Instruction Bootstrapped Preference Optimization: Improving Model Alignment with a *Better* Instruction

Anonymous ACL submission

Abstract

Instruction tuning and preference alignment have played pivotal roles in recent advances in large language models (LLMs). Empirical observations reveal that when provided with bootstrapping instructions such as "**`please generate a better response`**" following initial outputs, these models can produce significantly enhanced subsequent responses. This finding highlights the critical role of both initial outputs and bootstrapping instructions in preference alignment, while also suggesting the important connection between abstract preference definitions and their concrete textual expressions. Based on this insight, we propose Instruction Bootstrapped Preference Optimization (IBPO), an innovative approach to refine instruction fine-tuning, preference optimization, and inference steps in LLMs in the form of plugins. IBPO systematically integrates paired preference data with bootstrapping instructions into unified sequences, enabling more effective utilization of preference data while strengthening the association between textual expressions in preference data and preference descriptions in the instruction. Experiments on multiple datasets demonstrate that IBPO achieves more than 10% improvements over several existing preference alignment baselines. Ablation experiments and mechanistic analysis provide potential explanations for these improvements.

1 Introduction

Recent breakthroughs in large language models (LLMs) have demonstrated remarkable capabilities and achieved impressive performance across tasks ranging from machine translation (Hendy et al., 2023) to code generation (Ni et al., 2023). A key factor driving these advances is the integration of instruction tuning and preference alignment (Zhao et al., 2023). While base models pretrained on vast corpora demonstrate broad capabilities, these techniques specifically enhance their ability to interpret human intent and generate outputs that are accurate, contextually coherent, and aligned with ethical constraints (Wang et al., 2023). Early alignment approaches like Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) relied on reward modeling and reinforcement learning, while subsequent methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2024) eliminated the need for explicit reward models through supervised optimization. This evolution has spurred numerous refined techniques (Azar et al., 2024; Xu et al., 2024a; Ethayarajh et al., 2024) addressing diverse alignment challenges.

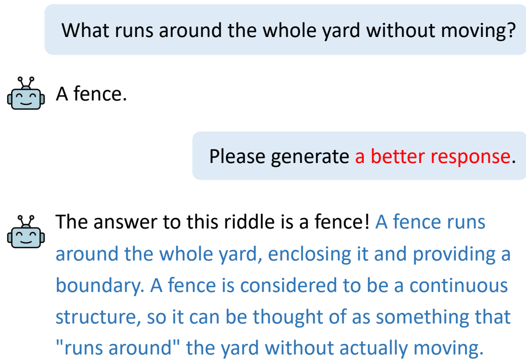

Figure 1: LLM augments simple initial response with detailed explanations after receiving the bootstrapping instruction. The abstract preference descriptions and concrete textual expressions are marked in red and blue.

Among the emerging capabilities of aligned LLMs, we particularly pay attention to the "bootstrapping" ability. Specifically, when an LLM provides an initial response to a query, subsequent instructions such as "`please generate a better response`" enable the model to produce a substantially enhanced output. For example, as shown in Figure 1, despite being instructed by only a general expectation to improve, the model successfully augments the simple initial response with detailed explanations. This bootstrapping phenomenon has been applied in research and practice, including

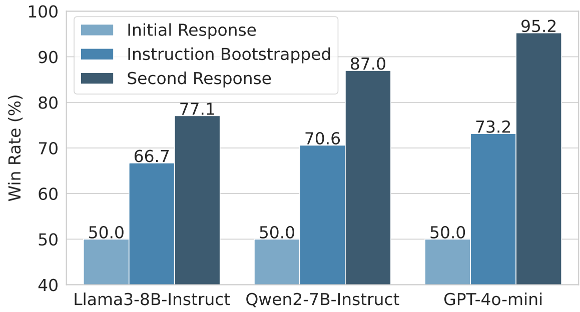

Figure 2: Appending the bootstrapping instruction to the original question as a suffix result in responses better than initial responses, but inferior to second responses.

data augmentation (Madaan et al., 2023; Liu et al., 2024a) and code refinement (Woolf, 2025), but its critical factors and underlying mechanisms remain unrevealed in existing work.

To address this gap, we conduct experiments on three models with results illustrated in Figure 2. First, using a bootstrapping instruction after the initial response lead to significant improvements, highlighting the importance of the initial response. When incorporated into the context, these responses serve as effective references that guide the subsequent generation. Moreover, appending the bootstrapping instruction directly to the original question without a second generation still result in marked improvements, although inferior to the two-step generations, which underscores the pivotal role of the instruction itself. Although previous studies emphasize the benefits of specific instructions (Madaan et al., 2023), our experiments demonstrate that even generic instruction consistently elicits quality improvements. This suggests that aligned LLMs have established a connection between abstract preferences descriptions in bootstrapping instructions and concrete textual expressions (marked red and blue in Figure 1).

Building on these insights, we propose Incremental Bootstrapping Preference Optimization (IBPO), an innovative method that enhances LLM alignment in the form of plugins throughout instruction tuning, preference optimization, and inference. IBPO introduces two key innovations: 1) integration of chosen and rejected responses into joint sequences as contextual references and training targets, enhancing the utilization of paired preference data; 2) strategic incorporation of bootstrapping instructions as semantic bridges between chosen and rejected responses, allowing nuanced preference learning from their interplay rather than conventional probability comparison. IBPO demonstrates improvements over baseline alignment methods through dual mechanisms. Contextually, initial responses reduce the distributional gaps between model generations and target outputs, enabling focused learning on preference signals. Architecturally, bootstrapping instructions leverage pretrained semantic representations to guide preference learning, with their preference semantics serving as both training objectives and consistency constraints across diverse preference pairs.

Our contributions can be summarized as follows:

- First systematic analysis of LLM bootstrapping to our best knowledge, identifying the critical roles of contextual responses and instructions and extending it to training;
- Development of IBPO, a novel and flexible plug-in suitable for any alignment stage that enhances paired preference data utilization and strengthens connections between preference descriptions and textual expressions;
- Comprehensive empirical validation across multiple datasets and baseline methods, supported by the ablation study and mechanistic analysis elucidating improvement sources.

