Step-Controlled DPO: Leveraging Stepwise Error for Enhanced Mathematical Reasoning

Zimu Lu*, **Aojun Zhou***, **Ke Wang**, **Houxing Ren**, **Weikang Shi**
Junting Pan, **Mingjie Zhan**†, **Hongsheng Li**†
Multimedia Laboratory (MMLab), The Chinese University of Hong Kong
luzimu@mail.ustc.edu.cn {aojunzhou, zmjdll}@gmail.com
hsli@ee.cuhk.edu.hk

Abstract

Direct Preference Optimization (DPO) has proven effective at improving the performance of large language models (LLMs) on downstream tasks such as reasoning and alignment. In this work, we propose Step-Controlled DPO (SCDPO), a method for automatically providing stepwise error supervision by creating negative samples of mathematical reasoning rationales that start making errors at a specified step. By applying these samples in DPO training, SCDPO can better align the model to understand reasoning errors and output accurate reasoning steps. We apply SCDPO to both code-integrated and chain-of-thought solutions, empirically showing that it consistently improves the performance compared to naive DPO on three different SFT models, including one existing SFT model and two models we finetuned. Qualitative analysis of the credit assignment of SCDPO and DPO demonstrates the effectiveness of SCDPO at identifying errors in mathematical solutions. We then apply SCDPO to an InternLM2-20B model, resulting in a 20B model that achieves high scores of 88.5% on GSM8K and 58.1% on MATH, rivaling all other open-source LLMs, showing the great potential of our method.

1 Introduction

Large language models (LLMs) have shown great potential in mathematical problem-solving. Recently, Direct Preference Optimization (DPO; [26]) has emerged as a popular choice for aligning LLMs with relative feedback to improve the quality of generated text. Prior works [8, 24, 41] have demonstrated that reinforcement learning algorithms and DPO can improve the mathematical reasoning abilities of LLMs, making the generated reasoning process more controllable. Different from other tasks that need human or AI feedback, the final answer to a mathematical problem serves as a reliable way to judge the quality of the model's response, since a mathematical problem typically has a single correct answer. As a result, the responses producing the correct final answers are desirable and can serve as the preferred samples, while the ones reaching incorrect final answers are undesirable and can serve as the dispreferred samples.

However, solutions to a mathematical problem can be diverse, with many different reasoning paths arriving at the correct final answer and many subtle ways to make mistakes. Determining the preferred and dispreferred responses based on the final answer is coarse and may be inadequate for capturing *the intricacies of the multi-step mathematical reasoning process.* Previous studies introduce process supervision [16], but it requires large amounts of meticulous and expensive human annotation and only applies to traditional RL algorithms.

*Equal contribution †Corresponding author

Preprint. Under review.

In this paper, we show how to automatically provide explicit stepwise preference supervision by generating dispreferred solutions that start making errors at a specific step. We propose *Step-Controlled DPO (SCDPO)*, an algorithm that introduces stepwise supervision without necessitating extra human annotation. This approach starts with a model finetuned with question-solution pairs and possessing initial math-solving capabilities, which is used to generate solutions to a set of math problems. We choose the solutions whose final answers match those of the ground truth. The reasoning steps in these solutions can be seen as correct, since the cases of a wrong solution reaching the right answer are rare. We take each of these correct solutions and start generating with the model via modulating the hyperparameter of the model, i.e., increasing the temperature of the final softmax function, from various intermediate steps of that solution, and retain the samples where the final answer is incorrect. In this way, the steps before the intermediate step are the same as the original correct solution, while the steps after are the ones with possible errors. During DPO training, the correct solutions are the preferred samples, and they are paired with the wrong solutions generated in this way, with the question and the steps before the intermediate step as the prompts. These step-controlled training samples help models learn detailed reasoning abilities and are mixed with naive DPO training data produced by only checking the final answer, which optimizes the general form of the solution.

Our contributions are as follows:

- We introduce Step-Controlled DPO (SCDPO), which we empirically show improves the performance of DPO in enhancing LLMs' mathematical reasoning abilities. We also conduct qualitative analysis of credit assignment of SCDPO.
- We conduct experiments on chain-of-thought and code-integrated solutions, showing that SCDPO can effectively improve mathematical problem-solving performance of three different SFT models.
- Using SCDPO, we finetune an InternLM2-20B model, which reaches 88.5% on GSM8K [9] and 58.1% on MATH [13], rivaling all other open-source models, demonstrating the great potential of our method.

2 Related Work

LLM for Mathematical Reasoning. Prior works have explored various methods to enhance mathematical reasoning abilities of LLMs. Prompting methods, such as Chain-of-Thought [38], Tree-of-Thought [44], PAL [11], Program-of-Thought [7], and CSV [53], use carefully engineered prompts to bring out LLMs' mathematical skills without changing their parameters. Other works optimize parameters of LLMs for enhanced mathematical reasoning through either pretraining or finetuning. Llemma [2], and MathPile [37] continue pretraining LLMs on large amounts of math-related data, while RFT [49], Mammoth [50], MathCoder [35], WizardMath [20], ToRA [12], MetaMath [47], MathGLM [43], and MathGenie [19] finetune pretrained models on question-solution pairs. These methods effectively improves LLMs' ability to solve challenging mathematical problems, demonstrating impressive performance on mathematical benchmarks such as GSM8K [9], MATH [13], etc. Our work builds upon models that have undergone pretraining and finetuning, using DPO to further enhance their mathematical abilities.

Aligning LLMs Using Relative Feedback. Methods that align LLMs with human or AI annotated preference data have been used to improve performance on a variety of downstream tasks such as translation [15], summarization [31, 54], and instruction-following [23, 27]. Reinforcement learning from human (or AI) feedback [8, 4] first trains a reward model, then uses reinforcement learning algorithms such as REINFORCE [40], PPO [28], or variants [27] to maximize the reward. To simplify the pipeline, several direct alignment methods [26, 1, 52] have been proposed. Among them, DPO [26] and several of its variants [24, 10, 17, 1] offer a way to optimize the reward function without having to train an extra reward model, proving highly effective on various tasks [33, 48].

Recently, these preference alignment methods have also been applied to mathematical problem-solving tasks. DeepSeekMath [29] uses RL to improve mathematical accuracy, while ChatGLM-Math [41] uses DPO to improve model's mathematical generation quality. Process supervision [16] uses stepwise preference of mathematical solution in its RL finetuning, which is highly effective but needs costly fine-grained human annotation. Our work offers a way to create stepwise error annotations of preferred and dispreferred solution pairs, and uses the data to improve DPO's performance on mathematical problem-solving tasks.

