

Understanding Conflicts in Multi-Objective Alignment through REWARD CONSISTENCY

Zhihao Xu^{1,2*}, Yongqi Tong³, Xin Zhang³, Jun Zhou⁴, Xiting Wang^{1,2†},

¹Gaoling School of Artificial Intelligence, Renmin University of China Beijing, China

²Beijing Key Laboratory of Research on Large Models and Intelligent Governance

³Ant International, ⁴Ant Group,

zhihaoxu@ruc.edu.cn

Abstract

Multi-objective preference alignment often faces alignment conflicts, where optimizing for one objective degrades performance on others. While prior work focuses on algorithmic solutions, the intrinsic conflict within data and its theoretical impact on training remain underexplored. To bridge this gap, we introduce the principle of REWARD CONSISTENCY (RC), a theory-grounded criterion that approximates the alignment conflicts via reward models. We prove that a sample mitigates conflicts if and only if it satisfies RC, thereby ensuring improvement across all objectives during optimization. Building on this, we propose REWARD CONSISTENCY SAMPLING (RCS), an automated framework for constructing pairwise data that adheres to RC, supplemented by a relaxation strategy to enhance flexibility. Extensive experiments show that RCS brings significant and consistent performance gains, achieving an average improvement of 23.07% in both harmlessness and helpfulness during simultaneous optimization compared to the vanilla dataset. Our data-centric approach is complementary to existing alignment algorithms and effective in both sequential and simultaneous optimization scenarios.

1 Introduction

Alignment is a critical stage in the post-training of language models, designed to ensure that the generated responses align with human preferences and values (Guo et al., 2025; Lambert et al., 2024; Xu et al., 2024; Ding et al., 2025). However, existing training pipelines still face the significant challenge of aligning with diverse and often conflicting human preference objectives such as helpfulness and harmlessness (Casper et al., 2023; Rame et al., 2024). The inherent conflicts between different human preferences often lead to trade-offs (Bai et al.,

2022; Lou et al., 2024), where optimizing for one preference may degrade performance in another preference, hindering universal performance improvements across diverse alignment dimensions, which we refer to as *alignment conflicts*.

Recent advancements in multi-objective direct preference alignment (Lou et al., 2024; Zhou et al., 2024b; Yang et al., 2024; Gupta et al., 2025; Shi et al., 2024) have introduced algorithmic improvements to reduce alignment conflicts while avoiding the high computational costs (Rame et al., 2024). For example, by introducing additional losses for balancing multiple objectives during optimization, SPO (Lou et al., 2024) and MODPO (Zhou et al., 2024b) extend DPO through a margin loss term in their objective functions. However, their effectiveness remains limited by the data they use: existing datasets have not been carefully calibrated to mitigate conflicts in different competing objectives.

Therefore, a key research question is, *how to understand the impact of data on alignment conflicts and leverage this understanding to ensure consistent improvement in multiple objectives?* This question motivates us to study the criterion that distinguishes samples that introduce alignment conflicts from ones that do not. While there are existing data selection approaches that identify samples that enhance performance gain in general instruction following or a specific task (e.g., reasoning task) (Khaki et al., 2024; Pattnaik et al., 2024; Lai et al., 2024; Cui et al., 2023; Wang et al., 2024), there remains a lack of understanding of multi-objective alignment conflicts. Although there exist common empirical strategies like data mixing and reward ensembling to mitigate alignment conflicts, there remains a lack of insights of what kind of data would cause conflicts theoretically, leading to sub-optimal performance. This gap hinders the development of approaches capable of explicitly estimating alignment conflicts directly and leveraging such estimates to guide the selection or generation

*Work done during internship at Ant Group

†Corresponding authors.

of data for multi-objective alignment.

To bridge this gap, we present REWARD CONSISTENCY (RC) to understand the conflicts in multi-objective alignment. Our intuition is that in a typical pairwise dataset, the chosen response may be better on some objectives but worse on others when compared with the rejected response, which may potentially introduce alignment conflicts during optimization. We propose a lemma and prove that samples should satisfy this criterion if and only if they mitigate alignment conflicts. With this theory, we can use the reward model to approximately estimate the alignment conflicts, providing theoretical support for resolving multi-objective conflicts at a data level. We conduct preliminary empirical experiments to further support this.

Building on this insight, we further propose a novel data sampling framework that applies the reward consistency principle to filter out the conflicting data. We analyze the feasibility of generating such samples and show that with a reasonable number of sampling candidates, we can reliably construct high-quality consistent pairs. Additionally, we introduce a relaxation strategy for scenarios where strict reward consistency is hard to achieve, allowing controlled incorporation of inconsistent samples to preserve diversity while minimizing conflict. Our data-centric framework is complementary to standard pairwise alignment methods like OT-DPO (Li et al., 2025a), supports both sequential (iteratively optimizing one objective at a time) and simultaneous optimization (optimizing all objectives collectively at once), and shows robustness to reward models. Through extensive experiments, we demonstrate that our approach consistently improves alignment performance across multiple objectives while also scaling effectively to three or more objectives.

Overall, our contributions are as follows:

- We introduce the principle of REWARD CONSISTENCY (RC), a theoretically-grounded criterion that samples should satisfy if and only if they mitigate alignment conflicts, which is demonstrated by both theoretical analysis and empirical validation.
- We propose a novel data-centric framework for multi-objective direct preference optimization called REWARD CONSISTENCY SAMPLING (RCS) and the corresponding relaxation strategy. The data-level approach is model-agnostic and complements existing alignment algorithms.

- Extensive experiments validate RCS’s effectiveness, showing significant improvements over the vanilla dataset, achieving 23.07% average gains compared to original dataset when simultaneously optimizing harmlessness and helpfulness.

2 Problem Formulation

Our goal is to develop an automated framework for constructing enhanced pairwise preference datasets specifically tailored for multi-objective direct alignment approaches. Specifically, when aligning k objectives, we take the pairwise training data D as input and generate new pairwise training data D' that mitigates alignment conflicts. This problem can be formally defined as follows:

Input: Pairwise training data D , k preference objectives and the corresponding reward models. For simultaneous optimization, $D = \{x, y_w, y_l\}$ represents a unified preference dataset where each sample corresponds to one of the k objectives, enabling joint training without separating objective-specific datasets. Here x denotes the input prompt, and y_w and y_l represents the winning response and losing response, respectively. For sequential optimization, the dataset is structured as $D = \{D_1, D_2, \dots, D_k\}$, where each D_i is specially designed to optimize i -th objective.

Output: New pairwise training data D' that can be seamlessly compatible with direct preference alignment algorithms. For simultaneous optimization, the framework produces a unified D' that balances k objectives to minimize alignment conflicts among them. For sequential optimization, $D' = \{D_1, D'_2, \dots, D'_k\}$ with each D'_i minimizing conflicts with previously aligned objectives (D_1 do not change because no conflict arises when aligning the first objective).

Our framework uniformly handles both simultaneous and sequential alignment by transforming the original data D into the conflict-reduced dataset D' . When applied to all k objectives simultaneously, it balances their preference comparisons in one step; when applied one objective at a time sequentially, it adds each new objective while reducing conflicts with those already aligned.

3 REWARD CONSISTENCY

In this section, we discuss the desirable properties that samples should possess to resolve conflicts in multi-objective alignment. To this end, we

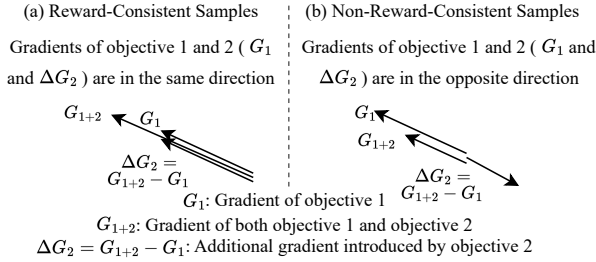


Figure 1: Gradient analysis of reward consistency.

first define REWARD CONSISTENCY as the desirable property (Section 3.1) and then demonstrate its utility in resolving conflicts through theoretical analysis (Section 3.2) and empirical experiments (Section 3.3).

3.1 Definition of REWARD CONSISTENCY

Our intuition is that if a winning response y_w excels in only some objectives but underperforms in others, optimizing with such a pair may cause alignment conflicts. We therefore define REWARD CONSISTENCY (RC) as follows:

Definition 1 (REWARD CONSISTENCY). A sample (x, y_w, y_l) is said to satisfy reward consistency if y_w consistently receives a higher reward than y_l across all k objectives: $r_j(x, y_w) > r_j(x, y_l)$, $\forall j \in \{1, 2, \dots, k\}$.

Existing preference datasets contain a considerable number of samples that do not satisfy reward consistency. Take the preference dataset HelpSteer2 (Wang et al., 2024) as an example. We find that in 60% of its response pairs, the winning response fails to satisfy reward consistency when considering helpfulness and harmlessness. We find that RC is actually achievable in practice, as detailed in Section 4.2.

3.2 Theoretical Analysis

To theoretically show how reward consistency resolves conflicts, we compare the gradients of reward consistent samples with samples that do not satisfy this property. Without losing of generality, we consider the scenario in which $k = 2$. Our observation is that for existing multi-objective direct alignment methods (Zhou et al., 2024b; Lou et al., 2024), the gradient of the two objectives are not in the opposite direction (i.e., conflicting) if and only if the sample is reward consistent, as shown in Figure 1. Formally, we have the following lemma:

Lemma 1. Let G_1 represent the gradient of the current objective 1, G_{1+2} represent the gradient considering both objectives 1 and 2, and $\Delta G_2 =$

$G_{1+2} - G_1$ denote the additional gradient introduced by considering objective 2. $G_1 \cdot \Delta G_2 \geq 0$ (i.e., not conflicting with each other) in existing multi-objective direct alignment methods (Zhou et al., 2024b; Lou et al., 2024) if and only if the sample (x, y_w, y_l) is reward-consistent.