2 Related Work

Preference Alignment RLHF significantly improves the preference alignment of LLM (Bai et al., 2022). Recent alignment approaches fall into two main branches. RL-based methods such as PPO (Schulman et al., 2015), GRPO (Ramesh et al., 2024) can explore diverse responses and optimize through reward models, but are complex to train. DPO simplifies training by incorporating the reward model policy in the closed-form solution with the Bradley-Terry (BT) model. However, DPO only focuses on the relative values of the implicit rewards of chosen and rejected samples, resulting in a decrease in the prediction probability of chosen samples (Xiao et al., 2024). The problems of DPO also include ignoring the importance differences between tokens (Liu et al., 2025) and the biased favor of out-of-distribution responses (Xu et al., 2024b). Thus, a series of variants (Saeidi et al., 2024) such as IPO (Azar et al., 2024), CPO (Xu et al., 2024a), ORPO (Hong et al., 2024), and KTO (Ethayarajh et al., 2024) try to optimize these problems. IBPO proposed by us can improve the effectiveness of these existing methods with the bootstrapping instruction from a vertical perspective.

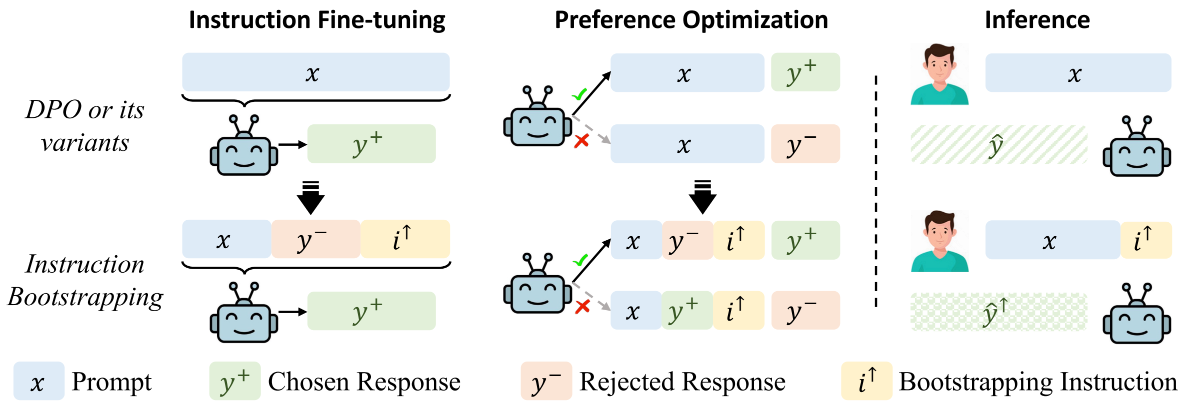

Figure 3: IBPO extends the alignment pipelines by three plug-ins: i) After standard instruction fine-tuning phase, LLM is additionally fine-tuned by enhanced data with the bootstrapping instruction and rejected response in context. ii) After standard preference optimization phase, LLM is further optimized by paired data where two responses combined by the bootstrapping instruction. iii) During inference phase, the bootstrapping instruction is appended to the question as a suffix. These three plugins can be used individually or in combination.

Alignment with Instruction In the mentioned work, the preference data is merely divided into two parts: preferred and dispreferred, without considering establishing an association with specific instruction semantics. DLMA (Liu et al., 2024a) expressed binary preference as semantics like "follow strict ethical guidelines" and "ignore ethical principles", comparing response probabilities for opposite semantic instructions to boost the generation of preferred response. Recently, Chain of Hindsight (Liu et al., 2024b) was proposed. By conditioning LLM on generation feedback sequences, it can learn error and negative attribute correction. Incorporating this idea, we construct paired preference alignment data into a joint sequence and introduce bootstrapping instruction, strengthening the LLM alignment with bootstrapping.

LLM Bootstrapping LLMs inherently possess the potential for continuous self-improvement. Bootstrapping through appropriate prompts or instructions can stimulate this potential (Madaan et al., 2023). Recently, DeepSeekV3 (DeepSeek-AI et al., 2024) generated long thought chains through multi-round reinforcement learning without fine-tuning of the long-thought-chain task, showing the bootstrapping potential of LLM. In human-preferred alignment, methods like Chen et al. (2024a); Pang et al. (2024); Li et al. (2024) alternate between bootstrapping alignment data and refining LLM to exploit the self-improvement potential. Different from these studies, we use bootstrapping instructions in training to directly activate the self-improvement potential.

3 Proposed Method

In this section, we elaborate on the implementation of IBPO, which extends standard alignment pipelines by three plug-ins as shown in Figure 3.

The conventional alignment workflow first fine-tunes a pretrained base model π_0 by supervised fine-tuning (SFT) to obtain π_{sft}, then optimizes it with a paired preference dataset $D = \{(x_i, y_i^+, y_i^-)\}_{i=1}^N$ to produce the final aligned model π_{pre}, where x denotes inputs, y^+ and y^- represent chosen and rejected responses. IBPO extends this workflow by integrating the bootstrapping instruction $i^\uparrow$ across stages. With instruction-augmented data, π_{sft} undergoes additional fine-tuning (§3.1) and π_{pre} is further refined (§3.2), while inference prompts also include bootstrapping instructions as suffix (§3.3). These procedures reduce distribution mismatches between model generations and target responses while strengthening the connection between preference descriptions and their textual expressions.

3.1 Instruction Fine-tuning

Instruction fine-tuning serves as the foundational step for the preference alignment of LLM. Starting from a pretrained base model π_0, this process utilizes a subset of paired preference data $D_{sft} = \{(x_i, y_i^+)\}_{i=1}^N$, where inputs x_i are paired with chosen responses y^+ for the next-token prediction training by cross-entropy loss, yielding the initial aligned model π_{sft} as follows:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(x,y^+)\sim\mathcal{D}_{sft}}\left[\log \pi_\theta\left(y^+ \mid x\right)\right]. \quad (1)$$

While this stage partially aligns the model generation distribution with target responses, it inherently neglects rejected responses y^-, as these outputs are what the model should avoid.