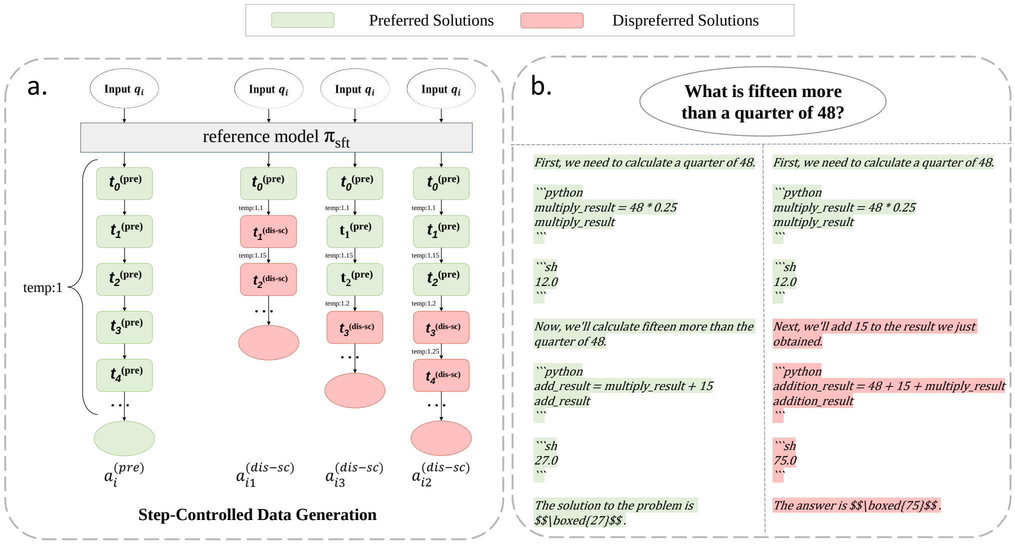

Figure 1: Demonstration and example of the step-controlled data generation process. **a.** Step-controlled data generation. First, a solution reaching the correct final answers is collected, which we denote as $a_i^{(\text{pre})}$. Then, erroneous solutions that reach incorrect final answers are generated, starting from intermediate steps of $a_i^{(\text{pre})}$, creating dispreferred solutions $a_{i1}^{(\text{dis-sc})}$, $a_{i2}^{(\text{dis-sc})}$, and $a_{i3}^{(\text{dis-sc})}$. These dispreferred solutions share the steps before the intermediate steps with $a_i^{(\text{pre})}$. The temperature of the newly generated steps gradually increases with each step to make the generation more erroneous. **b.** An example of a pair of preferred and dispreferred solutions. The dispreferred solution starts making errors after a particular intermediate step.

3 Step-Controlled DPO Pipeline

In this section, we introduce Step-Controlled DPO (SCDPO), a pipeline for automatically generating preferred and dispreferred responses to math problems, with annotations of erroneous solving steps, and using these responses in DPO training to enhance the mathematical reasoning abilities of LLMs. Our method consists of two stages: step-controlled data generation, and step-aware DPO training. The two stages construct a feedback-alignment framework that is both effective and cost-efficient.

Initial Model. Our method starts with an initial model, denoted as π_{SFT}, which has been finetuned with question-solution pairs from the training sets of GSM8K and MATH, two high-quality mathematical datasets that contain grad-school math word problems and competition-level math problems, respectively. When prompted with a math problem q, π_{SFT} is able to generate a step-by-step solution, denoted as a. a can be broken down into a sequence of reasoning steps, for example, $a = (t_0, \ldots, t_m)$. Here, t_i $(i = 0, \ldots, m)$ represents either a code reasoning step or a natural language reasoning step within a.

We experiment with two solution formats: code-integrated solution format [53], and chain-of-thought solution format [38]. For the code-integrated solution format, t_i in $(t_0, \ldots, t_m)$ alternates between a code reasoning step and a natural language reasoning step, while for the chain-of-thought solution format, all the steps are in natural language. We primarily use the code-integrated solution format, as previous works [35, 19, 12] show that it results in higher accuracy than chain-of-thought. We finetune a Mistral-7B model with 34K samples of code-integrated solutions from GSM8K and 47K from MATH to create the initial model. We also validate our method on the chain-of-thought format using the off-the-shell MetaMath-Mistral-7B model, as well as MathCoder-Mistral-7B, which we trained using the MathCodeInstruct dataset [35].

3.1 Step-Controlled Data Generation

The data we collect is in two parts: naive DPO data D_{naive} and Step-Controlled DPO data D_{SC}.

Generation of D_{naive}. D_{naive} contains pairs of preferred-dispreferred samples, used to optimize the general form of the solution. To create D_{naive}, we prompt π_{SFT} with math questions in the training sets of GSM8K and MATH. The training set of GSM8K and MATH each contains 7.5K questions. Each question is presented to π_{SFT} multiple times and various solutions are generated, with a temperature of 1. The quality of the generated solutions is judged by the final answers. If a solution reaches the same final answer as the ground truth, and no errors or adjustments occur at any of the reasoning steps (we detect these by looking for strings like "error" or "apologies"), the solution is seen as preferred, while the solutions that reach answers different from the ground truth are considered dispreferred. The questions in GSM8K and MATH are all open-ended, and it is unlikely for incorrect reasoning steps without errors and adjustments to lead to the correct answer, so the reasoning steps in the preferred solutions can reliably be seen as correct. We randomly sampled 87 solutions that reach correct final answers, and found that of the 369 reasoning steps in these solutions, only 2 contain errors, which is a very small percentage (about 0.5%). The solution generation of each question stops when at least one preferred solution and one dispreferred solution are generated, or the number of solutions generated reaches an upper limit of K. For the code-integrated solution format, K is set as 100. This results in around 6.5K preferred-dispreferred solution pairs for GSM8K and MATH respectively, combining into a total of approximately 13K DPO training pairs. The resulting data can be expressed as:

$$D_{\text{naive}} = \{(q_i, a_i^{(\text{pre})}, a_i^{(\text{dis})}) : i = 1, \ldots, N_{\text{naive}}\} \tag{1}$$

Here, q_i denotes the ith question, while $a_i^{(\text{pre})}$ and $a_i^{(\text{dis})}$ represent the preferred and dispreferred solution to the ith question.