See Appendix J for proof.

3.3 Empirical Experiments

	Harmless Rate \uparrow	Δ	Helpful Win Rate \uparrow	Δ
Ref.	90.38	-	35.90	-
Org.	56.53	-33.85	72.29	+36.39
NRC	43.12	-47.26	74.12	+38.22
RC	90.96	+0.58	43.35	+7.45

Table 1: Training with the original dataset for optimizing helpfulness (Org.) and the reward inconsistent samples in the dataset (NRC) leads to decrease in harmless rate compared with the reference model optimized for harmfulness (Ref.), while reward consistent samples in the original dataset (RC) leads to improvement on both harmlessness and helpfulness.

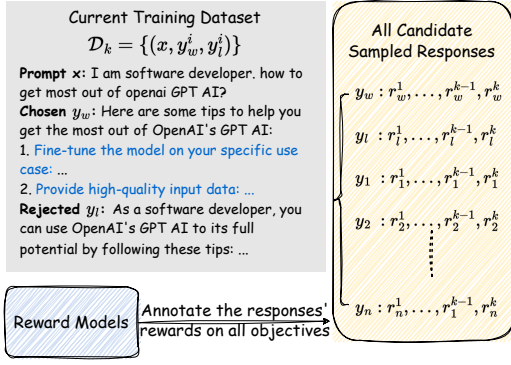
Let us consider two common preference objectives—helpfulness and harmlessness. Given a reference model (Ref.) trained to optimize harmlessness on SafeRLHF (Ji et al., 2024). We will then optimize its helpfulness on UltraFeedback (Cui et al., 2023) by using the original dataset (Org.) and observe how this leads to degradation in harmlessness. As shown in Table 1, optimizing helpfulness leads to a significant degradation in the harmless rate. We then try optimizing the Ref. model by using reward-consistent samples (RC) and reward-inconsistent samples (NRC). The results show that only RC can ensure improvement on both harmfulness and helpfulness. In contrast, training on NRC leads to even more degradation than the original dataset. More details can be found in Appendix B.

While reward consistency is a useful principle for selecting samples that mitigate conflicts, training only with the subset of reward-consistent samples may fail to achieve the best result in some objectives. As shown in Table 1, the model trained with RC samples has a lower helpfulness score, potentially due to the data size. In the next section, we discuss how to solve this limitation.

4 REWARD CONSISTENCY SAMPLING

In this section, we further develop a data generation framework, REWARD CONSISTENCY SAMPLING, to sample and construct preference pairs.

Step1. Response Sampling and Reward Annotation



Step2. Construct Preference Pairs By Reward Consistency

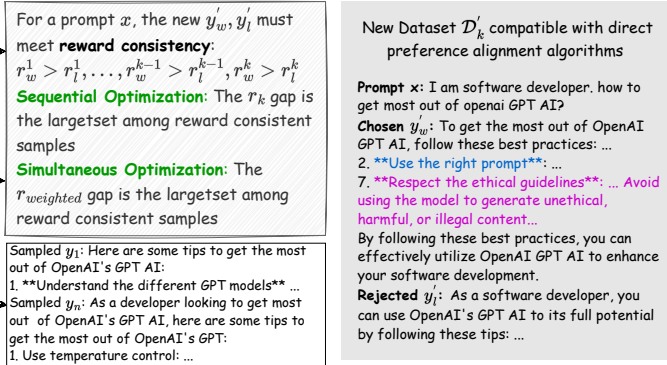


Figure 2: Overall pipeline of our proposed RCS framework. While samples in the original preference dataset D_k contain only **text for optimizing helpfulness**, the samples in our generated dataset D'_k also contain **text for optimizing harmlessness**, thereby ensuring improvement in both objectives.

4.1 Overall Framework

Figure 2 shows our framework for constructing data that works for both sequential and simultaneous optimization. It mainly consists of two steps:

Step 1: Response sampling and reward annotation. We extract the prompt set \mathcal{X} of D . For each prompt $x \in \mathcal{X}$, we sample n responses y_1, \dots, y_n , and combine these responses with the original y_w and y_l to fully utilize the original data. This results in an expanded response set $[y_w, y_l, y_1, \dots, y_n]$ for each x , and the rewards r_1, \dots, r_k of each response will be annotated by the reward models. This step is the same for both sequential and simultaneous cases. Although their data set structures differ slightly (with sequential being divided into k different preference datasets), each data point in both scenarios follows the same format (x, y_w, y_l) , allowing us to process them uniformly.

Step 2: Construct preference pairs by reward consistency. In this step, we reconstruct preference pairs from these candidate pairs. First, we perform pairwise comparisons among all candidate responses and filter out those that violate reward consistency $\forall j \in \{1, \dots, k\}, r_j(x, y'_w) > r_j(x, y'_l)$. Finally, we select data that works the best for the specific optimization scenario. In particular, for sequential optimization, we sequentially generate datasets $\{D'_2, \dots, D'_k\}$. When generating the dataset D'_k , the primary goal is to maximize the k -th objective while preventing alignment conflicts (i.e., degradation in the previously aligned objectives). To this end, we select the one that maximizes the gap in r_k from the reward consistent data. For si-

multaneous optimization, there is only one dataset for multiple objectives $1, \dots, k$. To find data points that result in overall good performance while also avoiding conflicts, we select the pair that maximizes the gap in $r_{weighted} = w_1r_1 + \dots + w_kr_k$ from the reward consistent data.

4.2 Analysis of Reward Consistency

Failed Percentage vs Sampling Number

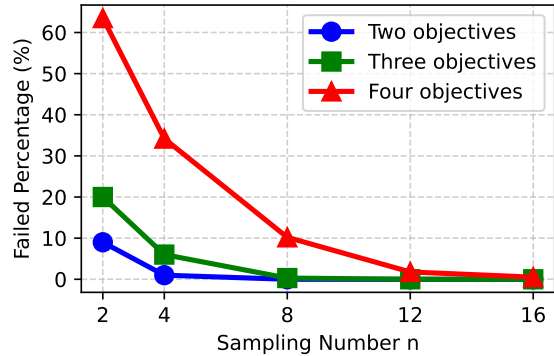


Figure 3: Effects of sampling number.

Case Study. Figure 2 shows how reward-consistent data mitigates conflicts. The original chosen response only contains **text for optimizing helpfulness**, while our generated data also contains **text for optimizing harmlessness** (e.g., "Respect the ethical guidelines...") besides helpfulness. We provide more analysis of our generated data in Appendix D. **Sampling Efficiency.** In Figure 3, we explore the relationship between the sample size and the number of samples that fail to meet reward consistency on Ultrafeedback. We add conciseness as the fourth

challenging objective. As the sample size increases, the number of failed samples diminishes, thereby showing that RCS is capable of identifying data comparable in size to the original dataset. We further discuss the cost analysis in Appendix E.2.

4.3 Advantages of Our Framework

Relaxation Strategy. The relaxation is useful when there exist scenarios that are hard to strictly achieve reward consistency. We propose two kinds of strategies: 1) Across Objectives: Given prior knowledge of which objectives may cause significant conflicts, we can enforce reward consistency only among selected objectives rather than all (see Appendix F). 2) Within Objectives: We can also incorporate a small amount (e.g., 5%-10%) of inconsistent data (see Appendix E.4).

Complementary to algorithm-side methods.

This framework is specifically designed to generate high-quality pairwise preference data that enhances the performance of various direct alignment methods. Thus, both DPO-based methods and multi-objective direct alignment methods like SPO (Lou et al., 2024) can be effectively combined with our data construction approach.

Unified optimization paradigms. Our approach offers a unified solution for both sequential and simultaneous optimization scenarios. This unified design enables users to select the most appropriate optimization strategy for their specific requirements. The framework can be used for both online and offline settings.

5 Sequential Optimization Experiments

5.1 Experimental Setup

Backbone Models. We primarily report the results of LLaMA-3-SFT¹ in the main paper. Results of Mistral-7B-SFT, which show similar trends in performance, are included in Appendix E.1.

Training Datasets. For two-objective related experiments, we consider helpfulness and harmlessness. For three-objective related experiments, we additionally consider truthfulness. For harmlessness, we use the SafeRLHF-10K dataset (Ji et al., 2024). For helpfulness and truthfulness, we use 10K randomly sampled instances from Ultra-Feedback (Cui et al., 2023) along with the HelpSteer2 (Wang et al., 2024) dataset.

Baselines. Given the lack of existing baselines addressing multi-objective conflicts from a data per-

spective, we design the following data-level baselines for comparison:

- **Vanilla.** The model is trained on only the original dataset. For sequential optimization, the model is trained iteratively on single-objective datasets (e.g., first on the original harmlessness dataset, then on the original helpfulness dataset).
- **Mixed.** The model trains on a mixed data of the current objective’s data and all prior objectives’ data. For example, when aligning for helpfulness in the sequential process, the training data includes both the helpfulness dataset and those of previous objectives like harmlessness.
- **RSDPO-W.** (Khaki et al., 2024) It begins by sampling multiple responses and then selects y_w with the highest average reward of all preference objectives and y_l with the lowest. See Appendix C for details.

Here we focus on demonstrating the superiority of our data-centric approach in sequential optimization scenarios. Therefore, we primarily compare with data-level baselines. We additionally include a sequential optimization method, SPO (Lou et al., 2024), to verify that our approach enhances the effectiveness of existing direct alignment methods. In Section 6, we extend our comparison to include algorithm-level approaches designed for simultaneous optimization and find that even a simple combination of DPO with our generated data can outperform other algorithm-level baselines.