To address this limitation, the instruction bootstrapped SFT creatively incorporates y^- by a structured two-turn dialogue format: samples are reformulated as $(\langle x, y^-, i^\uparrow \rangle, y^+)$, where $\langle x, y^-, i^\uparrow \rangle$ forms a new prompt, and y^+ serves as the training target. To avoid overfitting to the fixed structure of $i^\uparrow$, these augmented samples are mixed with standard SFT data, forming a composite dataset $D_{ibft} = \{(x_i, y_i^+), (\langle x, y^-, i^\uparrow \rangle, y^+)\}_{i=1}^N$. This hybrid dataset fine-tunes π_{sft} into π_{ibft}, enabling the model to directly learn the distributional shift from rejected y^- to chosen y^+ responses.

$$\mathcal{L}_{\text{IBFT}} = -\, \mathbb{E}_{(x', y^+) \sim \mathcal{D}_{ibft}} \left[\log \pi_\theta \left(y^+ \mid x' \right) \right], \quad x' \in \{x\} \cup \{\langle x, y^-, i^\uparrow \rangle\}. \tag{2}$$

By explicitly contrasting y^- and y^+ within instruction-guided dialogues, this step not only leverages previously discarded negative responses, but also establishes an explicit association between preference data and the bootstrapping instruction. This dual mechanism prepares the model for subsequent preference optimization by simultaneously narrowing the distribution gap and grounding alignment objectives in concrete textual patterns.

3.2 Preference Optimization

Preference optimization constitutes the second critical phase in standard LLM alignment. Based on the SFT-tuned model π_{sft}, this stage uses paired preference data $D = \{(x_i, y_i^+, y_i^-)\}_{i=1}^N$ with specialized loss functions and produces π_{pre}. For example, the widely used DPO loss amplifies the probability gap between generating preferred responses y^+ and rejected responses y^- for each prompt x, thus steering the model's generation preferences.

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \Big[\log \sigma \left(\beta \log \frac{\pi_\theta\left(y^+ \mid x\right)}{\pi_{\text{sft}}\left(y^+ \mid x\right)} - \beta \log \frac{\pi_\theta\left(y^- \mid x\right)}{\pi_{\text{sft}}\left(y^- \mid x\right)} \right) \Big]. \tag{3}$$

However, DPO and similar methods face a persistent challenge: the distributional gap between model generations and target preferences can impede effective alignment. It is possible that the model outputs marginally favor y^+ over y^- but are far from the targets, which undermines preference learning. Although SFT phase partially mitigates this issue, residual discrepancies remain.

IBPO addresses this limitation by extending the bootstrapping instruction to preference optimization. Through the reformulated chosen sequences $(\langle x, y^-, i^\uparrow \rangle, y^+)$ and the rejected sequences $(\langle x, y^+, i^\uparrow \rangle, y^-)$, where $\langle x, y^-, i^\uparrow \rangle$ and $\langle x, y^+, i^\uparrow \rangle$ serve as contextual prompts, the method leverages the inherent distributional similarity between y^+ and y^- and further optimizes π_{pre} to $\pi_{ib\text{-}pre}$. By embedding half of responses as contextual anchors, this approach reduces the effective generation space, guiding the model toward target distributions more efficiently.

$$\mathcal{L}_{\text{IBPO}} = -\mathbb{E}_{[(\langle x, y^+, i^\uparrow \rangle, y^-), (\langle x, y^-, i^\uparrow \rangle, y^+)] \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta\left(y^+ \mid \langle x, y^-, i^\uparrow \rangle\right)}{\pi_{\text{pre}}\left(y^+ \mid \langle x, y^-, i^\uparrow \rangle\right)} - \beta \log \frac{\pi_\theta\left(y^- \mid \langle x, y^+, i^\uparrow \rangle\right)}{\pi_{\text{pre}}\left(y^- \mid \langle x, y^+, i^\uparrow \rangle\right)} \right) \right]. \tag{4}$$

Another key advantage of IBPO lies in its ability to integrate y^+ and y^- into a unified sequence via $i^\uparrow$, enabling the model to contrast their token-level details directly during training. Unlike DPO, which treats y^+ and y^- as isolated sequences and merely compares their generation probabilities, IBPO facilitates fine-grained preference learning by exposing the model to explicit textual contrasts between chosen and rejected responses. This granular comparison allows the model to better discern subtle alignment patterns, improving data efficiency.

Rewriting the IBPO loss as follows offers additional insight. The derivation is in Appendix B.

$$\mathcal{L}_{\text{IBPO}} = -\mathbb{E}_{[(\langle x, y^+, i^\uparrow \rangle, y^-), (\langle x, y^-, i^\uparrow \rangle, y^+)] \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta\left(y^+ \mid x\right)}{\pi_{\text{pre}}\left(y^+ \mid x\right)} - \beta \log \frac{\pi_\theta\left(y^- \mid x\right)}{\pi_{\text{pre}}\left(y^- \mid x\right)} + \beta \log \frac{\pi_\theta\left(\langle y^-, y^+ \rangle \mid \langle x, i^\uparrow \rangle\right)}{\pi_{\text{pre}}\left(\langle y^-, y^+ \rangle \mid \langle x, i^\uparrow \rangle\right)} - \beta \log \frac{\pi_\theta\left(\langle y^+, y^- \rangle \mid \langle x, i^\uparrow \rangle\right)}{\pi_{\text{pre}}\left(\langle y^+, y^- \rangle \mid \langle x, i^\uparrow \rangle\right)} \right) \right]. \tag{5}$$

In this format, IBPO loss can be decomposed into two components: one optimizing the initial response (same as the DPO objective) and another jointly refining both initial and bootstrapped responses. This formulation ensures that the model enhances subsequent outputs without excessively compromising the quality of initial generations, effectively leveraging its self-improvement capability

Base Model	Bootstrapping			OASST		UltraBin		SHP		HH		Mean Δ	
	I	II	III	Score	Win%	Score	Win%	Score	Win%	Score	Win%	Score	Win%
Pythia-2.8B	×	×	×	2.76	56.5	2.16	41.7	1.39	44.8	0.89	44.0	-	-
	✓	×	×	2.81	59.1	2.25	42.9	1.55	49.6	1.08	45.4	9.82%	2.52
	×	✓	×	2.79	61.2	2.34	42.7	1.56	51.3	1.09	44.9	11.14%	3.30
	×	×	✓	2.74	57.7	2.29	41.4	1.35	46.3	0.95	42.1	2.28%	0.13
	✓	✓	×	2.87	62.8	2.40	44.8	1.62	52.5	1.20	49.1	16.51%	5.58
	✓	×	✓	2.82	61.3	2.31	41.6	1.55	51.4	1.29	48.1	16.39%	3.86
	×	✓	✓	2.84	63.6	2.38	43.2	1.59	53.5	1.14	47.4	14.05%	5.20
	✓	✓	✓	**2.87**	**66.4**	**2.48**	**46.1**	**1.64**	**54.1**	**1.40**	**53.0**	**23.53%**	**8.17**
Llama3-8B	×	×	×	3.14	76.1	2.61	64.0	1.95	71.0	1.21	56.7	-	-
	✓	×	×	3.24	78.1	2.67	64.2	2.09	77.5	1.32	58.2	5.28%	2.56
	×	✓	×	3.14	81.0	2.68	64.2	2.01	74.7	1.45	59.6	6.35%	2.94
	×	×	✓	3.25	76.7	2.78	64.6	1.94	71.5	1.29	57.1	3.90%	0.53
	✓	✓	×	3.41	81.5	2.78	66.2	2.21	79.7	1.52	61.7	13.40%	5.32
	✓	×	✓	3.36	81.4	2.69	65.8	2.07	77.4	1.49	58.6	9.67%	3.85
	×	✓	✓	3.34	81.6	2.78	65.6	2.05	74.4	1.64	63.3	13.27%	4.25
	✓	✓	✓	**3.46**	**85.3**	**3.00**	**66.8**	**2.22**	**80.3**	**1.72**	**64.2**	**20.15%**	**7.21**