Generation of D_{SC}. In order to generate solutions with stepwise error annotations for DPO training, we propose a method to automatically generate training data with errors starting to occur at a controlled step. The process is demonstrated in Fig. 1. To do this, we first take a preferred solution from D_{naive}, denoted as $a_i^{(\text{pre})} = (t_0^{(\text{pre})}, \ldots, t_k^{(\text{pre})}, t_{k+1}^{(\text{pre})}, \ldots, t_{m_i}^{(\text{pre})})$. Here, $t_k^{(\text{pre})}$ is a random intermediate step within $a_i^{(\text{pre})}$. As $a_i^{(\text{pre})}$ is a correct solution, $t_0^{(\text{pre})}, \ldots, t_{m_i}^{(\text{pre})}$ are all correct steps, as we have taken care to retain only those solutions with no execution errors, apologies or rectifications. As shown in Fig. 1 a, to create a solution with errors occurring after step k, we present π_{SFT} with sequence $(q_i, t_0^{(\text{pre})}, \ldots, t_k^{(\text{pre})})$, and raise the temperature of the final softmax function to affect the generation quality, increasing the occurrence of errors in the following steps. Raising the temperature causes the model performance to become unstable and erroneous. The effect of raised temperature on accuracy is demonstrated in Fig. 5 of Appendix. A. We observe that when the temperature is instantly raised and remains at a high value, the model can generate garbled strings as errors accumulate, which does not represent any reasoning mistakes and contains no valuable information. To avoid this, we adopt a gradually increasing temperature, which initially starts at 1.1, and increases by 0.05 with each generated step, until the generation ends or the temperature reaches 1.4. This setting empirically reduces the frequency of the occurrence of garbled text, while increasing the error rate. We generate the steps following $(q_i, t_0^{(\text{pre})}, \ldots, t_k^{(\text{pre})})$ multiple times, until one reaching an incorrect answer is generated. Appending the generated steps to $(t_0^{(\text{pre})}, \ldots, t_k^{(\text{pre})})$, we get a dispreferred solution with step-controlled error, denoted as $a_{ik}^{(\text{dis-sc})} = (t_0^{(\text{pre})}, \ldots, t_k^{(\text{pre})}, t_{k+1}^{(\text{dis-sc})}, \ldots, t_{n_i}^{(\text{dis-sc})})$, where the sequence $(t_{k+1}^{(\text{dis-sc})}, \ldots, t_{n_i}^{(\text{dis-sc})})$ is erroneous. An example is presented in Fig. 1 b. The resulting data can be expressed as:

$$D_{\text{SC}} = \{(q_i, a_i^{(\text{pre})}, a_{ik}^{(\text{dis-sc})}) : i = 1, \ldots, N_{\text{SC}}; k \in [0, m_i - 1]\} \tag{2}$$

Here, q_i denotes the ith question, while $a_i^{(\text{pre})}$ is the preferred solution, and $a_{ik}^{(\text{dis-sc})}$ is the dispreferred solution with step-controlled error that occurs after $t_k^{(\text{pre})}$. N_{SC} is the number of questions in D_{SC}, while m_i is the index of the last step of $a_i^{(\text{pre})}$.

3.2 Step-Controlled DPO Training

Having collected D_{naive} and D_{SC}, we apply them to DPO training. D_{naive} serves to regulate the general form of solutions, while D_{SC} supervises the model's reasoning on a step level. During DPO training, samples in D_{naive} and D_{SC} are mixed together randomly, and the DPO loss is applied to each sample. For samples from D_{naive}, the loss is applied to all steps in the preferred and dispreferred solutions, which can be written as:

$$\begin{aligned}&\mathcal{L}_{\text{naive}}(\pi_\theta;\pi_{\text{SFT}})\\&= -\mathbb{E}_{(q_i,a_i^{(\text{pre})},a_i^{(\text{dis})})\sim\mathcal{D}_{\text{naive}}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(a_i^{(\text{pre})}|q_i)}{\pi_{\text{SFT}}(a_i^{(\text{pre})}|q_i)}-\beta\log\frac{\pi_\theta(a_i^{(\text{dis})}|q_i)}{\pi_{\text{SFT}}(a_i^{(\text{dis})}|q_i)}\right)\right]\end{aligned}\tag{3}$$

For a pair of preferred and dispreferred solutions in D_{SC}, $a_i^{(\text{pre})}=(t_0^{(\text{pre})},\dots,t_k^{(\text{pre})},t_{k+1}^{(\text{pre})},\dots,t_{m_i}^{(\text{pre})})$ and $a_{ik}^{(\text{dis-sc})}=(t_0^{(\text{pre})},\dots,t_k^{(\text{pre})},t_{k+1}^{(\text{dis-sc})},\dots,t_{n_i}^{(\text{dis-sc})})$, the loss is only applied to the steps after $t_k^{(\text{pre})}$, i.e. $(t_{k+1}^{(\text{pre})},\dots,t_{m_i}^{(\text{pre})})$ and $(t_{k+1}^{(\text{dis-sc})},\dots,t_{n_i}^{(\text{dis-sc})})$. We denote $(t_0^{(\text{pre})},\dots,t_k^{(\text{pre})})$ as $a_{ik-\text{front}}^{(\text{pre})}$, $(t_{k+1}^{(\text{pre})},\dots,t_{m_i}^{(\text{pre})})$ as $a_{ik-\text{end}}^{(\text{pre})}$, and $(t_{k+1}^{(\text{dis-sc})},\dots,t_{n_i}^{(\text{dis-sc})})$ as $a_{ik-\text{end}}^{(\text{dis-sc})}$, so $a_i^{(\text{pre})}=(a_{ik-\text{front}}^{(\text{pre})},a_{ik-\text{end}}^{(\text{pre})})$, and $a_{ik}^{(\text{dis-sc})}=(a_{ik-\text{front}}^{(\text{pre})},a_{ik-\text{end}}^{(\text{dis-sc})})$. The loss function can be written as:

$$\begin{aligned}&\mathcal{L}_{\text{SC}}(\pi_\theta;\pi_{\text{SFT}})=\\&-\mathbb{E}_{(q_i,a_i^{(\text{pre})},a_{ik}^{(\text{dis-sc})})\sim\mathcal{D}_{\text{SC}}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(a_{ik-\text{end}}^{(\text{pre})}|q_i,a_{ik-\text{front}}^{(\text{pre})})}{\pi_{\text{SFT}}(a_{ik-\text{end}}^{(\text{pre})}|q_i,a_{ik-\text{front}}^{(\text{pre})})}-\beta\log\frac{\pi_\theta(a_{ik-\text{end}}^{(\text{dis-sc})}|q_i,a_{ik-\text{front}}^{(\text{pre})})}{\pi_{\text{SFT}}(a_{ik-\text{end}}^{(\text{dis-sc})}|q_i,a_{ik-\text{front}}^{(\text{pre})})}\right)\right]\end{aligned}\tag{4}$$

Combining $\mathcal{L}_{\text{naive}}$ and $\mathcal{L}_{\text{SC}}$, the final loss function of Step-Controlled DPO is as follows:

$$\mathcal{L}_{\text{SCDPO}}=\mathcal{L}_{\text{naive}}+\mathcal{L}_{\text{SC}}\tag{5}$$

In this way, $\mathcal{L}_{\text{naive}}$ optimizes the general form of the solution, while $\mathcal{L}_{\text{SC}}$ focuses on detailed reasoning steps, thus improving the model's accuracy in solving mathematical problems.