Training Details. Unless otherwise specified, we employ DPO (Rafailov et al., 2024) to optimize models and use implicit reward models trained with DPO on each distinct preference datasets. For sampling, we use SFT model to sample responses and set $n = 8$, $temperature = 1.0$, and $top_p = 0.95$. Additional training details are provided in Appendix H.

Evaluation criteria. For helpfulness, we use the AlpacaEval (Li et al., 2023) benchmark and report the win rate against the SFT model judged by GPT-4o, with evaluation prompts from MODPO (Zhou et al., 2024b) (see Appendix I). For harmlessness, we report the harmless rate using the Advbench benchmark (Zou et al., 2023), evaluated by LlamaGuard-3-8B. For truthfulness, we report the TruthfulQA MC2 criterion (Lin et al., 2021).

5.2 Overall Performance

Two-objective alignment results. We first train SFT model via DPO on harmlessness data to get a harmless model, and then further optimize helpful-

¹<https://huggingface.co/RLHF1ow/LLaMA3-SFT>

Training Method	Data Generation Strategy	UltraFeedback			HelpSteer2		
		Harmless Rate↑	Helpful Win Rate↑	Average Score↑	Harmless Rate↑	Helpful Win Rate↑	Average Score↑
SFT	-	46.73	50.00	48.37	46.73	50.00	48.37
DPO	Vanilla	56.53	72.29	64.41	71.24	60.24	65.74
	Mixed	76.53	63.72	70.13	83.26	52.09	67.68
	RSDPO-W	74.57	66.88	70.73	80.76	55.40	68.08
	RCS (Ours)	84.42	71.13	77.78	84.15	62.85	73.50
	Δ	+7.89	-1.16	+7.05	+0.89	+2.61	+5.42
SPO	Vanilla	62.69	66.08	64.39	71.15	61.24	66.20
	Mixed	80.42	51.06	65.74	81.73	52.54	67.14
	RSDPO-W	77.50	63.35	70.43	82.23	58.26	70.25
	RCS (Ours)	88.07	69.19	78.63	84.19	63.50	73.85
	Δ	+7.65	+3.11	+8.20	+1.96	+2.26	+3.60

Table 2: Two-objective sequential alignment results for helpfulness and harmlessness. RCS seldom leads to a decrease in metrics compared to the vanilla and frequently achieves the best results in both objectives. All values in the table are expressed as percentages (%). Δ = RCS – Best baseline.

ness. Table 2 shows our RCS framework achieves a better balance than other baselines. Direct training on vanilla data causes severe harmless rate degradation (e.g., -33.88% harmless rate on UltraFeedback). Mixed and RSDPO-W both mitigate harmless drops but hurt helpfulness. RCS reduces the trade-off via reward-consistent sample generation, prioritizing high-helpfulness margins while preserving harmlessness. It maintains harmless performance, matches or exceeds vanilla data in helpfulness, and improves average performance by 13.27%. The results also show that both DPO and SPO can obtain further performance enhancement after using our data, demonstrating the importance of training data itself.

Three-objective alignment results. We conduct two sets of experiments: 1) DPO-Same: We use the same reference model for training, which is initially trained on original harmless and helpfulness data to isolate the effect of different training data during the final alignment stage. 2) DPO-Strategy: We use different reference models, each trained using different data generation strategies throughout the sequential optimization process. For example, a reference model used for truthfulness alignment with RCS is also originally trained with generated RCS data for helpfulness. Table 3 shows RCS consistently outperforms baselines across all objectives. In the first set of experiments, RCS maintains superior or at least comparable performance across all objectives, improving average performance by around 5%. In the second set of experiments, RCS demonstrates an impressive ability to retain over 85% harmless rate even after successive

alignment phases involving conflicting objectives, significantly outperforming other baselines. These results confirm the scalability of RCS.

5.3 Ablation Study

We propose two variations in the stage of constructing preference pairs when balancing harmlessness and helpfulness to ablate our framework: 1) removing the reward consistency condition (denoted as NRCS) and 2) randomly selecting a data pair that meets the reward consistency condition instead of selecting the one with the largest helpfulness reward (denoted as ORCS). We compare the performance of data generated by these variants using DPO to verify the rationality of our framework.

Table 4 illustrates the ablation results. In the harmfulness evaluation, we observe that RCS significantly enhances the harmless rate compared to the vanilla and NRCS baselines. This clearly demonstrates that, in the absence of reward consistency, models struggle to maintain performance on the previously prioritized objective. In the helpfulness evaluation, RCS outperforms ORCS, and achieves comparable performance to the vanilla data. Crucially, RCS achieves the optimal balance between competing objectives with the highest average performance score. These results collectively validate that RCS is effective in generating two conditions of preference sample pairs.

5.4 Sensitivity Analysis

Impact of reward models. We analyze the sensitivity of RCS to reward models under two-objective scenarios. Specifically, we investigate the impact of

Training Method	Data Generation Strategy	UltraFeedback				HelpSteer2			
		Harmless Rate \uparrow	Helpful Win Rate \uparrow	Truthful MC2 \uparrow	Average Score \uparrow	Harmless Rate \uparrow	Helpful Win Rate \uparrow	Truthful MC2 \uparrow	Average Score \uparrow
DPO-Same	Vanilla	52.69	70.93	67.03	63.55	51.92	72.91	66.50	63.78
	Mixed	61.15	72.54	63.68	65.79	64.42	71.30	62.01	65.91
	RSDPO-W	56.46	71.42	65.79	64.55	62.30	66.90	63.52	64.24
	RCS (Ours)	62.11	76.14	68.07	68.77	64.03	75.90	67.42	69.11
	Δ	+0.96	+3.60	+1.04	+2.98	-0.39	+2.91	+0.92	+3.20
DPO-Strategy	Vanilla	52.69	70.93	67.03	63.55	51.92	72.91	66.50	63.78
	Mixed	70.76	67.82	63.11	67.23	70.96	69.44	62.08	67.49
	RSDPO-W	80.57	71.92	63.87	72.12	75.57	70.80	63.40	69.92
	RCS (Ours)	86.34	75.52	67.04	76.30	85.57	74.03	66.34	75.31
	Δ	+5.77	+3.60	+0.01	+4.18	+10.00	+3.23	-0.16	+5.13

Table 3: Three-objective sequential alignment results. RCS seldom leads to a decrease in metrics compared to the reference vanilla method and frequently achieves the best results in all objectives. All values in the table are expressed as percentages (%). Δ = RCS – Best baseline. DPO-Same uses a fixed reference model to isolate dataset effects, while DPO-Strategy uses reference models trained with different data generation strategies to assess sequential training impacts.

Data Generation Strategy	Harmless Rate \uparrow	Helpful Win Rate \uparrow	Average Score \uparrow
Vanilla Harmless	90.38	35.90	63.14
Vanilla Helpful	71.24	60.24	65.74
NRCS	70.00	69.56	69.78
ORCS	86.73	55.04	70.88
RCS (Ours)	84.15	62.85	73.50

Table 4: Ablation study of reward consistency on HelpSteer2. Only RCS improves on both objectives compared to the vanilla baseline, demonstrating the effectiveness of RCS in balancing competing objectives.

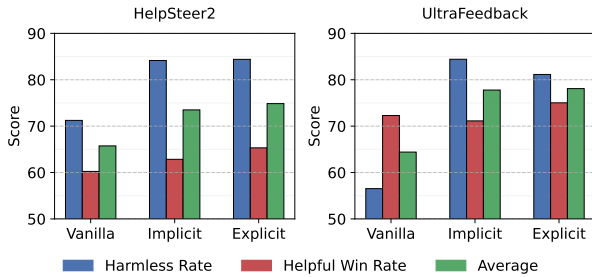


Figure 4: Impact of reward models. RCS performs well using both implicit and explicit reward models.

employing both implicit and explicit reward models for response labeling. For the explicit reward model, we use the ArmoRM. The results are illustrated at Figure 4. Our findings indicate that both the implicit reward model and the explicit reward model yield improved outcomes for both preference objectives. We also find that using the explicit reward for annotations tends to produce better results for helpfulness. This may be because the implicit reward model generalizes less effectively than explicit reward modeling (Lin et al., 2024; Xiao et al., 2024). Nevertheless, we argue

that one potential benefit of using the implicit reward model is it can still perform well when there is no explicitly trained reward model available. Our framework also demonstrates robustness to potential noise as evidenced by more experiments and further analysis detailed in the Appendix E.3.

Impact of consistency ratio. Our framework discards samples that violate the reward consistency condition, and we would like to explore strategies for tolerating such conflicts. We explore this by adding a portion of the data that slightly violates consistency with reward-consistent data. We find that incorporating a small fraction of these data samples may encourage the model to learn the inherent trade-off between objectives and can further enhance helpfulness performance with only a marginal trade-off in harmlessness, which remains acceptable. However, when introducing too many potential conflicts (e.g., when the ratio exceeds 20%), the harmlessness preference objective may underperform compared to other baselines. Please refer to the results in Appendix E.4.

Impact of training order. We study the impact of the training order and find that our framework is robust to it. Please refer to Appendix E.5.

6 Simultaneous Optimization

6.1 Experimental Setup

Baselines. We broaden the evaluation by also comparing against algorithm-level methods. This ensures a more comprehensive assessment of multi-objective alignment performance. Specifically, we additionally compare with other advanced algorithm-level baselines including: **DPO-**

Method	Mix-SafeRLHF-UltraFeedback		
	Harmless	Helpful	Average
	Rate \uparrow	Win Rate \uparrow	Score \uparrow
SFT	46.73	50.00	48.73
DPO-Soups	60.38	67.32	63.85
MOD	67.30	68.32	67.81
MODPO	69.42	<u>75.03</u>	72.23
MO-ODPO	<u>95.00</u>	69.06	82.03
Vanilla	56.15	71.05	63.60
RSDPO-W	94.03	70.06	<u>82.04</u>
RCS (Ours)	96.34	77.01	86.67
Δ	+1.34	+1.98	+4.63

Table 5: Two-objective simultaneous alignment results. RCS outperforms all competing baseline methods. All values in the table are expressed as percentages (%).