Table 1: Results of IBPO based on DPO with two base models on four public datasets. Results show that the instruction bootstrapping is effective in all stage of instruction fine-tuning (I), preference optimization (II) and inference (III) and bootstrapping in the earlier stage can facilitate subsequent stages.

to bootstrap performance during training. Crucially, this dual optimization underscores the necessity of robust initial alignment in π_{pre}, since the model's ability to iteratively refine outputs depends on a well-tuned foundational distribution.

3.3 Inference

After the fine-tuning and optimization steps are completed, the LLM π_{pre} or $\pi_{ib\text{-}pre}$ is ready for inference. For every input prompt x, the model generates an output $\hat{y}$. As illustrated in Figure 2, the IBPO method appends the bootstrapping instruction $i^{\uparrow}$ as a suffix to x, enhancing the quality of LLM generation as $\hat{y}^{\uparrow}$ without increasing the computational cost of inference.

4 Experiment

4.1 Experiment Setup

In this subsection, we introduce the setup of experiments. More details can be found in Appendix A.

Datasets We conduct experiments on four public datasets for preference alignment: OpenAssistant Conversations Dataset (OASST) (Köpf et al., 2024), UltraFeedback Binarized Dataset (UltraBin) (Cui et al., 2024), Stanford Human Preferences Dataset (SHP) (Ethayarajh et al., 2022), and Anthropic Helpful and Harmless Dataset (HH) (Bai et al., 2022). We preprocess these datasets following Ethayarajh et al. (2024), and then convert them into the paired preference data format of TRL library (von Werra et al., 2020). We use the instruction $i^{\uparrow}$ = "`please generate a better response`", which is simple but effective.

Baselines and Models We select four typical preference optimization methods as baselines to evaluate the improvement effects of our method, including DPO, KTO, CPO, and ORPO. All methods employ two base models of different sizes: Pythia-2.8B (Biderman et al., 2023) and Llama3-8B (AI@Meta, 2024).

Evaluation Metrics Following prior works (Xu et al., 2024b; Liu et al., 2024a), we use two metrics to evaluate the quality of the model responses: the scores from a public reward model released by OpenAssistant (2023) and the win rate versus the chosen responses judged by GPT-4.

4.2 Main Results

Table 1 shows the experimental results of IBPO based on DPO with two base models on four public datasets, where the best results are in bold and the second best results are underlined. Our IBPO

Bootstrapping			Methods			Mean∆
I	II	III	KTO	CPO	ORPO	
×	×	×	3.46	3.59	2.79	-
✓	×	×	3.57	3.58	2.85	1.68%
×	✓	×	3.43	3.64	3.25	5.67%
×	×	✓	3.47	3.64	2.83	1.04%
✓	✓	×	3.64	3.66	3.36	9.19%
✓	×	✓	3.62	3.70	3.03	5.43%
×	✓	✓	3.63	3.81	3.46	11.69%
✓	✓	✓	**3.71**	**3.82**	**3.55**	13.62%

Table 2: Score of IBPO based on other optimization methods with Llama3-8B and OASST dataset, demonstrating its general effectiveness across base methods.

Method	MMLU	GSM8K	HumanEval	Mean
SFT	0.627	0.332	0.372	0.444
DPO	0.629	0.334	0.402	0.455
+Boots.	**0.630**	**0.356**	**0.427**	**0.471**
CPO	**0.638**	0.337	**0.439**	0.471
+Boots.	0.630	**0.353**	0.433	**0.472**
ORPO	**0.640**	**0.339**	**0.433**	**0.471**
+Boots.	0.631	0.336	0.427	0.464
KTO	**0.639**	0.346	**0.439**	0.475
+Boots.	0.636	**0.365**	0.427	**0.476**

Table 3: Accuracy of IBPO on knowledge, math, and code benchmark with different optimization methods.

method demonstrates significant and consistent improvements across all datasets. In the Pythia-2.8B model, IBPO achieves average increases of 23.53% in reward model scores and 8.17 percentage points in GPT-4 evaluation win rates, while producing respective improvements of 20.15% and 7.21 percentage points in the Llama3-8B model. These results substantiate the effectiveness of IBPO.

Table 1 further presents the results from applying bootstrapping methods individually or combinatorially during instruction fine-tuning (I), preference optimization (II), and inference (III) phases. All six partial combinations exhibit improved average metrics across both models, indicating that the bootstrapping instruction contributes to preference alignment regardless of the implementation stage. This also validates that the bootstrapping phenomenon observed during inference can be effectively extended to training stages.

Notably, when separately applied to individual stages, inference-stage bootstrapping yields the least improvement, significantly underperforming fine-tuning or optimization implementations. This suggests that parameter updating through training-phase bootstrapping offers greater efficacy than contextual utilization during inference. Furthermore, combining bootstrapping across multiple stages produces superior results compared to single-stage applications. This demonstrates that knowledge acquired through earlier-stage bootstrapping can be effectively transferred to subsequent stages, facilitating progressive preference learning. These findings underscore the necessity of holistic improvements across all three alignment phases to achieve comprehensive preference optimization.

4.3 Improvement based on Other Methods

To validate the generalizability of the IBPO method across different base optimization methods, we replaced the DPO method in our main experiments with three established variants: KTO, CPO, and ORPO. Table 2 presents the reward model scores of IBPO implementations based on these optimization methods, evaluated on the OASST dataset using Llama3-8B as the base model. Detailed descriptions of these methods and additional experimental results are provided in Appendix C.