4 Theoretical Explanation of Step-Controlled DPO

Theoretical Insight. In this section, we provide some theoretical insights into why SCDPO can effectively enhance the reasoning ability of LLMs. As explained in [25], the DPO loss can be cast into token-level MDP. Similarly, we can also derive a step-level MDP for $\mathcal{L}_{\text{SC}}$ as follows:

$$\begin{aligned}&\mathcal{L}_{\text{SC}}(\pi_\theta;\pi_{\text{SFT}})=\\&-\mathbb{E}_{(q_i,a_i^{(\text{pre})},a_{ik}^{(\text{dis-sc})})\sim\mathcal{D}_{\text{SC}}}\left[\log\sigma\left(\left(\sum_{j=k+1}^{m_i}\beta\log\frac{\pi_\theta(t_j^{(\text{pre})}|q_i,t_{<j})}{\pi_{\text{SFT}}(t_j^{(\text{pre})}|q_i,t_{<j})}\right)-\left(\sum_{j=k+1}^{n_i}\beta\log\frac{\pi_\theta(t_j^{(\text{dis-sc})}|q_i,t_{<j})}{\pi_{\text{SFT}}(t_j^{(\text{dis-sc})}|q_i,t_{<j})}\right)\right)\right]\end{aligned}\tag{6}$$

Here, $\beta\log\frac{\pi_\theta(t_j^{(\text{pre})}|q_i,t_{<j})}{\pi_{\text{SFT}}(t_j^{(\text{pre})}|q_i,t_{<j})}$ and $\beta\log\frac{\pi_\theta(t_j^{(\text{dis-sc})}|q_i,t_{<j})}{\pi_{\text{SFT}}(t_j^{(\text{dis-sc})}|q_i,t_{<j})}$ represent the reward of a single preferred or dispreferred step. For naive DPO, all steps in the preferred and dispreferred solutions have their rewards affecting the loss. However, many steps in the dispreferred solution are actually correct, as the error often occurs in a later step. Step-Controlled DPO reduces the range of steps, starting from the $(k+1)$th step, from which the dispreferred steps are more likely to be erroneous due to the raised sampling temperature. The focus of the optimization is thus cast on the errored steps rather than the whole solution, letting the model learn more detailed reasoning abilities.

Qualitative Evaluation of Credit Assignment of SCDPO. We perform qualitative evaluation of credit assignment on two models trained with SCDPO and DPO respectively. For a sequence of tokens $\mathbf{x}=(x_0,\dots,x_m)$, where x_i is the ith token in the sequence, we denote all the tokens before x_i as $\mathbf{s}_i$, written as $\mathbf{s}_i=(x_0,\dots,x_{i-1})$. As introduced in recent research [47], the DPO implicit reward can be expressed as follows:

$$r(\mathbf{s}_i,x_i)=\beta\log\pi(x_i|\mathbf{s}_i)-\beta\log\pi_{\text{SFT}}(x_i|\mathbf{s}_i)\tag{7}$$

```
To find out how many fireflies remained, we need to follow these
steps:
1. Start with the initial count of fireflies: 3.
2. Add 4 less than a dozen more fireflies: \( \text{dozen} = 12 \),
so \( 4 \times \text{(less than a dozen)} = 4 \times (12 - 4) \).
3. Find the total number of fireflies before two flew away.
4. Subtract 2 because two of the fireflies flew away.
```

```
To find out how many fireflies remained, we need to follow these
steps:
1. Start with the initial count of fireflies: 3.
2. Add 4 less than a dozen more fireflies: \( \text{dozen} = 12 \),
so \( 4 \times \text{(less than a dozen)} = 4 \times (12 - 4) \).
3. Find the total number of fireflies before two flew away.
4. Subtract 2 because two of the fireflies flew away.
```

Figure 2: Credit assignment of part of a solution for a GSM8K problem. Each token is colored corresponding to the DPO implicit reward as expressed in Eq. 7 (darker is higher). The left is the credit assignment of SCDPO, which correctly highlights the error – 4 less than a dozen is not 4 times (12 - 4), while the credit assignment of DPO on the right fails to highlight it.

```
When adding or subtracting numbers, the remainders behave
accordingly. In our case, we're adding the terms, so the overall
remainder will be:

\[
\text{Remainder of sum} = (8 + 1 + 1) \mod 9
\]
```

```
When adding or subtracting numbers, the remainders behave
accordingly. In our case, we're adding the terms, so the overall
remainder will be:

\[
\text{Remainder of sum} = (8 + 1 + 1) \mod 9
\]
```

Figure 3: Credit assignment of part of a solution for a MATH problem. Each token is colored corresponding to the DPO implicit reward as expressed in Eq. 7 (darker is higher). The left is the credit assignment of SCDPO, which correctly highlights the error – as the original question was "Find the remainder when $8 \cdot 10^{18} + 1^{18}$ is divided by 9", the remainders of the terms 8, 10^{18}, and 1^{18} should not be summed, while the credit assignment of DPO on the right fails to highlight the error.

Here $r(\mathbf{s}_i, x_i)$ denotes the DPO implicit reward of token x_i, which is the value we visualize as the background color of the token. A darker color represents a higher reward value. As demonstrated in Fig. 2 and Fig. 3, when presented with an incorrect reasoning step, SCDPO more accurately identifies the incorrect tokens compared to DPO. Fig. 2 shows part of a solution for a GSM8K question. In step 2, the solution incorrectly interprets "4 less than a dozen" as "$4 \times (12 - 4)$", when it should have been "$(12 - 4)$". The SCDPO model correctly highlights "$4 \times (12 - 4)$", while the DPO does not. Fig. 3 shows part of a solution for a MATH question. The solution sums the terms in the expression when two of the terms should have been multiplied. SCDPO correctly highlights the incorrect solution, while DPO does not. These examples show that the stepwise supervision provided in SCDPO results in a better token-level understanding of reasoning errors.

5 Experiments

In this section, we first perform a comprehensive empirical comparison between SCDPO and DPO on three kinds of Mistral-7B SFT models. Then, we increase the data used in SFT, DPO, and SCDPO training, using InternLM2-20B as the foundation model, demonstrating the great potential of our method.

