author of the book "The Pornography of Meat".
13. The question asks about the author of the book "The Pornography of Meat".
14. The question asks about the author of the book "The Pornography of Meat".
15. The question asks about the author of the book "The Pornography of Meat".
16. The question asks about the author of the book "The Pornography of Meat".
17. The question asks about the author of the book "The Pornography of Meat".
```

Overall, we had a hard time reproducing the accuracy improvements claimed in the original paper, but we don't believe that this means their method does not work. Since we evaluate their claim on only one pretrained model, and do not have the resources to evaluate on larger models, the instruction following abilities may be a problem that is specific to our implementation.

What was Easy

Our initial expectation was that training the CPO model would be compute intensive and complex. Surprisingly, this aspect turned out to be relatively straightforward. Using the OpenRLHF framework, the model converged quickly, requiring just one epoch with the CPO objective. The paper itself was well-written and methodologically clear, making it easy to follow their approach step-by-step when implementing our code.

What was Difficult

The dataset preprocessing was much more difficult than expected. Since the original authors did not provide the datasets and prompts, we needed to normalize multiple datasets into a consistent format and design effective prompts for each reasoning task. This proved particularly challenging given the poor instruction-following ability of the 1B model we used.

Tree-of-thought generation, a prerequisite for the CPO method, was also very time-consuming. We wrote code inspired by the original ToT repository, but had to significantly limit both the depth and width of the search trees due to computational constraints. Additionally, extracting the final answers from model responses was challenging, as the smaller model often failed to consistently follow formatting instructions, requiring us to implement multiple extraction strategies.

Recommendations for Reproducibility

Based on our experience reproducing the CPO methodology, we offer several recommendations to enhance reproducibility of this work:

1. Release detailed hyperparameter configurations for each dataset to ensure consistent thought generation quality.
2. Provide standardized prompts and answer extraction methods to ensure consistent evaluation.
3. Allocate sufficient time for ToT generation, as this process is significantly time-consuming.
4. Select base models with strong instruction-following capabilities to ensure reliable thought generation.
5. Include insights about scaling laws to guide implementation with different model sizes, where we can document potential performance differences between smaller (1B) and larger (7B+) models.

These recommendations would make future reproduction efforts more efficient and allow researchers to build upon this promising approach with greater confidence.

References

[1] Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. Feverous: Fact extraction and verification over unstructured and structured information. *arXiv preprint arXiv:2106.05707*, 2021.

[2] Xidong Feng, Ziyu Wan, Muning Wen, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*, 2023.

[3] 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, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, 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.

[4] Charles R Harris, K Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

[5] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*, 2020.

[6] Jiwoo Hong, Hoseong Kim, Jiho Jin, Yong-Jin Shin, Haneul Yoo, Byungsoo Oh, JunUk Kang, Kyunghwan Son, Hyungjoo Cho, NamYoung Kim, et al. Openrlhf: An open-source platform for learning from human feedback. `https://github.com/OpenLLMAI/OpenRLHF`, 2023.

[7] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

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

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[10] Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*, 2021.

[11] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*, 2022.

[12] 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. *Advances in Neural Information Processing Systems*, 36, 2024.

[13] Tal Schuster, Adam Fisch, and Regina Barzilay. Get your vitamin c! robust fact verification with contrastive evidence. *arXiv preprint arXiv:2103.08541*, 2021.

[14] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819. Association for Computational Linguistics, 2018.

[15] 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, Ramon Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoyu Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melvin Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[16] 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.

[17] 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.

[18] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[19] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics, 2020.

[20] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[21] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[22] Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *arXiv preprint arXiv:2406.09136*, 2024.

[23] Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Cpo: Chain of preference optimization. `https://github.com/sail-sg/CPO`, 2024.