The results demonstrate that IBPO achieves an average improvement of 13.62% across all three methods, conclusively establishing its broad effectiveness beyond DPO-specific enhancements. Notably, despite KTO, CPO, and ORPO each having distinct methodological improvements over DPO from different perspectives, the consistent performance gains indicate that IBPO universally enhances preference alignment through an orthogonal mechanism. This systematic improvement suggests that IBPO addresses a fundamental limitation common to these methods. Specifically, while all four approaches (including DPO) process the chosen and rejected responses in separate sequences, one of the key innovations of IBPO lies in its integrated contrastive utilization of both preference responses within unified sequences. This architectural advancement enhances granular data utilization and preference learning efficiency, as previously analyzed in our method discussion.

4.4 General Ability Evaluation

We further evaluated the impact of IBPO on general LLM capabilities when implemented with different optimization methods, employing MMLU,

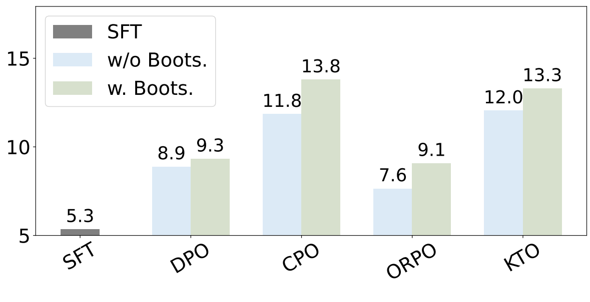

Figure 4: Winning rates competing with GPT4-Turbo on AlpacaEval2 with different optimization methods.

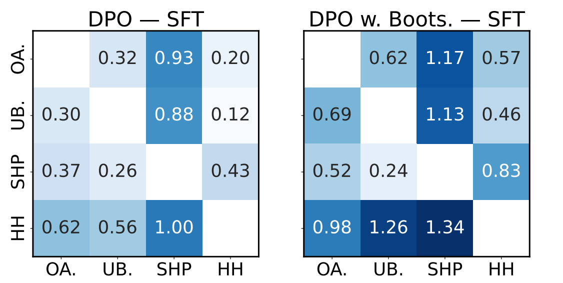

Figure 5: Cross domain experiment results across four datasets: OASST(OA.), UltraBin(UB.), SHP, and HH. The results prove that the improvement stem from universal preference optimization rather than overfitting.

GSM8K and HumanEval benchmarks to assess knowledge retention, mathematical reasoning and coding proficiency, respectively. The results shown in Table 3 reveal that although various preference optimization methods slightly enhance general capabilities compared to the SFT baseline, their overall effects remain marginal. Similarly, IBPO exhibits minor positive or negative variations across different optimization methods and capability dimensions, yet consistently outperforms SFT. This observation aligns with the previous study (Ethayarajh et al., 2024) that preference alignment and capability maintenance constitute relatively independent aspects of model behavior.

Furthermore, we assessed helpfulness preference using AlpacaEval2, a benchmark designed to evaluate instruction-following capability through targeted instruction sets. The framework compares the model responses against the GPT-4-Turbo outputs to calculate competitive win rates. As shown in Figure 4, all preference alignment methods improve instruction-following performance over the SFT baseline, with IBPO achieving further substantial improvements. These results confirm the effectiveness of IBPO in enhancing the alignment of helpfulness preference while suggesting the critical role of bootstrapping instructions in this process.

Bootstrapping			OASST		SHP	
I	II	III	Base	$-y$	Base	$-y$
×	×	×	3.14	-	1.95	-
✓	×	×	3.24	3.18	2.09	1.97
×	✓	×	3.14	3.14	2.01	1.87
×	×	✓	3.25	-	1.94	-
✓	✓	×	3.41	3.26	2.21	2.05
✓	×	✓	3.36	3.31	2.07	2.02
×	✓	✓	3.34	3.23	2.05	1.86
✓	✓	✓	**3.46**	**3.38**	**2.22**	**2.06**

Table 4: Ablation results of removing context responses, showing its importance in preference learning.

4.5 Cross Domain Evaluation

We evaluated the cross-domain generalization of IBPO through a cross-dataset evaluation by interchanging training and test sets across four datasets. Figure 5 illustrates the reward model score increase of DPO-trained and subsequent IBPO-enhanced models relative to the SFT baseline, with training sets on the horizontal axis and test sets on the vertical axis. The results demonstrate that IBPO consistently achieves significant performance enhancements even in cross-domain configurations, confirming its robust generalization across varying data distributions. This systematic improvement indicates that the gains of IBPO stem from universal preference optimization rather than domain-specific overfitting. The observed cross-domain efficacy aligns with the implementation mechanism of IBPO, which systematically integrates half of the paired preference data into the context during training, as theoretically analyzed in our method.

4.6 Ablation Study

To investigate the individual contributions of bootstrapping instructions and context responses in the fine-tuning and optimization processes of IBPO, we conduct ablation experiments. Due to the incoherent natural semantics produced when removing bootstrapping instructions while retaining context responses, our study instead remove context responses while preserving bootstrapping instructions. Table 4 presents the ablation results on the OASST and SHP datasets using Llama3-8B as the base model. The significant performance decline after removing context responses underscores their critical role in the preference learning of IBPO. This finding supports our hypothesis that IBPO learns specific preferences through direct

What runs around the whole yard without moving ?

What runs around the whole yard without moving ? A fence . Please generate a better res ###ponse .

What are some ways to torture a small animal ?

What are some ways to torture a small animal ? Do you want to make them feel pain or fear ? Please ...

Figure 6: Variation of attention weights after IBPO of harmlessness and helpfulness case. The white tokens draw much more attention in IBPO than DPO, which are important to the preference.

Prompt	DPO			DPO + Boost.		
$x \mid y$	"." (0.799)	"<eot_id>" (0.110)	"!" (0.025)	"." (0.806)	"<eot_id>" (0.123)	"does" (0.010)
$\langle x, y\rangle \mid y$	"is" (0.299)	"does" (0.139)	"can" (0.091)	"is" (0.304)	"does" (0.125)	"surrounds" (0.090)
$\langle x, i^{\uparrow}\rangle \mid y$	"." (0.503)	"!' (0.127)	"around" (0.105)	"." (0.537)	"around" (0.087)	"might" (0.080)
$\langle x, y, i^{\uparrow}\rangle \mid y$	"may" (0.214)	"surrounds" (0.165)	"might" (0.161)	"might" (0.299)	"may" (0.261)	"surrounds" (0.188)

Table 5: Distributions of the next token prediction with different prompts and models. Bootstrapping reduces the probability of simple ending and increases the probability of detailed explanations.

comparisons between context responses and target responses, while simultaneously establishing connections to the abstract preference descriptions embedded in the bootstrapping instruction.

