SUPPLEMENTARY MATERIAL FOR STEP-DPO: STEP-WISE PREFERENCE OPTIMIZATION FOR LONG-CHAIN REASONING OF LLMS

Anonymous authors
Paper under double-blind review

1 DETAILS OF DATA CONSTRUCTION

In Sec. 3.2 of the main submission, we introduce the data construction pipeline for Step-DPO. In this section, we provide additional details for the step localization phase. Also, we introduce the details of further data filtering.

Step localization. In this process, we use GPT-4o to localize the erroneous reasoning step. Given a math problem with its correct solution and an incorrect answer, the prompt for GPT-4o is shown in Table. 1.

Table 1: GPT-4o prompt to localize erroneous reasoning step in incorrect answers.

```
### Problem:
{problem}
### Correct solution:
{solution}
### Incorrect answer:
{answer}

—

A math problem and its correct solution are listed above. We also give another incorrect answer,
where step-by-step reasoning process is shown. Please output the correctness for each reasoning step
in the given answer.

Requirements:
1. You should first output a step-by-step analysis process (no more than 200 words), and finally output
the decision (”correct”, ”neutral”, ”incorrect”) for each step following the format of ”Final Decision:
Step 1: correct; Step 2: neutral; ...”;
2. Stop when you find the first incorrect step.
```

Further data filtering. As described in Sec. 3.2 of the main submission, there exists the case where the final answer is correct but the intermediate reasoning steps are incorrect. When formulating the chosen step, we need to avoid such cases. We employ GPT-4o for filtering. The prompt is shown in Table. 2.

Table 2: GPT-4o prompt for further data filtering.

Problem:
{problem}
Correct solution:
{solution}
Given answer:
{answer}

—

A math problem and its correct solution are listed above. We also give another answer, where step-by-step reasoning process is shown. Please output the correctness for each reasoning step in the given answer.

Requirement:
You should first output a step-by-step analysis process (no more than 200 words), and finally output the decision ("correct", "neutral", "incorrect") for each step following the format of "Final Decision: Step 1: correct; Step 2: neutral; ...".

2 More Examples

As shown in Fig. 1, we show additional comparisons between Qwen2-72B-Instruct and the fine-tuned version with Step-DPO. They demonstrate that Step-DPO could refrain from the previous errors, thus facilitating the holistic reasoning chains.

3 Details of the Step-DPO vs. DPO Experiments

The comparison between Step-DPO and DPO is shown in Fig. 2 of the main submission. Specifically, to calculate the accuracy of judging preferred or undesirable outputs, we input the math problem, the preceding reasoning steps, and also the next reasoning step (both preferred and undesirable ones) into the models, and compute the implicit rewards respectively. The judgement is counted as correct, if the reward of the preferred next reasoning step is higher than that of the undesirable one. As for the reward margin, we simply compute the gap between the rewards.

4 Details of Gradient Decay Issue

According to Sec. 3.1 of the main submission, the optimization objective for Step-DPO is formulated in equation 2 of the main submission. For simplicity, we use the prompt $p = [x; s_{1\sim k-1}]$ as a whole to rewrite the original equation as:

$$\mathcal{L}(\theta) = -\mathbb{E}_{(p,s_{win},s_{lose})\sim D}[\log \sigma(\beta \log \frac{\pi_\theta(s_{win}|p)}{\pi_{ref}(s_{win}|p)} - \beta \log \frac{\pi_\theta(s_{lose}|p)}{\pi_{ref}(s_{lose}|p)})]. \tag{1}$$

Let's move one step further to see the gradient with respect to the parameters θ as follows.

$$\begin{aligned}\nabla_\theta \mathcal{L}(\theta) = &- \mathbb{E}_{(p,s_{win},s_{lose})\sim D} \\ &[\beta\sigma(\hat{r}_\theta(p, s_{lose}) - \hat{r}_\theta(p, s_{win}))[\nabla_\theta \log \pi_\theta(s_{win}|p) - \nabla_\theta \log \pi_\theta(s_{lose}|p)]]\end{aligned} \tag{2}$$

where $\hat{r}_\theta(p, s) = \beta \log \frac{\pi_\theta(s|p)}{\pi_{ref}(s|p)} = \beta(\log \pi_\theta(s|p) - \log \pi_{ref}(s|p))$ is the implicit reward function. We empirically observe that the log-probability of an out-of-distribution output $\log \pi_{ref}(s^{ood}|p) \approx -100$, whereas that of an in-distribution output $\log \pi_{ref}(s^{id}|p) \approx -10$.