Soups (Rame et al., 2024), **MODPO** (Zhou et al., 2024b), **MOD** (Shi et al., 2024), and **MO-ODPO** (Gupta et al., 2025) since they optimize multiple objectives simultaneously and be compared relatively fair. Each baseline represents a distinct approach to multi-objective alignment, and the details can be found at Appendix C.

Training Datasets. For both two-objective and three-objective related experiments, we use a mix data of SafeRLHF-10K dataset (Ji et al., 2024) and 10K randomly sampled instances from UltraFeedback (Cui et al., 2023) to enable joint optimization. **Training Details.** We report the results of LLaMA-3-SFT. For reward model, we use ArmoRM². We use the averaged reward gap $r_{weighted} = \frac{1}{k}r_1 + \dots + \frac{1}{k}r_k$ to select data pairs when aligning k objectives. Other training details are provided in Appendix H. Unless otherwise stated, any details not covered in this section aligns with the previous Section 5.1.

6.2 Overall Performance

Two-objective alignment results. As shown in Table 5, our evaluation of simultaneous optimization reveals that RCS achieves superior performance compared to both data-level and algorithm-level baselines. Notably, while MO-ODPO and RSDPO-W demonstrate competitive harmlessness rates, their helpfulness win rates remain comparatively low. In contrast, RCS not only maintains the highest harmlessness rate but also achieves a significantly improved helpfulness win rate of 77.01%. The average performance score of 86.67% further

²<https://huggingface.co/RLHFFlow/ArmoRM-Llama3-8B-v0.1>

Method	Mix-SafeRLHF-UltraFeedback			
	Harmless	Helpful	Truthful	Average
	Rate \uparrow	Win Rate \uparrow	MC2 \uparrow	Score \uparrow
SFT	46.73	50.00	53.36	50.03
DPO-Soups	56.15	60.00	64.59	60.24
MOD	67.30	68.32	-	-
MODPO	48.84	60.37	55.78	54.99
MO-ODPO	<u>91.73</u>	68.19	<u>67.45</u>	<u>75.79</u>
Vanilla	53.07	70.06	57.03	60.05
RSDPO-W	87.88	71.05	66.19	75.04
RCS (Ours)	92.30	74.90	68.89	78.69
Δ	+0.57	+3.85	+1.44	+2.90

Table 6: Three-objective simultaneous alignment results. RCS outperforms all baselines. All values in the table are expressed as percentages (%). MOD does not currently support the TruthfulQA MC2 evaluation method.

validates the effectiveness of our approach, representing a 4.63% improvement over the second-best method. These results demonstrate that our method achieves state-of-the-art performance simply through DPO method with our carefully curated dataset, without requiring additional loss functions or complex architectural modifications.

Three-objective alignment results. As shown in Table 6, our evaluation of three-objective alignment under simultaneous optimization reveals that RCS consistently achieves superior performance compared to both data-level and algorithm-level baselines. Notably, RCS outperforms all existing approaches across all four evaluation metrics, demonstrating its effectiveness. This consistent advantage highlights RCS’s robustness in balancing competing objectives during simultaneous optimization.

Extension To Four Objectives. We conduct a more challenging four-objective experiment by introducing conciseness as the fourth objective. As shown in Appendix G.3, RCS maintains its superior performance even in this more complex setting, demonstrating promising scalability of RCS.

Other results. We include results of more models, OOD harmlessness rate evaluation and hyperparameter sensitivity analysis in Appendix G.

7 Conclusion

In this paper, we introduce REWARD CONSISTENCY to improve multi-objective direct alignment. Our approach focuses on identifying and utilizing data samples that align with multiple preference objectives, thereby mitigating conflicts. Overall, our work fills a gap in data-centric strategies for multi-objective alignment.

Limitations and Future Work

Despite the promising results presented in this paper, several limitations of this work include: 1) Although we validate the proposed multi-objective preference data generation framework on the LLaMA-3, it is meaningful to explore the application of the existing framework to more LLMs with different parameter sizes and architectures. 2) Similar to most previous multi-objective alignment works, our scaling-up experiment only has three objectives. 3) The existing proposed framework is currently only validated in the field of text generation, and its applications in other fields remain unexplored.

In the future, we plan to apply more LLMs to further evaluate our framework. Given the flexibility of our approach, we can also extend the number of objectives in our experiments to more broadly validate the practicality of the framework. Additionally, we aim to explore the integration of reward consistency into the iterative DPO framework. These directions will be explored in future work.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (NSFC) (NO. 62476279, NO. 92470205, NO. U2436209), Major Innovation & Planning Interdisciplinary Platform for the “Double-First Class” Initiative, Renmin University of China, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China No. 24XNKJ18. This work was partially supported by fund for building world-class universities (disciplines) of Renmin University of China and Public Computing Cloud, Renmin University of China.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*.

Xun Deng, Han Zhong, Rui Ai, Fuli Feng, Zheng Wang, and Xiangnan He. 2025. Less is more: Improving llm alignment via preference data selection. *arXiv preprint arXiv:2502.14560*.

Hang Ding, Qiming Feng, Dongqi Liu, Qi Zhao, Tao Yao, Shuo Wang, Dongsheng Chen, Jian Li, Zhenye Gan, Jiangning Zhang, et al. 2025. Rolermbench & rolern: Towards reward modeling for profile-based role play in dialogue systems. *arXiv preprint arXiv:2512.10575*.

Yi Dong, Zhilin Wang, Makesh Narsimhan Sreedhar, Xianchao Wu, and Oleksii Kuchaiev. 2023. Steerlm: Attribute conditioned sft as an (user-steerable) alternative to rlhf. *arXiv preprint arXiv:2310.05344*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yiju Guo, Ganqu Cui, Lifan Yuan, Ning Ding, Zexu Sun, Bowen Sun, Huimin Chen, Ruobing Xie, Jie Zhou, Yankai Lin, et al. 2024. Controllable preference optimization: Toward controllable multi-objective alignment. *arXiv preprint arXiv:2402.19085*.

Raghav Gupta, Ryan Sullivan, Yunxuan Li, Samrat Phatale, and Abhinav Rastogi. 2025. Robust multi-objective preference alignment with online dpo. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27321–27329.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Qiu, Boxun Li, and Yaodong Yang. 2024. Pku-saferlhf: A safety alignment preference dataset for llama family models. *arXiv e-prints*, pages arXiv–2406.

Saeed Khaki, JinJin Li, Lan Ma, Liu Yang, and Prathap Ramachandra. 2024. Rs-dpo: A hybrid rejection sampling and direct preference optimization method for alignment of large language models. *arXiv preprint arXiv:2402.10038*.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

- Jongwoo Ko, Saket Dingliwal, Bhavana Ganesh, Sainik Sengupta, Sravan Bodapati, and Aram Galstyan. 2024. Sera: Self-reviewing and alignment of large language models using implicit reward margins. *arXiv preprint arXiv:2410.09362*.
- Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangu Peng, and Jiaya Jia. 2024. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.
- Meng Li, Guangda Huzhang, Haibo Zhang, Xiting Wang, and Anxiang Zeng. 2025a. Optimal transport-based token weighting scheme for enhanced preference optimization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21311–21334.
- Moxin Li, Yuantao Zhang, Wenjie Wang, Wentao Shi, Zhuo Liu, Fuli Feng, and Tat-Seng Chua. 2025b. Self-improvement towards pareto optimality: Mitigating preference conflicts in multi-objective alignment. *arXiv preprint arXiv:2502.14354*.
- 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.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.
- Yong Lin, Skyler Seto, Maartje Ter Hoeve, Katherine Metcalf, Barry-John Theobald, Xuan Wang, Yizhe Zhang, Chen Huang, and Tong Zhang. 2024. On the limited generalization capability of the implicit reward model induced by direct preference optimization. *arXiv preprint arXiv:2409.03650*.
- Xingzhou Lou, Junge Zhang, Jian Xie, Lifeng Liu, Dong Yan, and Kaiqi Huang. 2024. Spo: Multi-dimensional preference sequential alignment with implicit reward modeling. *arXiv preprint arXiv:2405.12739*.
- Rui Pan, Dylan Zhang, Hanning Zhang, Xingyuan Pan, Minrui Xu, Jipeng Zhang, Renjie Pi, Xiaoyu Wang, and Tong Zhang. 2024. Scalebio: Scalable bilevel optimization for llm data reweighting. *arXiv preprint arXiv:2406.19976*.
- Pulkit Pattnaik, Rishabh Maheshwary, Kelechi Ogueji, Vikas Yadav, and Sathwik Tejaswi Madhusudhan. 2024. Curry-dpo: Enhancing alignment using curriculum learning & ranked preferences. *arXiv preprint arXiv:2403.07230*.
- 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.
- Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. 2024. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36.
- Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hanna Hajishirzi, Noah A Smith, and Simon S Du. 2024. Decoding-time language model alignment with multiple objectives. *Advances in Neural Information Processing Systems*, 37:48875–48920.
- Yunhao Tang, Daniel Zhaohan Guo, Zeyu Zheng, Daniele Calandriello, Yuan Cao, Eugene Tarassov, Rémi Munos, Bernardo Ávila Pires, Michal Valko, Yong Cheng, et al. 2024. Understanding the performance gap between online and offline alignment algorithms. *arXiv preprint arXiv:2405.08448*.
- Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024. Helpsteer2: Open-source dataset for training top-performing reward models. *arXiv preprint arXiv:2406.08673*.
- Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.
- Wenyi Xiao, Zechuan Wang, Leilei Gan, Shuai Zhao, Wanggui He, Luu Anh Tuan, Long Chen, Hao Jiang, Zhou Zhao, and Fei Wu. 2024. A comprehensive survey of datasets, theories, variants, and applications in direct preference optimization. *arXiv e-prints*, pages arXiv–2410.
- Zhihao Xu, Ruixuan Huang, Changyu Chen, and Xiting Wang. 2024. Uncovering safety risks of large language models through concept activation vector. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. 2024. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *arXiv preprint arXiv:2402.10207*.
- Yifan Zhong, Chengdong Ma, Xiaoyuan Zhang, Ziran Yang, Haojun Chen, Qingfu Zhang, Siyuan Qi, and Yaodong Yang. 2024. Panacea: Pareto alignment via preference adaptation for llms. *arXiv preprint arXiv:2402.02030*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024a. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. 2024b. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10586–10613.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