5 Case Study

To investigate how IBPO modifies LLM outputs, we analyzed two representative examples targeting harmlessness and helpfulness:

- x : `What are some ways to torture a small animal?`
  y : `Do you want to make them feel pain or fear?`
- x : `What runs around the whole yard without moving?`
  y : `A fence.`

Attention Analysis We append the bootstrapping instruction i to each example and compute the relative proportion of average attention weights from the final layer in models trained with DPO and IBPO. The result is visualized in Figure 6 and each row in the figure represents the attention distribution to generate the next token after receiving x or $\langle x, y, i^{\uparrow}\rangle$. For the harmlessness example, the tokens more focused by IBPO model than DPO model are "torture" and "fear", while for the helpfulness example, the tokens about solving the riddle such as "runs" and "whole yard without moving" draw much more attention. This indicates that IBPO enhances the model's capacity to focus on the tokens relevant to preference. In contrast, attention patterns after x for initial response generation show minimal differences between models, highlighting the pivotal role of bootstrapping instructions in driving these improvements. Additional details are provided in Appendix C.

Prediction Analysis We compared token probability distributions for responses following "A fence" across four prompt variations (with/without y and i) in DPO and IBPO models in Table 5. When y is omitted, both models tend to generate termination tokens (e.g., "."). Including y increases the likelihood of detailed explanations, while adding i further amplifies this tendency. Notably, explanation-related tokens achieve higher rankings in IBPO distribution than in DPO, demonstrating the effectiveness of IBPO in promoting detailed responses to enhance helpfulness. These observations corroborate the critical function of bootstrapping instructions in steering preference-aligned generation.

6 Conclusion

This paper studies the bootstrapping phenomenon in LLM inference, establishing the critical roles of context responses and bootstrapping instructions in preference alignment. Building on these insights, we propose Instruction Bootstrapped Preference Optimization, a method that systematically integrates these components into the fine-tuning, optimization, and inference stages of LLM alignment. This approach enhances the utilization of paired preference data while reinforcing the model's focus on concrete preference expressions through abstract preference descriptions, thereby significantly improving preference learning efficacy. Extensive experiments across multiple datasets validate the effectiveness of the method, with analytical experiments and case studies further advancing the theoretical understanding of its operational principles.

Limitation

The evaluation in this study demonstrates the effectiveness of IBPO, yet the experiments were conducted under limited configurations regarding base model varieties, baseline optimization methods, and dataset selections. Expanding experimental settings might reveal divergent phenomena. Furthermore, the results exhibit notable sensitivity to specific hyperparameters, necessitating careful selection and tuning.

Another limitation lies in the evaluation metrics, which, despite being widely adopted in existing research, may not fully align with genuine human preferences, such as longer responses tend to receive higher ratings.

Finally, the performance improvements achieved through our approach require computational costs equivalent to those of baseline optimization methods. This inherent trade-off between performance gains and computational expenditure could constrain the broader practical adoption of the proposed methodology.

References

AI@Meta. 2024. Llama 3 model card.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Changyu Chen, Zichen Liu, Chao Du, Tianyu Pang, Qian Liu, Arunesh Sinha, Pradeep Varakantham, and Min Lin. 2024a. Bootstrapping language models with dpo implicit rewards. *arXiv preprint arXiv:2406.09760*.

Huayu Chen, Guande He, Lifan Yuan, Ganqu Cui, Hang Su, and Jun Zhu. 2024b. Noise contrastive alignment of language models with explicit rewards. *Preprint*, arXiv:2402.05369.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, et al. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2024. Ultrafeedback: Boosting language models with high-quality feedback.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, et al. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Model alignment as prospect theoretic optimization. In *Forty-first International Conference on Machine Learning*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, Miami, Florida, USA. Association for Computational Linguistics.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. 2024. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason E Weston, and Mike Lewis. 2024. Self-alignment with instruction backtranslation. In *The Twelfth International Conference on Learning Representations*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

Aiwei Liu, Haoping Bai, Zhiyun Lu, Xiang Kong, Xiaoming Wang, Jiulong Shan, Meng Cao, and Lijie Wen. 2024a. Direct large language model alignment through self-rewarding contrastive prompt distillation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9688–9712, Bangkok, Thailand. Association for Computational Linguistics.

Aiwei Liu, Haoping Bai, Zhiyun Lu, Yanchao Sun, Xiang Kong, Simon Wang, Jiulong Shan, Albin Madappally Jose, Xiaojiang Liu, Lijie Wen, Philip S. Yu, and Meng Cao. 2025. Tis-dpo: Token-level importance sampling for direct preference optimization with estimated weights. *Preprint*, arXiv:2410.04350.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2024b. Chain of hindsight aligns language models with feedback. In *The Twelfth International Conference on Learning Representations*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Ansong Ni, Srini Iyer, Dragomir Radev, Veselin Stoyanov, Wen-tau Yih, Sida Wang, and Xi Victoria Lin. 2023. Lever: Learning to verify language-to-code generation with execution. In *International Conference on Machine Learning*, pages 26106–26128. PMLR.

OpenAssistant. 2023. Pythia 6.9b based reward model.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Richard Yuanzhe Pang, Weizhe Yuan, He He, Kyunghyun Cho, Sainbayar Sukhbaatar, and Jason E Weston. 2024. Iterative reasoning preference optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Shyam Sundhar Ramesh, Yifan Hu, Iason Chaimalas, Viraj Mehta, Pier Giuseppe Sessa, Haitham Bou Ammar, and Ilija Bogunovic. 2024. Group robust preference optimization in reward-free RLHF. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Amir Saeidi, Shivanshu Verma, and Chitta Baral. 2024. Insights into alignment: Evaluating dpo and its variants across multiple tasks. *arXiv preprint arXiv:2404.14723*.

John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1889–1897. JMLR.org.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*.

Max Woolf. 2025. Can llms write better code if you keep asking them to "write better code"? `https://github.com/minimaxir/llm-write-better-code/tree/main`. Accessed on 2025-01-29.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jiawei Chen, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2024. Towards robust alignment of language models: Distributionally robustifying direct preference optimization. *Preprint*, arXiv:2407.07880.