5.1 Comparison using 7B Models

Baseline Models. We introduce three baseline SFT models: Mistral-7B-Ours, MetaMath-Mistral-7B, and MathCoder-Mistral-7B. All three SFT models use Mistral-7B as the foundation model. Mistral-7B-Ours is finetuned with a math problem-solution dataset we created by collecting multiple solutions from the GPT-4 Code Interpreter for each problem in the GSM8K and MATH training sets and retaining those reaching the correct final answer. This SFT dataset contains 34K question-solution pairs from GSM8K, and 47K from MATH. MetaMath-Mistral-7B is downloaded from the MetaMath HuggingFace repository[1]. The model is reported to have been trained on the 395K MetaMathQA

[1]https://huggingface.co/meta-math/MetaMath-Mistral-7B

Table 1: Effect of using Step-Controlled DPO (SCDPO) on three different SFT models: a Mistral-7B model finetuned with code-integrated solutions we collected from GPT-4 Code Interpreter, the MetaMath-Mistral-7B model, and MathCoder-Mistral-7B model we finetuned using the MathCoderInstruct dataset, compared to DPO. "(data-equal)" denote the DPO baseline using the same amount of data as SCDPO. "GS" and "MA" are short for GSM8K and MATH respectively.

Method	**GSM8K**	**MATH**	**Data**				
			$\mathbf{GS_{dpo}}$	$\mathbf{MA_{dpo}}$	$\mathbf{GS_{scdpo}}$	$\mathbf{MA_{scdpo}}$	**SFT**
Mistral-7B-Ours							
SFT	76.8%	43.2%	-	-	-	-	81K
DPO	78.8%	45.1%	7K	5K	-	-	-
$\text{DPO}_{\text{(data-equal)}}$	79.0%	45.7%	13K	17K	-	-	-
SCDPO	**80.1%**	**47.7%**	7K	5K	6K	12K	-
MetaMath-Mistral-7B							
SFT	77.7%	28.2%	-	-	-	-	395K
DPO	81.0%	28.7%	7K	6K	-	-	-
$\text{DPO}_{\text{(data-equal)}}$	81.4%	29.0%	13K	17K	-	-	-
SCDPO	**81.7%**	**29.3%**	7K	6K	6K	11K	-
MathCoder-Mistral-7B							
SFT	78.1%	39.3%	-	-	-	-	80K
DPO	79.2%	42.9%	6K	6K	-	-	-
$\text{DPO}_{\text{(data-equal)}}$	78.3%	41.1%	12K	19K	-	-	-
SCDPO	**80.4%**	**43.3%**	6K	6K	6K	13K	-

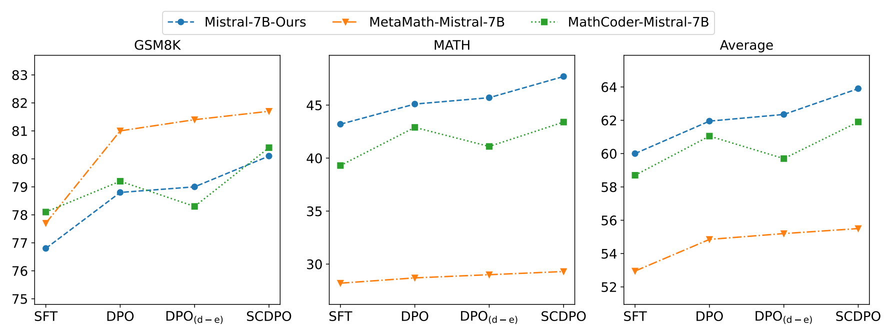

Figure 4: Comparison between SCDPO and DPO. On all three models, the SCDPO method achieves the best performance. Here "(d-e)" means data-equal, denoting DPO using the same amount of data as SCDPO.

dataset [47], and we do not do any further SFT training on it. MathCoder-Mistral-7B is finetuned using the MathCodeInstruct dataset [35], downloaded from HuggingFace[2].

Implementation Details. The supervised finetuning of Mistral-7B-Ours and MathCoder-Mistral-7B is conducted with a learning rate of 1.0×10^{-5} for 3 epochs, with a context length of 2048 tokens. DPO and SCDPO of Mistral-7B-Ours and MathCoder-Mistral-7B are trained with a learning rate of 1.0×10^{-7} for 2 epochs, with a context length of 1024 tokens and β set as 0.1. DPO and SCDPO of MetaMath-Mistral-7B are trained with a learning rate of 1.0×10^{-7} for 2 epochs, with a context length of 2048 tokens and β set as 0.5. The details of data composition are shown in Tab. 1. The training code is implemented based on HuggingFace's alignment-handbook repository[3]. The models are trained on 8 NVIDIA A800 80GB GPUs with a batch size of 64.

Comparison between SCDPO and DPO. The results of SFT, DPO, and SCDPO on GSM8K and MATH are shown in Tab. 1. We perform two DPO experiments, with different amounts of training

[2] `https://huggingface.co/datasets/MathLLMs/MathCodeInstruct`

[3] `https://github.com/huggingface/alignment-handbook`

Table 2: Performance of open-source and closed-source models on two English datasets, GSM8K and MATH, and three Chinese datasets, APE210K, CMATH, and MGSM-zh. All results reported are based on greedy decoding. The best models are marked in **bold**, and the second best models are underlined.

Model	Size	English		Chinese		
		GSM8K	MATH	APE210K	CMATH	MGSM-zh
Closed-Source Models						
GPT-3.5	-	80.8%	34.1%	-	73.8%	-
GPT-4 [22]	-	93.6%	53.6%	84.2%	**89.3%**	-
GPT-4 Code Interpreter [53]	-	**97.0%**	**69.7%**	-	-	-
GLM-4 [4]	-	91.8%	49.0%	**93.5%**	89.0%	-
Baichuan-3	-	88.2%	49.2%	-	-	-
Open-Source Models						
Math-Shepherd [36]	7B	84.1%	33.0%	-	-	-
SeaLLM-v2 [21]	7B	78.2%	27.5%	-	-	64.8%
DeepSeekMath-RL [29]	7B	86.7%	**58.8%**	71.9%	87.6%	78.4%
Skywork-13B-Math [42]	13B	72.3%	17.0%	74.4%	77.3%	-
InternLM2-Math [46]	20B	80.7%	54.3%	-	-	-
MathGenie [19]	20B	87.7%	55.7%	-	-	-
ChatGLM3-32B-RFT-DPO [41]	32B	82.6%	40.6%	**89.4%**	85.6%	-
Yi-Chat [45]	34B	76.0%	15.9%	65.1%	77.7%	-
ToRA [12]	34B	80.7%	50.8%	-	53.4%	41.2%
MAmmoTH [50]	70B	76.9%	41.8%	-	-	-
MathCoder [35]	70B	83.9%	45.1%	-	-	-
WizardMath-v1.0 [20]	70B	81.6%	22.7%	-	65.4%	64.8%
Qwen [3]	72B	78.9%	35.2%	77.1%	88.1%	-
InternLM2-SFT	20B	86.4%	55.8%	77.1%	88.4%	74.8%
InternLM2-SFT-DPO	20B	87.0%	57.6%	78.7%	89.9%	76.0%
InternLM2-SFT-DPO$_{\text{(data equal)}}$	20B	88.2%	57.5%	78.8%	89.3%	76.0%
InternLM2-SFT-SCDPO	20B	**88.5%**	58.1%	79.3%	**90.3%**	**80.4%**