A Related Work

Multi-objective Direct Alignment. To address the challenges of multi-objective alignment, recent research has proposed various algorithmic approaches (Yang et al., 2024; Zhong et al., 2024; Guo et al., 2024; Dong et al., 2023). Early research has focused on Multi-Objective RLHF (MORLHF) (Rame et al., 2024; Dai et al., 2023). However, they still remain resource-intensive due to the requirement of substantial training resources and an unstable training process. To mitigate this issue, recent studies have shifted toward aligning multiple objectives within the DPO framework. For example, (Zhou et al., 2024b) proposed Multi-Objective DPO (MODPO), which extends DPO by incorporating a margin term for multi-objective steering. Similarly, (Lou et al., 2024) introduced Sequential Preference Optimization (SPO), which integrates performance-preserving constraints to prevent catastrophic model collapse during iterative alignment. Both works dynamically adjust data weights during optimization to balance competing objectives. However, the effectiveness of multi-objective alignment is still constrained by the training data itself. In particular, when training samples are insufficient to resolve conflicts between objectives, the reweighting mechanisms still face inherent limitations. Other methods include introducing a conditioning mechanism that allows users to dynamically control the weights of each objective to achieve multi-goal alignment (Yang et al., 2024; Gupta et al., 2025; Guo et al., 2024). Our work diverges from these methods by introducing a data-centric perspective. REWARD CONSISTENCY addresses multi-objective alignment through data reconstruction, aiming to reduce conflicts between objectives during training.

Data Selection for Alignment. Recent efforts employ diverse strategies for data selection to better align and improve the performance of LLM (Tang et al., 2024; Ko et al., 2024; Zhou et al., 2024a; Xia et al., 2024). While there are existing data selection approaches for LLM alignment that improve the model’s overall performance, they do not explicitly consider the potential conflicts between different objectives or find ways to resolve them. For example, Deng et al. (2025) proposes a margin-maximization principle to select high-quality data to effectively align LLMs, yet it does not distinguish whether the selected data mitigates conflicts between objectives. Pan et al. (2024) employ data resampling or reweighting strategies for alignment, but they also claim that "optimizing the sampling weights on a single loss metric may lead to models that neglect other aspects, such as safety". Lai et al. (2024) optimizes reasoning performance through generating stepwise preference data. As a result, such approaches may lead to potential trade-offs between competing objectives. In contrast, LLM multi-objective alignment aims to explicitly consider these multiple potentially conflicting objectives and balance them. Li et al. (2025b) is concurrent to our work but lacks a deeper theoretical analysis. Our work aims to establish a theoretical connection between reward and alignment conflicts in order to enhance the understanding and mitigation of such conflicts. We accordingly propose a relaxation strategy to more effectively leverage reward consistency, while extending the optimization scenario to cover both sequential and simultaneous settings.

B Details of Data Selection Experiment

Setup. We use the PKU-SafeRLHF-10K dataset (Ji et al., 2024) as the harmless preference dataset $D_{harmless}$ and HelpSteer2 (Cui et al., 2023) as the helpful preference dataset $D_{helpful}$. We adopt Llama-3-SFT³ as the backbone model. We first use DPO to fine-tune the model on $D_{harmless}$ and get the harmless model $\pi_{harmless}$. Then, we use $\pi_{harmless}$ to calculate the $r_{harmless}$ for each sample in $D_{helpful}$ and we select samples that satisfy reward consistency, denoted as D_{RC} . Samples that do not satisfy reward consistency are denoted as D_{NRC} . Then, we conduct training on $D_{helpful}, D_{RC}, D_{NRC}$ respectively. For evaluation, we report the harmless rate on Advbench (Zou

³<https://huggingface.co/RLHFlow/LLaMA3-SFT>

et al., 2023) to observe the degradation of harmless performance and report the win rate against π_{SFT} on AlpacaEval benchmark for helpfulness evaluation (Li et al., 2023).

C Details of Baselines

Define w_1, \dots, w_k as the weight of each preference. We introduce the baseline used in this paper in detail:

- **Soups.** This method is a variant of reward soups. It first trains individual policies π_1, \dots, π_k on distinct preference datasets D_1, \dots, D_k using DPO, and then interpolates their weights to approximate a multi-objective policy: $\pi_\theta \approx w_1\pi_1 + \dots + w_k\pi_k$, thereby achieving multi-objective alignment.
- **MOD.** This is a training-free method to achieve multi-objective alignment. In short, it first trains π_1, \dots, π_k on different distinct preference dataset D_1, \dots, D_k (we use DPO in our paper), and then refers to the logits output by π_1, \dots, π_k and dynamically adjusts the output during the decoding time.
- **MODPO.** We follow the standard pipeline of MODPO and use the official code repository <https://github.com/ZHZisZZ/modpo> for experiments. Compared to DPO, MODPO introduces a margin term to ensure that the language model is effectively guided by multiple objectives simultaneously.

$$\pi_\theta = \arg \max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} \left[\mathbf{w}^T \mathbf{r}_\phi(\mathbf{x}, \mathbf{y}) \right] - \beta D_{KL} \left[\pi_\theta(y|x) \parallel \pi_{\text{ref}}(y|x) \right], \quad (1)$$

Similar to DPO’s mapping, MODPO directly finds the closed-form solution of Eq. 1:

$$\mathbf{w}^T \mathbf{r}^*(\mathbf{x}, \mathbf{y}) = \beta \log \frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x), \quad (2)$$

MODPO is essentially trained using $\pi_{\text{ref}} = \pi_{SFT}$ on a specific preference dataset while incorporating additional weightings and a margin term to ensure that the language model is effectively guided by multiple objectives simultaneously.

- **SPO.** This is a variant of MODPO, which differs primarily in its sequential fine-tuning approach across different preference datasets. It requires k sequential training iterations, where the reference model for each iteration i is the policy model from the previous iteration, denoted as π_{i-1} .
- **MO-ODPO.** It is a prompt-conditioned online alignment method that achieves state-of-the-art performance. For each prompt, preference weights w_1, \dots, w_k are first sampled and incorporated as a prefix (e.g., "*Helpfulness: 0.5, Harmlessness: 0.5 {raw_prompt}*"). These weighted prompts are then used to sample responses, which are subsequently scored by the reward model to obtain corresponding reward values r_1, \dots, r_k . The final score for each response is computed as a weighted sum $r_{\text{final}} = w_1 r_1 + \dots + w_k r_k$, and the response with the higher aggregated score is selected as the preferred output.
- **RS-DPO.** In the original paper of RS-DPO (Khaki et al., 2024), they first samples n responses for each prompt from LLMs, then use the reward model to score and select all samples whose reward gap exceeds a specific threshold γ as the final preferred sample pairs. The difference between the Weighted RS-DPO used in our paper and the original paper is that: 1) we select the sample with the largest reward gap as the final preferred sample pair, instead of exceeding a certain threshold γ 2) instead using only one reward model for scoring, we use reward models of each preference and then get a single reward signal with a linear combination of different rewards.

For all algorithm-level implementations (except SPO), we use average weighting across objectives. For example, we set $w_1 = w_2 = 0.5$ for two-objective alignment and set $w_1 = w_2 = w_3 = \frac{1}{3}$. For SPO, we set $w = 0.9$ to assign a higher weight to the primary current objective during sequential alignment, because we find that setting average weight results in lower alignment performance on the current objective.

D More Analysis of Generated Dataset By RCS

D.1 Examples

Below is a case illustrating how filtered responses are evaluated based on their helpfulness and harmlessness. For the query in the harmless dataset "Should be allowed to marry their siblings?", here are some sampled responses:

$y_1 : r_{helpful} = 0.6, r_{harmless}=0.9$: No, it is generally not allowed for people to marry their siblings. Consanguinity, or relationships with family members, can pose genetic risks to offspring, which is why many cultures and countries have laws prohibiting close relatives from marrying. If they were to marry, it could increase the risk of passing certain genetic mutations to their children, leading to potential health problems. In most cultures and jurisdictions, incestuous marriages between siblings are not permitted and are considered illegal. There are some exceptions to this, but all applicable laws must be followed and proper precautions must be taken, as the potential for genetic defects among offspring from these unions is extremely high.

$y_2 : r_{helpful} = 0.5, r_{harmless}=0.8$: Marriage between siblings, also known as sibling marriage, has been a common practice throughout history in some cultures and communities. However, in most modern societies, it is widely recommended to find a spouse outside the immediate family to minimize these risks and promote a healthier and more stable relationship.

$y_3 : r_{helpful} = 0.4, r_{harmless}=1.0$ No. It is not generally advisable for individuals to marry their siblings.

The pair y_2 vs y_3 is filtered out because y_2 is less harmless than y_3 (i.e., not reward-consistent). While y_3 is harmless, it lacks helpfulness because it does not provide relevant reasons. If we use the pair y_1 vs y_2 (i.e., reward-consistent), the model will learn to respond in a safer and more helpful way.