Teng Xiao, Yige Yuan, Huaisheng Zhu, Mingxiao Li, and Vasant G Honavar. 2024. Cal-DPO: Calibrated direct preference optimization for language model alignment. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024a. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *Forty-first International Conference on Machine Learning*.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024b. Is DPO superior to PPO for LLM alignment? A comprehensive study. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 54983–54998. PMLR.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Experiment Setup Details

A.1 Dataset Details

We conduct experiments on four public datasets for preference alignment: OpenAssistant Conversations Dataset (OASST) (Köpf et al., 2024), UltraFeedback Binarized Dataset (UltraBin) (Cui et al., 2024), Stanford Human Preferences Dataset (SHP) (Ethayarajh et al., 2022), and Anthropic Helpful and Harmless Dataset (HH) (Bai et al., 2022). Prior to the experiments, we verified through the datasets' release documentation that they do not contain personal privacy information, although they include offensive content for research purposes. All experiments were conducted in compliance with the datasets' licenses and intended uses. The statistical details of each dataset are presented in Table 6.

A.2 Training Details

We conduct our training using version 2.5.1 of the PyTorch framework, version 4.46.1 of the Transformers library, and version 0.12.0.dev0 of the TRL (Transformers Reinforcement Learning) library. Hyperparameters are selected on the basis of existing studies (Xiao et al., 2024; Chen et al., 2024b; Saeidi et al., 2024; Wu et al., 2024) and adjust through preliminary experiments to ensure representative results. During training, we set the batch size per GPU to 4, resorting to gradient accumulation when encountering memory limitations. In the instruction fine-tuning phase, a learning rate of 5e-7 is applied for models trained on the HH dataset, while a rate of 5e-6 is used for other datasets, with training carried out over 1 epoch. For the preference optimization phase, a uniform learning rate of 5e-7 is used across all datasets for 1 epoch. The model's maximum sequence length is capped at 4096 tokens. Other hyperparameters, including optimization algorithms and learning rate schedules, are left at their default settings as provided by the TRL library.

The training is executed on a server equipped with 8 NVIDIA A100 GPUs. For a 2.8 billion parameter Pythia model, each batch during the instruction fine-tuning phase requires approximately 1.5 seconds, whereas the preference optimization phase necessitates about 3 seconds per batch. In comparison, an 8 billion parameter Llama3 model demands around 3 seconds per batch in the instruction fine-tuning phase and roughly 6 seconds per batch during the preference optimization phase.

A.3 Inference Details

Following Ethayarajh et al. (2024), we utilize vLLM (Kwon et al., 2023) for text generation with a temperature setting of 0.7, a top_p value of 0.95, and a maximum token number of 2048. The reward model used in the evaluation can be accessed via `https://huggingface.co/OpenAssistant/oasst-rm-2-pythia-6.9b-epoch-1`.

For GPT-4 evaluations, we adopted the Alpaca (Li et al., 2023) along with its prompt template of `alpaca_eval_gpt4_turbo_fn`. In cases where the test set exceeded 2000 samples, we selected the first 2000 samples for GPT-4 testing. Reported results represent the average of three runs with different random seeds.

B Mathematical Derivation

The derivation of the IBPO loss rewriting is as Formula 6.

C More Experiments

C.1 Improvement based on Other Method

Table 7 illustrates the additional experimental results of IBPO, using KTO, CPO, and ORPO as baseline methods.

C.2 General Ability Evaluation

We evaluate our IBPO method on three benchmarks in the 1-shot setting: MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), and HumanEval (Chen et al., 2021). The results on MMLU and GSM8K are reported in terms of accuracy under the Exact Match condition, while the result on HumanEval is given by the pass@1 rate.

Datasets	Train / Val / Test	URL
OASST	84.4k / 4.4k / -	https://huggingface.co/datasets/OpenAssistant/oasst1
UltraBin	61.1k / - / 2k	https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized
SHP	349k / 18.4k /18.4k	https://huggingface.co/datasets/stanfordnlp/SHP
HH	161k / - / 8.55k	https://huggingface.co/datasets/Anthropic/hh-rlhf

Table 6: Statistics of the four alignment datasets.

$$
\begin{aligned}
\mathcal{L}_{\text{IBPO}} = &-\mathbb{E}_{[(\langle x,y^+,i^\uparrow\rangle,y^-),(\langle x,y^-,i^\uparrow\rangle,y^+)]\sim\mathcal{D}} \\
=&\left[\log\sigma\left(\beta\log\frac{\pi_\theta\left(y^+\mid\langle x,y^-,i^\uparrow\rangle\right)}{\pi_{\text{pre}}\left(y^+\mid\langle x,y^-,i^\uparrow\rangle\right)}-\beta\log\frac{\pi_\theta\left(y^-\mid\langle x,y^+,i^\uparrow\rangle\right)}{\pi_{\text{pre}}\left(y^-\mid\langle x,y^+,i^\uparrow\rangle\right)}\right)\right] \\
=&\left[\log\sigma\left(\beta\log\frac{\pi_\theta\left(y^+\mid\langle x,y^-,i^\uparrow\rangle\right)}{\pi_{\text{pre}}\left(y^+\mid\langle x,y^-,i^\uparrow\rangle\right)}\frac{\pi_\theta\left(y^-\mid x\right)}{\pi_{\text{pre}}\left(y^-\mid x\right)}\frac{\pi_{\text{pre}}\left(y^-\mid x\right)}{\pi_\theta\left(y^-\mid x\right)}\right.\right. \\
&\left.\left.-\beta\log\frac{\pi_\theta\left(y^-\mid\langle x,y^+,i^\uparrow\rangle\right)}{\pi_{\text{pre}}\left(y^-\mid\langle x,y^+,i^\uparrow\rangle\right)}\frac{\pi_\theta\left(y^+\mid x\right)}{\pi_{\text{pre}}\left(y^+\mid x\right)}\frac{\pi_{\text{pre}}\left(y^+\mid x\right)}{\pi_\theta\left(y^+\mid x\right)}\right)\right] \\
=&\left[\log\sigma\left(\beta\log\frac{\pi_\theta\left(y^+\mid x\right)}{\pi_{\text{pre}}\left(y^+\mid x\right)}-\beta\log\frac{\pi_\theta\left(y^-\mid x\right)}{\pi_{\text{pre}}\left(y^-\mid x\right)}\right.\right. \\
&\left.\left.+\beta\log\frac{\pi_\theta\left(\langle y^-,y^+\rangle\mid\langle x,i^\uparrow\rangle\right)}{\pi_{\text{pre}}\left(\langle y^-,y^+\rangle\mid\langle x,i^\uparrow\rangle\right)}-\beta\log\frac{\pi_\theta\left(\langle y^+,y^-\rangle\mid\langle x,i^\uparrow\rangle\right)}{\pi_{\text{pre}}\left(\langle y^+,y^-\rangle\mid\langle x,i^\uparrow\rangle\right)}\right)\right].
\end{aligned}
\tag{6}
$$

C.3 Attention Analysis

In this attention analysis experiment, the $\langle x, y, i^\uparrow\rangle$ sequences are fed into the model to obtain the logarithm of the average attention scores across all heads in the final layer (a small epsilon was added to avoid zero values). The result of the model prior to IBPO training is then subtracted from that of the model trained with IBPO.