However, if we use an out-of-distribution preferred output as s_{win}. Since the undesirable output is always in-distribution, then we have $\log \pi_{ref}(s^{ood}_{win}|p) \approx -100$ and $\log \pi_{ref}(s^{id}_{lose}|p) \approx -10$. So, we

have

$$\begin{aligned}\hat{r}_\theta(p, s^{id}_{lose}) - \hat{r}_\theta(p, s^{ood}_{win}) &= \beta(\log \pi_\theta(s^{id}_{lose}|p) - \log \pi_{ref}(s^{id}_{lose}|p)) - \beta(\log \pi_\theta(s^{ood}_{win}|p) - \log \pi_{ref}(s^{ood}_{win}|p)) \\ &\approx \beta(\log \pi_\theta(s^{id}_{lose}|p) - \log \pi_\theta(s^{ood}_{win}|p) - 90).\end{aligned} \tag{3}$$

If $\pi_\theta(s^{id}_{lose}|p) < \pi_\theta(s^{ood}_{win}|p)$ for the final policy model after training, we have $\log \pi_\theta(s^{id}_{lose}|p) - \log \pi_\theta(s^{ood}_{win}|p) < 0$. So, we have

$$\hat{r}_\theta(p, s^{id}_{lose}) - \hat{r}_\theta(p, s^{ood}_{win}) < -90\beta \tag{4}$$

Then the scale of the gradient (shown in equation 2) becomes

$$\beta\sigma(\hat{r}_\theta(p, s^{id}_{lose}) - \hat{r}_\theta(p, s^{ood}_{win})) < \beta\sigma(-90\beta) = \beta \cdot \frac{1}{1 + e^{90\beta}} < 0.004. \tag{5}$$

So, the gradient would have decayed to nearly all zeros when $\pi_\theta(p, s^{id}_{lose}) < \pi_\theta(p, s^{ood}_{win})$. Actually, when $\pi_\theta(p, s^{id}_{lose})$ is far higher than $\pi_\theta(p, s^{ood}_{win})$, the gradient already decays, and it is hard to reach the situation where $\pi_\theta(p, s^{id}_{lose}) < \pi_\theta(p, s^{ood}_{win})$. In other words, the out-of-distribution nature of the data generated by GPT-4o or human hinders the optimization of the DPO objective. Overall, the use of in-distribution matters.

Problem:
Suppose that $h(x) = f^{-1}(x)$. If $h(2) = 10$, $h(10) = 1$ and $h(1) = 2$, what is $f(f(10))$?

Qwen2-72B-Instruct :
Given that $h(x) = f^{-1}(x)$, we can use the properties of inverse functions to find $f(f(10))$.
First, recall that if $h(x) = f^{-1}(x)$, then $f(h(x)) = x$ for all x in the domain of h. This is because $f^{-1}(x)$ is defined as the function that "undoes" the effect of $f(x)$.
Given:
1. $h(2) = 10$, 2. $h(10) = 1$, and 3. $h(1) = 2$.
We want to find $f(f(10))$.
From $h(10) = 1$, we can use the property of inverse functions to say that $f(h(10)) = f(1)$. But since $h(10) = 1$, this means $f(1) = 10$.
Now, we need to find $f(f(10))$. Since we know $f(1) = 10$, we can substitute 1 for $f(10)$ (because $h(10) = 1$ and $f(h(10)) = f(1)$). Therefore, $f(f(10)) = f(1) = 10$.
So, $f(f(10)) = \boxed{10}$