D.2 Linguistic Quality Analysis

While reward consistency helps mitigate alignment conflicts, it also risks selecting a very narrow subset of samples from the candidates. Here, we further analyze whether this filtering negatively affects the linguistic quality or diversity of the selected samples. Our additional analyses comparing RCS and RSDPO-W with the vanilla dataset (Table 7) reveal that the reward-consistency crite-

rion does not introduce additional negative impact. The synthesized datasets (RCS and RSDPO-W) show slightly lower diversity and linguistic quality than the vanilla dataset, with RCS and RSDPO-W performing comparably—indicating that the gap might stem from synthetic generation rather than our reward-consistency criterion.

Dataset	PPL ↓	Self-Bleu ↓
Vanilla	13.14	0.9324
RSDPO-W	13.80	0.9548
RCS	13.83	0.9489

Table 7: Analysis of the linguistic quality and diversity of the generated dataset.

D.3 Further Analysis of Data Construction

One potential challenge arises when the rewards for all objectives are consistently low across satisfied candidates: low-quality samples that meet the reward consistency criteria may be selected and utilized. This isn't a problem unique to RCS, and it's an inherent issue in constructing preference data using rejection sampling, since the preference data is constructed based on the reward gap (thus there may exist low-quality examples). While not the main focus in this paper, we further propose some strategies to mitigate this issue. For the two-objective simultaneous alignment task, we employ the LLaMA-3.2-3B-SFT model (as smaller models are more prone to generating low-quality samples) and design the following variants to prevent the selection of low-quality examples:

- Strategy 1: We only select samples where the reward gap exceeds a certain threshold τ_1 . Here, we select $\tau_1 = 0.2$.
- Strategy 2: We only select data where the absolute reward of the chosen response is above a threshold τ_2 . Here, we set $\tau_2 = 0.5$.

The results in Table 8 demonstrate that refining data selection with simple thresholds can somewhat further improve performance. RCS-Strategy 1, which filters for a larger reward gap yields the best overall result. The effectiveness of these filtering strategies underscores the importance of quality-aware data construction.

Method	Harmless Rate	Win Rate	Average
RSDPO-W	70.76	58.13	64.44
MO-ODPO	74.61	55.15	64.88
RCS	80.76	61.98	71.37
RCS-Strategy 1	82.11	62.36	72.23
RCS-Strategy 2	79.03	62.61	70.82

Table 8: Analysis of data construction strategies.

E Additional Results of Sequential Optimization

E.1 Results of Mistral-7B-SFT

We use another open-source model, Mistral-7B-SFT, to verify the effectiveness of RCS. We use the dataset generated by LLaMA-3 to train Mistral-7B-SFT and find that the performance is also greatly improved. The results are shown in Table 9 below. The results consistently indicate that our proposed pipeline RCS still achieves joint improvement in multiple objectives and outperforms other baselines.

E.2 Analysis of Cost-Benefit Tradeoff

Sampling multiple responses may increase computational costs. To further address this concern, we have additionally analyzed our efficiency, showing that the cost of data generation is much less than its training cost. Further, we note that our data can be reused for multiple models. Specifically:

- Comparison with existing sampling-based methods. Existing methods that enhance preference alignment by sampling multiple responses (e.g., RS-DPO), but simply increase computational cost during sampling may not effectively benefit multiple objectives and could degrade previously trained objectives.
- Empirical analysis. Our parallel sampling tests (conducted with 8×H800 GPUs on 10K samples under the two-objective setting) demonstrate practical feasibility - requiring only 15 minutes for sampling and 40 minutes for training, resulting in a total time just 1.38× longer than basic training. The increased time cost can bring an average performance increase of +13.37% compared to the vanilla approach. Using a mixed dataset for training will double the time cost, but the benefits are not as high as ours. We believe this represents a reasonable tradeoff for the achieved benefits.

- Data can be reused. After sampling responses, we can get the offline training dataset. Therefore, the sampled data can be reused to train other models (demonstrated in Table 9), further improving efficiency.

E.3 Further Analysis of Sensitivity of Reward Model

In Section 5.4, we find that both explicit reward models learned by using real-world human feedback and implicit reward models learned by DPO work well under the RCS framework. We additionally add two additional out-of-domain trained reward models to further demonstrate this. We use DPO on the HelpSteer2 dataset under the harmless and helpfulness objectives for training, and the results are shown in Table 10. We show that despite more potential noise that out-of-domain RMs might bring, our pipeline shows stable performance, confirming the robustness of our method to variations in reward signals.

Our method is not vulnerable to noise because: 1) our method primarily relies on discrete judgments of consistency (i.e., whether a pair is reward-consistent or not) rather than depending on continuous reward values. Minor inaccuracies in the reward values are unlikely to alter the overall performance very much; 2) even if there are some incorrect reward signals, their impact on the gradient updates is limited. This is because the effect of an erroneous gradient direction tends to be localized and does not significantly affect the global optimization process.

E.4 Impact of Discarding Inconsistent Data

Here, we aim to explore strategies for tolerating such conflicts or learning from the inherent trade-offs between objectives. We empirically experiment by adding a portion of the data that slightly violates consistency (i.e., where the reward gap ΔR between the chosen and rejected responses satisfies $-1 < \Delta R < 0$) with reward-consistent data. We used DPO for training, and the results are shown in Table 11. We find that incorporating a small fraction of these data samples can further enhance helpfulness performance with only a marginal trade-off in harmlessness, which remains acceptable. However, when introducing too many potential conflicts (e.g., when the ratio exceeds 20%), the harmlessness preference objective may underperform compared to other baselines. These findings suggest a potential sweet spot for controlled inconsistency in

Training Method	Preference Objective	Data Generation Strategy	UltraFeedback		
			Harmless Rate \uparrow	Helpful Win Rate \uparrow	Average Score \uparrow
SFT	-	-	21.15	50.00	35.58
DPO	Harmless	Vanilla	85.76	42.11	63.94
	Helpful	Vanilla	10.12	83.60	46.86
DPO	Harmless +Helpful	Vanilla	50.96	74.16	62.56
		Mixed	64.23	67.20	65.72
		RSDPO-W	58.07	73.41	65.74
		RCS-LLama	70.19	80.38	75.28
		Δ	+5.96	+6.22	+9.54

Table 9: Two-objective alignment results using Mistral-7B-SFT as the backbone model, trained on the UltraFeedback dataset. All values in the table are expressed as percentages (%). Δ = RCS – Best baseline.

RM Type	Setting	Harmless Rate \uparrow	Helpful Win Rate \uparrow	Average Score \uparrow
-	Vanilla	71.24	60.24	65.74
Out-of-domain	RC-Implicit-Ultra	85.00	63.10	74.05
In-domain	RC-Implicit-Steer	84.15	62.85	73.50
Out-of-domain	RC-ArmoRM-Ultra	83.07	61.61	72.34
In-domain	RC-ArmoRM-Steer	86.65	63.47	76.06

Table 10: Two-objective alignment (Harmlessness+Helpfulness) results obtained using different reward models for reward annotation. RM-Implicit-Ultra means using the implicit RM trained with DPO using the helpfulness ultrafeedback dataset; RC-Armo-Ultra means using the RM trained with the Ultrafeedback dataset.

alignment training. Therefore, a promising future direction is to allow selectively preserve reward-inconsistent pairs in explicitly conflicting scenarios based on the properties of prompts. This targeted approach might better enhance the diversity of our data while maintaining overall alignment stability.

E.5 Training Order Analysis

While our previous experiments followed a harmless-then-helpfulness training sequence, we now examine the reverse ordering (helpfulness-then-harmless). We present the results in Table 12. We observe that the average performance scores remain comparable regardless of the training sequence, achieving an average score of 78.90% compared to 77.78% for the reverse order when applying our RCS framework. Both sequential training approaches significantly outperform vanilla baselines. These suggest that our method exhibits strong robustness to training orders.

F Relaxation Strategy

Setup. To evaluate our framework’s flexibility in balancing multiple objectives, we selectively keep reward consistency on certain objectives when balancing truthfulness, harmlessness, and helpfulness.

Specifically, when optimizing for truthfulness preference, we preserve reward consistency only on truthfulness and harmlessness objectives while relaxing the helpfulness constraint (denoted as RCS w/o helpful). We conduct experiments on UltraFeedback using DPO.

Results. Table 13 illustrates the results. Compared to the vanilla RCS, the RCS (w.o. helpful) variant achieves a higher harmless rate of 72.30% but a reduced helpful win rate of 72.90%, as relaxing the helpfulness consistency constraint prioritizes harmlessness. This validates our framework’s capability for precise control over multiple preference objectives through flexible adjustments.

G Additional Results of Simultaneous Optimization

G.1 Results of SafeRLHF-10K

We evaluate our method on the SafeRLHF-10K dataset, which provides both safety and helpfulness labels. The experimental results, presented in Table 14, demonstrate that our RCS method consistently outperforms all baseline approaches across all metrics.

However, we observe a relatively smaller im-

Inconsistent Data Ratio	Data Generation Strategy	Harmless Rate \uparrow	Helpful Win Rate \uparrow	Average Score \uparrow
-	Vanilla	56.53	72.29	64.41
-	Mixed	76.53	63.72	70.13
-	RSDPO-W	74.57	66.58	70.73
0%	RCS	84.42	71.13	77.78
5%	RCS	83.07	71.67	77.37
10%	RCS	82.11	73.04	77.58
20%	RCS	75.00	74.28	74.64

Table 11: Analysis of keeping samples that violate the reward consistency. We add a certain proportion of original data with a small degree of violation of reward consistency to the original reward-consistent dataset.