data. One is trained with the D_{naive} part (as explained in Sec. 3.1) of the SCDPO training data. The other, DPO$_{\text{(data-equal)}}$, is trained on data expanded from D_{naive} to include more preferred-dispreferred DPO training pairs, resulting in a training dataset consisting of approximately the same amount of samples as SCDPO's training dataset. This is to rule out the possibility that the performance gain of SCDPO is the effect of more training samples. As demonstrated in Tab 1 and Fig. 4, on all three SFT baseline models, SCDPO shows superior performance compared to DPO. This can be attributed to SCDPO's more detailed supervision on the reasoning steps of the math solutions, demonstrating the effectiveness of our method.

5.2 Scaling of Data Amount on 20B Model

Training Data. We increase the amount of SFT training data by collecting solutions for questions in the training set of APE210K [51] from GPT-4 Code Interpreter. APE210K is a dataset containing high-quality Chinese math word problems. After removing the solutions that reach incorrect final answers, we get 169K question-solution pairs. Combining the newly collected data with the original 34K GSM8K data and 47K MATH data, we get an SFT dataset of 250K question-solution pairs. The SCDPO and DPO training data is collected as described before in Sec. 3.1.

Training Settings. We use InternLM2-20B [6] as the foundation model, as it has demonstrated high performance in previous works [19, 6], even surpassing larger models such as Mixtral-8x7B [14] and Llama2-70B [32] in some cases. In the SFT stage, we finetune the model with a learning rate of 1.0×10^{-5} for 3 epochs, with a context length of 2048 tokens. In DPO and SCDPO training, we use a learning rate of 1.5×10^{-7} to train the SFT model for 2 epochs, with a context length of 1024 and β set to 0.1. The models are trained on 16 NVIDIA A800 80GB GPUs with a batch size of 64.

Evaluation Datasets. Five representative mathematical datasets are used in evaluating the models: GSM8K [9], MATH [13], APE210K [51], CMATH [39], and MGSM-zh [30]. GSM8K and MATH consist of English math questions, while APE210K, CMATH, and MGSM-zh are consisted of Chinese math questions. The evaluation datasets contain a wide range of problem types, covering mathematical problems from grade-school level to college level, comprehensively evaluating the models' mathematical reasoning abilities. We use greedy decoding for all evaluations.

Baselines. We compare our 20B models with powerful closed-source models such as GPT-3.5 [5], GPT-4 [22], GPT-4 Code Interpreter [22], GLM-4 [5], and Baichuan-3 [6], as well as open-source models such as DeepSeekMath-RL [29], Math-Shepherd [36], SeaLLM-v2 [21], Skywork-13B-Math [42], InternLM2-Math [7] [46], MathGenie [19], ChatGLM3-32B-RFT-DPO [41], Yi-Chat [45], ToRA [12], MAmmoTH [50], MathCoer [35], WizardMath [20], and Qwen [3].

Main Results. Tab. 2 displays our main results, as well as various closed-source and open-source baselines. Our model achieves a score of 88.5% on GSM8K, 90.3% on CMATH, and 80.4% on MGSM-zh, surpassing all models with published parameters, and obtaining second-best scores among open-source models on MATH and APE210K, with a score of 58.1% on MATH and 79.3% on APE210K. While our model rivals the performance of GPT-3.5 and Baichuan-3 on GSM8K and MATH, and surpasses GPT-4 and GLM-4 on MATH, it still underperforms GPT-4 Code Interpreter on GSM8K and MATH, and GLM-4 on APE210K.

Compared to InternLM2-SFT, InternLM2-SFT-SCDPO consistently increases the score on each of the five datasets by approximately 2% to 3%. Compared to both InternLM2-SFT-DPO, which uses the D_{naive} part of InternLM2-SFT-SCDPO's training data, and InternLM2-SFT-DPO$_{\text{(data-equal)}}$, which uses about the same amount of training data as InternLM2-SFT-SCDPO, InternLM2-SFT-SCDPO consistently achieves the best performance across all five datasets, highlighting the effectiveness of SCDPO in enhancing mathematical problem-solving abilities.

6 Limitations and Future Work

Our work contains the following limitations, and we leave them for future work. Firstly, our work is conducted on purely linguistic models, which struggle to solve mathematical problems requiring an understanding of images. For example, questions in the geometry subject of the MATH dataset exhibit lower accuracy compared to questions in other subjects. A possible solution would be to utilize multimodal techniques, to produce models that can be evaluated with multimodal reasoning datasets [18, 34]. Secondly, due to the stepwise attribute of SCDPO, it is not very effective on solution formats consisting of pure code. It only works on solutions consisting of natural language chain of thought or interleaved natural language and code. A method to properly enhance pure code solutions needs to be derived, which we leave for future work. Thirdly, as with all language models, our models can potentially generate hallucinations or produce misleading solutions, which can have a negative effect. This can be mitigated with methods such as verification, which we also leave for future work.

7 Conclusion

In this work, we propose Step-Controlled DPO (SCDPO), a method to automatically introduce stepwise error supervision to the process of DPO training by generating dispreferred samples that start making errors at a specified step. SCDPO effectively enhances the mathematical reasoning abilities of LLMs. We conduct experiments on three different 7B SFT models, consistently improving the models' performance on mathematical problem-solving tasks and demonstrating the effectiveness and robustness of our method. The 20B model trained with SCDPO on both English and Chinese data achieves the highest score among open-source models on GSM8K, CMATH and MGSM-zh, and second-best score on MATH and APE210K, demonstrating the significant potential of our method.

[5] `https://open.bigmodel.cn/dev/api#glm-4`

[6] `https://www.baichuan-ai.com`

[7] `https://github.com/InternLM/InternLM-Math`

References

[1] M. G. Azar, Z. D. Guo, B. Piot, R. Munos, M. Rowland, M. Valko, and D. Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR, 2024.