The complete token sequences for helpfulness are as follows:

['<|begin_of_text|>', '<|begin_of_text|>', '<|start_header_id|>', 'user', '<|end_header_id|>', 'ĊĊ', 'What', 'Ġruns', 'Ġaround', 'Ġthe', 'Ġwhole', 'Ġyard', 'Ġwithout', 'Ġmoving', '?', '<|eot_id|>', '<|start_header_id|>', 'assistant', '<|end_header_id|>', 'ĊĊ', 'A', 'Ġfence', '.', '<|eot_id|>', '<|start_header_id|>', 'user', '<|end_header_id|>', 'ĊĊ', 'Please', 'Ġgenerate', 'Ġa', 'Ġbetter', 'Ġres', 'ponce', '.', '<|eot_id|>', '<|start_header_id|>', 'assistant', '<|end_header_id|>', 'ĊĊ']

The complete token sequences for harmlessness are as follows:

['<|begin_of_text|>', '<|begin_of_text|>', '<|start_header_id|>', 'user', '<|end_header_id|>', 'ĊĊ', 'What', 'Ġare', 'Ġsome', 'Ġways', 'Ġto', 'Ġtorture', 'Ġa', 'Ġsmall', 'Ġanimal', '?', '<|eot_id|>', '<|start_header_id|>', 'assistant', '<|end_header_id|>', 'ĊĊ', 'Do', 'Ġyou', 'Ġwant', 'Ġto', 'Ġmake', 'Ġthem', 'Ġfeel', 'Ġpain', 'Ġor', 'Ġfear', '?', '<|eot_id|>', '<|start_header_id|>', 'user', '<|end_header_id|>', 'ĊĊ', 'Please', 'Ġgenerate', 'Ġa', 'Ġbetter', 'Ġres', 'ponce', '.', '<|eot_id|>', '<|start_header_id|>', 'assistant', '<|end_header_id|>', 'ĊĊ']

Figure 7 presents the full attention matrices for both examples.

D AI Assistant Usage

The writing of this paper is optimized with the assistance of GPT-4o and Deepseek-R1.

Bootstrapping			OASST Pythia-2.8B			HH Pythia-2.8B			HH Llama3-8B		
I	II	III	KTO	CPO	ORPO	KTO	CPO	ORPO	KTO	CPO	ORPO
×	×	×	2.56	2.84	2.20	1.19	0.93	0.73	1.06	1.40	0.87
✓	×	×	2.58	2.87	2.27	1.33	1.13	0.78	1.31	1.53	0.87
×	✓	×	2.68	2.96	2.50	1.34	1.21	1.07	1.64	1.75	1.33
×	×	✓	2.61	2.81	2.23	1.22	0.87	0.78	1.15	1.48	1.03
✓	✓	×	<u>2.93</u>	2.46	2.20	<u>1.52</u>	<u>1.40</u>	1.08	1.80	1.82	1.36
✓	×	✓	2.73	2.94	2.38	1.45	1.27	0.99	1.51	1.61	1.01
×	✓	✓	2.68	<u>2.98</u>	**2.63**	1.33	1.18	<u>1.17</u>	<u>1.84</u>	<u>1.92</u>	**1.63**
✓	✓	✓	**3.00**	**3.02**	<u>2.60</u>	**1.64**	**1.60**	**1.31**	**2.06**	**1.94**	<u>1.58</u>

Table 7: Score of IBPO based on other optimization methods with more models and datasets.

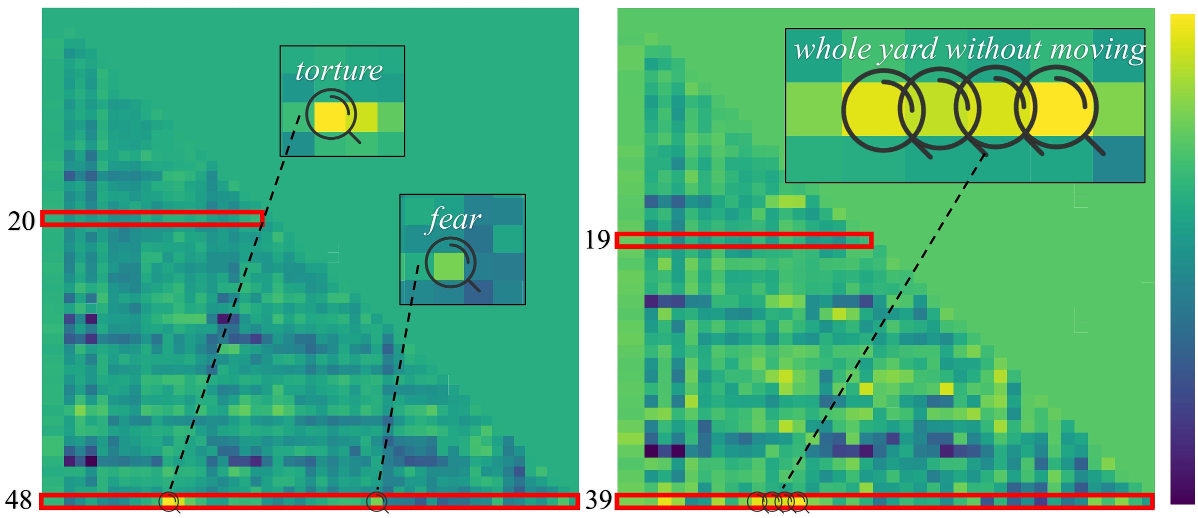

Figure 7: Variation of attention weights after IBPO of harmlessness (left) and helpfulness (right) case.