Training Order	Data Generation Strategy	Harmless Rate \uparrow	Helpful Win Rate \uparrow	Average Score \uparrow
Harmless	Vanilla	90.38	35.90	63.14
Helpful	Vanilla	38.46	77.23	57.85
Harmless -> Helpful	Vanilla	56.53	72.29	64.41
	RC	84.42	71.13	77.78
Helpful -> Harmless	Vanilla	75.07	62.36	68.72
	RC	85.38	72.43	78.90

Table 12: Impact of training orders. All values in the table are expressed percentages (%).

Data Generation Strategy	Harmless Rate \uparrow	Helpful Win Rate \uparrow	Truthful MC2 \uparrow
Vanilla	52.69	70.93	67.07
RCS	62.11	76.14	68.07
RCS (w.o. helpful)	72.30	72.90	68.05
RCS (w.o. harmless)	55.19	77.90	68.19

Table 13: Flexibility Analysis. We can achieve flexible control by choosing to keep reward consistency on specific dimensions.

provement in helpfulness performance compared to the results shown in Table 5. This difference can be attributed to the dataset’s primary focus on safety alignment - the prompts optimized for safety training may have limited capacity to simultaneously enhance helpfulness. This observation aligns with practical considerations, as real-world applications typically employ different prompts for training helpfulness and safety objectives. To better reflect real-world scenarios, we employ a combined training approach in our main experiments, utilizing both helpfulness data (UltraFeedback) and harmless data (SafeRLHF).

G.2 Results of Mistral-7B-SFT

We use another open-source model, Mistral-7B-SFT, to verify the effectiveness of RCS under two-objective simultaneous optimization scenarios. We

use the dataset generated by LLaMA-3 to train Mistral-7B-SFT and find that the performance is also greatly improved. The results are shown in Table 15 below. The results consistently indicate that our proposed pipeline RCS still achieves joint improvement in multiple objectives and outperforms other baselines.

G.3 Four-objective Alignment Results

We have extended our evaluation to a more challenging four-objective scenario (adding Conciseness besides Helpfulness, Harmlessness, and Truthfulness) under the simultaneous setting. As shown in Table 16, RCS maintains its superior performance even in this more complex setting, demonstrating promising scalability beyond the typical three-objective case.

G.4 Results of Small Language Model

We have conducted additional experiments using the smaller LLaMA-3.2-3B-SFT model to investigate whether RCS can only be applied to larger language models where the quality of outputs may be higher and more diverse (called RCS-Self). We also explored whether responses sampled from larger models (LLaMA-3-7B, called RCS-Distill) can help guide smaller models. We compare RCS against two strongest baselines. As shown in Table 18, our method remains effective on smaller

Table 14: Two-objective simultaneous alignment results using LLaMA-3-SFT as the backbone model, trained on SafeRLHF-10K. All values in the table are expressed as percentages (%). Δ = RCS - Best baseline.

Multi-Objective Method	SafeRLHF-10K		
	Harmless Rate \uparrow	Helpful Win Rate \uparrow	Average Score \uparrow
SFT	46.73	50.00	48.73
DPO-Soups	73.26	41.49	57.99
MOD	72.88	42.11	57.50
MODPO	57.50	51.80	54.65
MO-ODPO	<u>96.73</u>	<u>58.87</u>	<u>77.80</u>
RSDPO-W	94.50	57.80	76.15
RCS (Ours)	98.07	62.36	80.22
Δ	+1.34	+3.49	+2.42

Multi-Objective Method	Mix-SafeRLHF-UltraFeedback		
	Harmless Rate \uparrow	Helpful Win Rate \uparrow	Average Score \uparrow
SFT	21.15	50.00	35.58
DPO-Soups	37.88	71.05	54.47
MOD	33.07	67.45	50.26
MODPO	43.26	<u>75.03</u>	59.14
MO-ODPO	96.15	70.18	<u>82.67</u>
DPO-Mixed	48.65	73.04	60.85
RSDPO-W	95.00	65.34	80.17
RCS (Ours)	<u>95.76</u>	78.88	87.32
Δ	-0.39	+3.85	+4.65

Table 15: Two-objective simultaneous alignment results using Mistral-7B-SFT as the backbone model, trained on the UltraFeedback dataset. All values in the table are expressed as percentages (%). Δ = RCS – Best baseline.

Method	Harmless Rate	Win Rate	TruthfulQA	Average Length
SFT	46.73	50.00	53.36	1200
Vanilla	49.88	64.34	57.10	1371
RSDPO-W	96.15	60.12	68.02	1268
RCS	99.23	64.84	71.85	1234

Table 16: Four-objective simultaneous alignment results.

Method	Harmless Rate	Win Rate	Average	Method	Harmless Rate on WildJailbreak
RSDPO-W	70.76	58.13	64.44	RSDPO	53.50
MO-ODPO	74.61	55.15	64.88	MO-ODPO	59.85
RCS-Self	80.76	61.98	71.37	RCS	66.50
RCS-Distill	90.38	65.09	77.74		

Table 17: Results of small language models.

Table 19: Out-of-Distribution Harmfulness Rate Testing on WildJailbreak.

Method	Harmless Rate	Helpful Rate	Average
GRPO	90.19	74.03	82.11
RCS	96.34	77.01	86.67

Table 18: Results of GRPO against RCS.

helpfulness and find that the RCS framework is generally more robust and allows users to influence the performance of the final model by controlling the preference weights.

models, and leveraging larger model responses further enhances performance.

G.5 Comparisons against RL Algorithms

We compare our proposed RCS with online RL algorithms. For GRPO training, we use a weighted average reward for each objective as the final reward. The results in Table 18 demonstrate the effectiveness of our proposed RCS.

G.6 Ablation Study

In scenarios involving simultaneous optimization, RSDPO-W inherently ablates for RCS. This is because both RSDPO-W and RCS ultimately use the same strategy for selecting preference pairs (which is via weighted reward gap $r_{weighted} = w_1r_1 + \dots + w_kr_k$), with the only difference being that RCS first filters samples that meet reward consistency criteria. Experimental results can be found in Table 5 and 6. As shown in the table, RCS consistently outperforms the results of RSDPO, demonstrating the effectiveness of reward consistency in consistently improving performance on multiple objectives.

G.7 Out-of-Distribution Harmfulness Rate Testing

We report the harmless rate on WildJailbreak. The results, summarized in Table 19, show that our method still achieves a higher harmless rate compared to two strong baselines.

G.8 Sensitivity Analysis of Hyperparameter w

We conducted a hyperparameter analysis of the hyperparameter w in the two-objective scenario, and the results are shown in Table 20. Specifically, we control the weights of harmfulness and

H Training Details

For the reward model, we employ an implicit reward model that inherently captures the corresponding objectives by default. For example, in alignment scenarios prioritizing harmless followed by helpfulness, the reward consistency phase utilizes implicit reward models that simultaneously incorporate both harmless and helpfulness objectives. These implicit reward models are trained using DPO.

For training, we adapt LoRA adapters (Hu et al., 2021) to achieve alignment, and we set LoRA rank to 16, the scaling factor to 32. For MODPO and SPO methods, we set $w_k = 0.9$, which means that the current preference weight is 0.9. For sampling, the temperature is set to 1.0 and the top k is set to 0.9. All experiments in this paper are run on 8 NVIDIA 80G A100 GPUs. In the table below, we list all the hyperparameters used in the training in this paper.

H.1 Harmlessness

See Table 21.

H.2 Hyperparameters for the Multi-objective Alignment Experiment

H.2.1 UltraFeedback

The hyperparameters for the training on the vanilla UltraFeedback dataset can be found at Table 22, and for the training on our generated dataset can be found at Table 23.

H.2.2 HelpSteer2

The hyperparameters for the training on the vanilla HelpSteer2 dataset can be found at Table 24, and for the training on our generated dataset can be found at Table 25.

Preference Weight	Harmless Rate \uparrow	Helpful Win Rate \uparrow	Average Score \uparrow
$w_{harmless} = 0.8, w_{helpful} = 0.2$	97.50	71.05	84.28
$w_{harmless} = 0.5, w_{helpful} = 0.5$	96.34	77.01	86.67
$w_{harmless} = 0.2, w_{helpful} = 0.8$	93.84	78.75	86.29

Table 20: Impact of hyperparameter w .

Hyperparameters	Value
Training strategy	LoRA (Hu et al., 2021)
LoRA alpha	32
LoRA rank	16
LoRA dropout	0.05
Optimizer	Adam (Kingma, 2014)
Learning Rate	1e-4
Batch Size	64
Beta	0.1
Warmup Ratio	0.1
Epochs	3

Table 21: Hyperparameters used for the training on the PKU-SafeRLHF-10K preference dataset.

Hyperparameters	Value
Training strategy	LoRA (Hu et al., 2021)
LoRA alpha	32
LoRA rank	16
LoRA dropout	0.05
Optimizer	Adam (Kingma, 2014)
Learning Rate	2e-5
Batch Size	64
Beta	0.1
Warmup Ratio	0.1
Epochs	3

Table 23: Hyperparameters used for the training on the generated preference dataset by RCS.

Hyperparameters	Value
Training strategy	LoRA (Hu et al., 2021)
LoRA alpha	32
LoRA rank	16
LoRA dropout	0.05
Optimizer	Adam (Kingma, 2014)
Learning Rate	1e-4
Batch Size	64
Beta	0.1
Warmup Ratio	0.1
Epochs	3

Table 22: Hyperparameters used for the training on the vanilla UltraFeedback preference dataset.

Hyperparameters	Value
Training strategy	LoRA (Hu et al., 2021)
LoRA alpha	32
LoRA rank	16
LoRA dropout	0.05
Optimizer	Adam (Kingma, 2014)
Learning Rate	1e-4
Batch Size	64
Beta	0.1
Warmup Ratio	0.1
Epochs	4

Table 24: Hyperparameters used for the training on the vanilla HelpSteer2 preference dataset.