[2] Z. Azerbayev, H. Schoelkopf, K. Paster, M. D. Santos, S. McAleer, A. Q. Jiang, J. Deng, S. Biderman, and S. Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023.

[3] J. Bai, S. Bai, et al. Qwen technical report, 2023.

[4] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Z. Cai, M. Cao, H. Chen, K. Chen, K. Chen, X. Chen, X. Chen, Z. Chen, Z. Chen, P. Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024.

[7] W. Chen, X. Ma, X. Wang, and W. W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022.

[8] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[9] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[10] K. Ethayarajh, W. Xu, N. Muennighoff, D. Jurafsky, and D. Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

[11] L. Gao, A. Madaan, S. Zhou, U. Alon, P. Liu, Y. Yang, J. Callan, and G. Neubig. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR, 2023.

[12] Z. Gou, Z. Shao, Y. Gong, Y. Yang, M. Huang, N. Duan, W. Chen, et al. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*, 2023.

[13] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[14] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[15] J. Kreutzer, J. Uyheng, and S. Riezler. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning. *arXiv preprint arXiv:1805.10627*, 2018.

[16] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

[17] T. Liu, Y. Zhao, R. Joshi, M. Khalman, M. Saleh, P. J. Liu, and J. Liu. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*, 2023.

[18] P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, and J. Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

[19] Z. Lu, A. Zhou, H. Ren, K. Wang, W. Shi, J. Pan, M. Zhan, and H. Li. Mathgenie: Generating synthetic data with question back-translation for enhancing mathematical reasoning of llms. *arXiv preprint arXiv:2402.16352*, 2024.

[20] H. Luo, Q. Sun, C. Xu, P. Zhao, J. Lou, C. Tao, X. Geng, Q. Lin, S. Chen, and D. Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

[21] X. Nguyen, W. Zhang, X. Li, M. M. Aljunied, Q. Tan, L. Cheng, G. Chen, Y. Deng, S. Yang, C. Liu, H. Zhang, and L. Bing. Seallms - large language models for southeast asia. *CoRR*, abs/2312.00738, 2023.

[22] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

[23] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[24] A. Pal, D. Karkhanis, S. Dooley, M. Roberts, S. Naidu, and C. White. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.

[25] R. Rafailov, J. Hejna, R. Park, and C. Finn. From r to q: Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*, 2024.

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

[27] R. Ramamurthy, P. Ammanabrolu, K. Brantley, J. Hessel, R. Sifa, C. Bauckhage, H. Hajishirzi, and Y. Choi. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*, 2022.

[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[29] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. Li, Y. Wu, and D. Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[30] F. Shi, M. Suzgun, M. Freitag, X. Wang, S. Srivats, S. Vosoughi, H. W. Chung, Y. Tay, S. Ruder, D. Zhou, D. Das, and J. Wei. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[31] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

[32] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[33] L. Tunstall, E. Beeching, N. Lambert, N. Rajani, K. Rasul, Y. Belkada, S. Huang, L. von Werra, C. Fourrier, N. Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

[34] K. Wang, J. Pan, W. Shi, Z. Lu, M. Zhan, and H. Li. Measuring multimodal mathematical reasoning with math-vision dataset. *arXiv preprint arXiv:2402.14804*, 2024.

[35] K. Wang, H. Ren, A. Zhou, Z. Lu, S. Luo, W. Shi, R. Zhang, L. Song, M. Zhan, and H. Li. Mathcoder: Seamless code integration in llms for enhanced mathematical reasoning. *arXiv preprint arXiv:2310.03731*, 2023.

[36] P. Wang, L. Li, Z. Shao, R. Xu, D. Dai, Y. Li, D. Chen, Y. Wu, and Z. Sui. Math-shepherd: A label-free step-by-step verifier for llms in mathematical reasoning. *arXiv preprint arXiv:2312.08935*, 2023.

[37] Z. Wang, R. Xia, and P. Liu. Generative ai for math: Part i–mathpile: A billion-token-scale pretraining corpus for math. *arXiv preprint arXiv:2312.17120*, 2023.

[38] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[39] T. Wei, J. Luan, W. Liu, S. Dong, and B. Wang. Cmath: Can your language model pass chinese elementary school math test?, 2023.

[40] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 2004.

[41] Y. Xu, X. Liu, X. Liu, Z. Hou, Y. Li, X. Zhang, Z. Wang, A. Zeng, Z. Du, W. Zhao, et al. Chatglm-math: Improving math problem-solving in large language models with a self-critique pipeline. *arXiv preprint arXiv:2404.02893*, 2024.

[42] L. Yang, H. Yang, W. Cheng, L. Lin, C. Li, Y. Chen, L. Liu, J. Pan, T. Wei, B. Li, L. Zhao, L. Wang, B. Zhu, G. Li, X. Wu, X. Luo, and R. Hu. Skymath: Technical report, 2023.

[43] Z. Yang, M. Ding, Q. Lv, Z. Jiang, Z. He, Y. Guo, J. Bai, and J. Tang. Gpt can solve mathematical problems without a calculator. *arXiv preprint arXiv:2309.03241*, 2023.

[44] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[45] Yi. A series of large language models trained from scratch by developers at 01-ai. `https://github.com/01-ai/Yi`, 2023.

[46] H. Ying, S. Zhang, L. Li, Z. Zhou, Y. Shao, Z. Fei, Y. Ma, J. Hong, K. Liu, Z. Wang, et al. Internlm-math: Open math large language models toward verifiable reasoning. *arXiv preprint arXiv:2402.06332*, 2024.

[47] L. Yu, W. Jiang, H. Shi, J. Yu, Z. Liu, Y. Zhang, J. T. Kwok, Z. Li, A. Weller, and W. Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

[48] W. Yuan, R. Y. Pang, K. Cho, S. Sukhbaatar, J. Xu, and J. Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

[49] Z. Yuan, H. Yuan, C. Li, G. Dong, C. Tan, and C. Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

[50] X. Yue, X. Qu, G. Zhang, Y. Fu, W. Huang, H. Sun, Y. Su, and W. Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023.

[51] W. Zhao, M. Shang, Y. Liu, L. Wang, and J. Liu. Ape210k: A large-scale and template-rich dataset of math word problems, 2020.

[52] Y. Zhao, R. Joshi, T. Liu, M. Khalman, M. Saleh, and P. J. Liu. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.

[53] A. Zhou, K. Wang, Z. Lu, W. Shi, S. Luo, Z. Qin, S. Lu, A. Jia, L. Song, M. Zhan, et al. Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification. *arXiv preprint arXiv:2308.07921*, 2023.