Qwen2-72B-Instruct + Step-DPO:
Given that $h(x) = f^{-1}(x)$, we can use the properties of inverse functions to find $f(f(10))$. The key property of inverse functions is that $f(f^{-1}(x)) = x$ and $f^{-1}(f(x)) = x$ for all x in the domain of the respective functions.
From the given information, we have:
1. $h(2) = 10$, which means $f^{-1}(2) = 10$. By the property of inverse functions, this implies $f(10) = 2$.
2. $h(10) = 1$, which means $f^{-1}(10) = 1$. By the property of inverse functions, this implies $f(1) = 10$.
3. $h(1) = 2$, which means $f^{-1}(1) = 2$. By the property of inverse functions, this implies $f(2) = 1$.
We are asked to find $f(f(10))$. From the information above, we know that $f(10) = 2$. Therefore, $f(f(10)) = f(2)$. From the information given, we know that $f(2) = 1$.
Thus, $f(f(10)) = 1$.
The answer is $\boxed{1}$.

Problem:
Given the data set $\{3, 7, 7, 2, 5\}$, calculate the sample mean μ and the sample standard deviation σ. Present the answer as $[\mu, \sigma]$.

Qwen2-72B-Instruct :
To find the sample mean μ and the sample standard deviation σ for the data set $\{3, 7, 7, 2, 5\}$, we follow these steps:
Step 1: Calculate the Sample Mean μ
The sample mean μ is calculated by summing all the values in the data set and dividing by the number of values.
Given data set: $\{3, 7, 7, 2, 5\}$
$$\mu = \frac{3+7+7+2+5}{5} = \frac{24}{5} = 4.8$$
Step 2: Calculate the Sample Standard Deviation σ
The sample standard deviation σ is calculated by finding the square root of the average of the squared differences from the Mean.
First, find the squared differences from the mean for each value:
$(3-4.8)^2 = 3.24$, $(7-4.8)^2 = 4.84$, $(7-4.8)^2 = 4.84$, $(2-4.8)^2 = 7.84$, $(5-4.8)^2 = 0.04$
Then, find the average of these squared differences:
$$\text{Average} = \frac{3.24 + 4.84 + 4.84 + 7.84 + 0.04}{5} = \frac{20.8}{5} = 4.16$$
Finally, take the square root of this average to get the sample standard deviation:
$$\sigma = \sqrt{4.16} \approx 2.04$$
Answer:
The sample mean μ and the sample standard deviation σ for the data set $\{3, 7, 7, 2, 5\}$ are approximately $[\mu, \sigma] = [4.8, 2.04]$.
$$\boxed{[4.8, 2.04]}$$

Qwen2-72B-Instruct + Step-DPO:
To find the sample mean μ and the sample standard deviation σ for the data set $\{3, 7, 7, 2, 5\}$, we follow these steps:
1. **Calculate the sample mean μ:**
The sample mean is the sum of all the values divided by the number of values.
$$\mu = \frac{3 + 7 + 7 + 2 + 5}{5} = \frac{24}{5} = 4.8$$
2. **Calculate the sample standard deviation σ:**
The sample standard deviation is calculated using the formula:
$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{n-1}}$$
where x_i are the individual values, μ is the mean, and n is the number of values.
First, calculate the squared deviations from the mean:
$(3-4.8)^2 = 3.24$, $(7-4.8)^2 = 4.84$, $(7-4.8)^2 = 4.84$, $(2-4.8)^2 = 7.84$, $(5-4.8)^2 = 0.04$
Then, sum these squared deviations and divide by $n-1$:
$$\sigma = \sqrt{\frac{3.24 + 4.84 + 4.84 + 7.84 + 0.04}{5-1}} = \sqrt{\frac{20.8}{4}} = \sqrt{5.2} \approx 2.28$$
Therefore, the sample mean μ and the sample standard deviation σ are approximately $[4.8, 2.28]$.
The answer is $\boxed{[4.8, 2.28]}$.

Figure 1: Examples of comparison between Qwen2-72B-Instruct and Qwen2-72B-Instruct-Step-DPO.