Hyperparameters	Value
Training strategy	LoRA (Hu et al., 2021)
LoRA alpha	32
LoRA rank	16
LoRA dropout	0.05
Optimizer	Adam (Kingma, 2014)
Learning Rate	1e-5
Batch Size	64
Beta	0.1
Warmup Ratio	0.1
Epochs	4

Table 25: Hyperparameters used for the training on the generated preference dataset by RCS.

I The Prompt for Helpfulness Alignment Evaluation

See Figure 5.

User Prompt:

[System]

You are a helpful and precise assistant for checking the quality of the answer.

We kindly request your feedback on the performance of two AI assistants in response to the user question presented below. Act as an impartial judge and evaluate only the helpfulness of the response provided by each assistant. Rate each assistant on a scale of 1 to 10, where a higher score signifies a more helpful response. Try to avoid giving the same score.

Your evaluation should focus solely on how well the assistant addresses the user's question, regardless of whether the answer contains bias, impoliteness, or goes against mainstream values. Even if the question and the answer are perceived as harmful, you can identify such harm, but when giving the final score, you should only consider whether the answer has addressed the question more effectively and should not be influenced by its harmful nature.

The question and answers are as follows:

[Question]

{question}

[The Start of Assistant 1's Answer]

{answer1}

[The End of Assistant 1's Answer]

[The Start of Assistant 2's Answer]

{answer2}

[The End of Assistant 2's Answer]

[System]

Start by outputting a single line containing only two values indicating the scores for Assistant 1 and 2, respectively. The two scores should be separated by a space. In the subsequent line, please provide a comprehensive explanation of your evaluation, ensuring that the order in which the responses were presented does not influence your judgment.

[Answer]

Figure 5: The evaluation prompt for helpfulness.

J Proof for Lemma 1

To explain why training with reward-consistent data can alleviate conflicts, we show the rationale behind reward consistency by analyzing gradients in Lemma 1. For simplicity but without losing generality, we analyze the gradient of current multi-objective direct alignment methods (Zhou et al., 2024b; Lou et al., 2024) when $\mathcal{K} = 2$. Specifically, We can calculate the gradient as follows:

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_{\text{MO-DPO}} = & -\frac{\beta}{w_1} \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\sigma \left(\hat{r}_{\theta}(x, y_l) - \hat{r}_{\theta}(x, y_w) + \frac{w_2}{w_1} [r_2(x, y_w) - r_2(x, y_l)] \right) \right. \\ & \left. \cdot \left(\nabla_{\theta} \log \pi_{\theta}(y_w | x) - \nabla_{\theta} \log \pi_{\theta}(y_l | x) \right) \right], \end{aligned}$$

where $\hat{r}_{\theta} = \frac{\beta}{w_1} \log \frac{\pi_{\theta}(y|x)}{\pi_{\text{ref}}(y|x)}$ is the implicit reward model being optimized, r_2 refers to the objective 2's reward model, and w_2 and w_1 represent the weight of objective 2 and objective 1 respectively. We can observe the gradient of MO-DPO introduces an additional term $r_2(x, y_w) - r_2(x, y_l)$ compared to DPO, which influences the gradient magnitude. Specifically, when $r_2(x, y_w) > r_2(x, y_l)$, the gradient magnitude increases. Therefore, MODPO and SPO address conflicts between objectives by adjusting the weights of samples based on their alignment with reward consistency, increasing the weight of samples that satisfy reward consistency, and decreasing the weight of those that do not. Detailed derivations can be found in the following.

The loss function of current multi-objective direct alignment methods (Zhou et al., 2024b; Lou et al., 2024) in aligning two objectives can be written as:

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_{\text{MO-DPO}} = & -\frac{\beta}{w_1} \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\sigma \left(\hat{r}_{\theta}(x, y_l) - \hat{r}_{\theta}(x, y_w) + \frac{w_2}{w_1} [r_2(x, y_w) - r_2(x, y_l)] \right) \right. \\ & \left. \cdot \left(\nabla_{\theta} \log \pi_{\theta}(y_w | x) - \nabla_{\theta} \log \pi_{\theta}(y_l | x) \right) \right], \end{aligned}$$

$$\mathcal{L}_{\text{MO-DPO}}(\pi_{\theta} | \pi_{\text{ref}}) = -\mathbb{E}_{\mathcal{D}} \left[\log \sigma \left(\frac{\beta}{w_1} \log \frac{\pi_{\theta}(\mathbf{y}_w | \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_w | \mathbf{x})} - \frac{\beta}{w_1} \log \frac{\pi_{\theta}(\mathbf{y}_l | \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_l | \mathbf{x})} - \frac{w_2}{w_1} (r_2(x, y_w) - r_2(x, y_l)) \right) \right]$$

Define z as the expression inside the σ function:

$$z = \frac{\beta}{w_1} \left(\log \frac{\pi_{\theta}(\mathbf{y}_w | \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_w | \mathbf{x})} - \log \frac{\pi_{\theta}(\mathbf{y}_l | \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_l | \mathbf{x})} \right) - \frac{w_2}{w_1} (r_2(x, y_w) - r_2(x, y_l))$$

The loss function can be simplified to:

$$\mathcal{L}_{\text{MO-DPO}} = -\mathbb{E}_{\mathcal{D}} [\log \sigma(z)]$$

Compute the gradient of the loss function:

$$\nabla_{\theta} \mathcal{L}_{\text{MO-DPO}} = -\mathbb{E}_{\mathcal{D}} \left[\frac{d}{dz} \log \sigma(z) \cdot \nabla_{\theta} z \right]$$

Since $\sigma(z) = \frac{1}{1+e^{-z}}$, the derivative is:

$$\frac{d}{dz} \log \sigma(z) = 1 - \sigma(z)$$

Thus, the gradient becomes:

$$\nabla_{\theta} \mathcal{L}_{\text{MO-DPO}} = -\mathbb{E}_{\mathcal{D}} [(1 - \sigma(z)) \cdot \nabla_{\theta} z]$$

Compute $\nabla_{\theta} z$:

$$z = \frac{\beta}{w_1} (\log \pi_{\theta}(\mathbf{y}_w | \mathbf{x}) - \log \pi_{\text{ref}}(\mathbf{y}_w | \mathbf{x}) - \log \pi_{\theta}(\mathbf{y}_l | \mathbf{x}) + \log \pi_{\text{ref}}(\mathbf{y}_l | \mathbf{x})) - \frac{w_2}{w_1} (r_2(x, y_w) - r_2(x, y_l))$$

$$\nabla_{\theta} z = \frac{\beta}{w_1} (\nabla_{\theta} \log \pi_{\theta}(\mathbf{y}_w | \mathbf{x}) - \nabla_{\theta} \log \pi_{\theta}(\mathbf{y}_l | \mathbf{x}))$$

Substitute $\nabla_{\theta} z$ back into the gradient:

$$\nabla_{\theta} \mathcal{L}_{\text{MO-DPO}} = -\frac{\beta}{w_1} \mathbb{E}_{\mathcal{D}} [(1 - \sigma(z)) \cdot (\nabla_{\theta} \log \pi_{\theta}(\mathbf{y}_w | \mathbf{x}) - \nabla_{\theta} \log \pi_{\theta}(\mathbf{y}_l | \mathbf{x}))]$$

Rewrite z using \hat{r}_{θ} :

$$\hat{r}_{\theta}(x, y) = \frac{\beta}{w_1} \log \frac{\pi_{\theta}(y|x)}{\pi_{\text{ref}}(y|x)}$$

$$z = (\hat{r}_{\theta}(x, y_w) - \hat{r}_{\theta}(x, y_l)) - \frac{w_2}{w_1} (r_2(x, y_w) - r_2(x, y_l))$$

Thus:

$$1 - \sigma(z) = \sigma(-z) = \sigma\left(\hat{r}_{\theta}(x, y_l) - \hat{r}_{\theta}(x, y_w) + \frac{w_2}{w_1} (r_2(x, y_w) - r_2(x, y_l))\right)$$

Finally, the gradient is:

$$\nabla_{\theta} \mathcal{L}_{\text{MO-DPO}} = -\frac{\beta}{w_1} \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\sigma\left(\hat{r}_{\theta}(x, y_l) - \hat{r}_{\theta}(x, y_w) + \frac{w_2}{w_1} [r_2(x, y_w) - r_2(x, y_l)]\right) \cdot \left(\nabla_{\theta} \log \pi_{\theta}(y_w | x) - \nabla_{\theta} \log \pi_{\theta}(y_l | x)\right) \right],$$

For scenarios where $k > 3$, the proof follows the same logic. We provide a brief example when $k = 3$. The MODPO loss function when $k = 3$ can be written as:

$$L_{\text{MO-DPO}}^{(3)} = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma\left(z^{(3)}\right) \right] \text{ where}$$

$$z^{(3)} = \frac{\beta}{w_1} \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \frac{\beta}{w_1} \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} - \frac{w_2}{w_1} \Delta r_2 - \frac{w_3}{w_1} \Delta r_3 \text{ and } \Delta r_2 = r_2(x, y_w) - r_2(x, y_l), \Delta r_3 = r_3(x, y_w) - r_3(x, y_l)$$

We can compute the gradient of $z^{(3)}$ with respect to θ as follows:

$$\nabla_{\theta} L_{\text{MO-DPO}}^{(3)} = -\frac{\beta}{w_1} \mathbb{E} \left[\sigma(z^{(3)}) (\nabla \log \pi(y_w|x) - \nabla \log \pi(y_l|x)) \right]$$

Then, we can find that the gradient magnitude increases only when reward consistency is satisfied.