[54] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

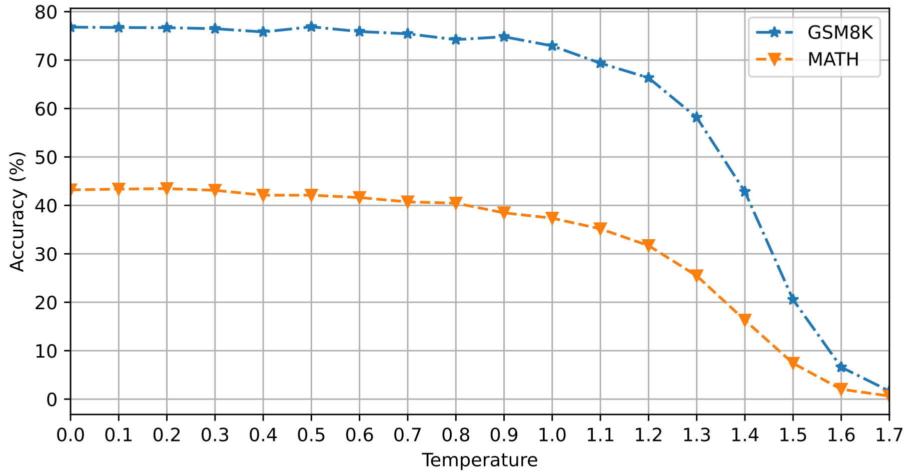

Figure 5: Accuracy of Mistral-7B-Ours (SFT) on GSM8K and MATH when temperature is set at different values.

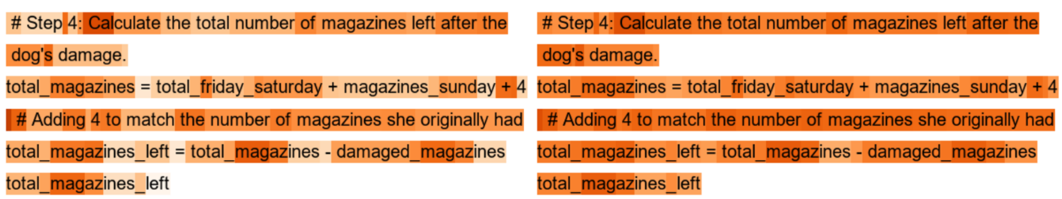

Figure 6: Credit assignment of part of a solution for a GSM8K problem. Each token is colored corresponding to the DPO implicit reward as expressed in Eq. 7 (darker is higher). The left is the credit assignment of SCDPO, which correctly highlighted the error – the number of damaged magazines (which is 4) should not be first added to and then extracted from "total_magazines", while the credit assignment of DPO on the right fails to highlight it.

A Pilot Study Regarding Correlation Between Temperature and Accuracy

In this section, we provide the results of the SFT version of Mistral-7B-Ours on GSM8K and MATH when temperature is set to different values (from 0.0 to 1.7) as a pilot study. As demonstrated in Fig. 5, with the increase of temperature, accuracy shows a trend of decreasing. When the temperature is between 0.0 and 1.0, the accuracy is relatively stable. When the temperature is higher than 1.0, accuracy on the two datasets starts to degrade, as errors are more likely to occur. This rise in the occurrence of errors can be used to create erroneous reasoning steps in SCDPO.

B Further Credit Assignment Analysis Examples

In this section, we present several credit assignment analysis examples, comparing SCDPO to DPO. Fig. 6, Fig. 7 and Fig. 8 show examples of part of the solutions of questions taken from GSM8K and MATH datasets, colored with the DPO implicit reward of each token (darker is higher). As demonstrated in the examples, SCDPO is better than DPO at identifying the errors in the reasoning steps.

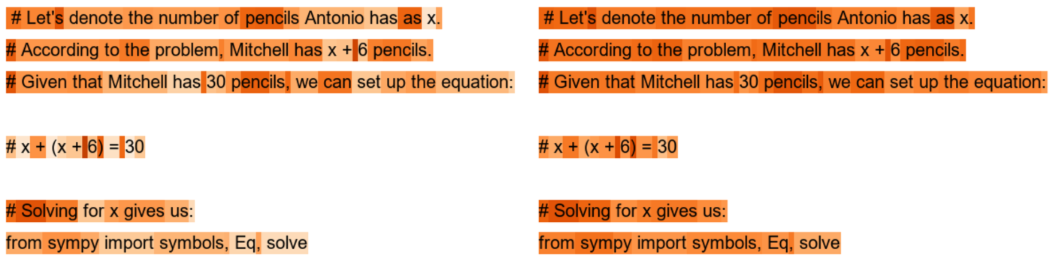

Figure 7: Credit assignment of part of a solution for a GSM8K problem. Each token is colored corresponding to the DPO implicit reward as expressed in Eq. 7 (darker is higher). The left is the credit assignment of SCDPO, which correctly highlighted the error – Mitchell has 30 pencils, and Antonio has 6 less pencils than Michell, which is $30 - 6$, so the introduction of x is not needed, and $x + (x + 6) = 30$ is incorrect, while the credit assignment of DPO on the right fails to highlight it.

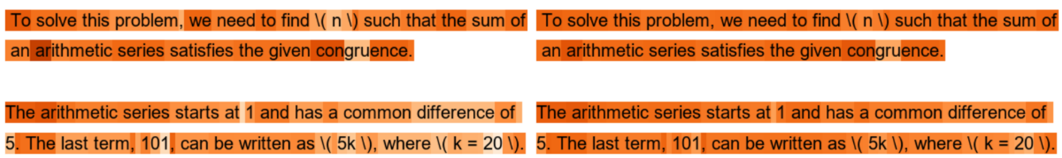

Figure 8: Credit assignment of part of a solution for a MATH problem. Each token is colored corresponding to the DPO implicit reward as expressed in Eq. 7 (darker is higher). The left is the credit assignment of SCDPO, which correctly highlighted the error – 101 cannot be written as $5k$ where $k = 20$, while the credit assignment of DPO on the right fails to highlight the error.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work is discussed in the Limitations and Future Work section of the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: For each theoretical result, the paper provide the full set of assumptions and a complete proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We disclose all the information needed to reproduce the main experimenta results of the paper in the Experiments section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We plan to provide open access to the data and code upon acceptance of the paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details in the Experiments section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported because it would be too computationally expensive for LLM training.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computer resources needed to reproduce the experiments are explained in the Experiments section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Potential positive societal impacts and negative societal impacts of the work performed are discussed in the Limitations and Future Work section.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work is limited to mathematical problem solving and contains no pretraining or scraped datasets. So the risk for misuse is small.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, are properly credited and the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The new assets introduced in the paper are not released upon submission, and will only be released upon the acceptance of the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.