Weights-Rotated Preference Optimization for Large Language Models

Chenxu Yang[1,2]*, **Ruipeng Jia**[3]*, **Mingyu Zheng**[1,2], **Naibin Gu**[1,2], **Zheng Lin**[1,2]†,
Siyuan Chen[1,2], **Weichong Yin**[3], **Hua Wu**[3], **Weiping Wang**[1]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]Baidu Inc., Beijing, China
{yangchenxu,linzheng,wangweiping}@iie.ac.cn

Abstract

Despite the efficacy of Direct Preference Optimization (DPO) in aligning Large Language Models (LLMs), reward hacking remains a pivotal challenge. This issue emerges when LLMs excessively reduce the probability of rejected completions to achieve high rewards, without genuinely meeting their intended goals. As a result, this leads to overly lengthy generation lacking diversity, as well as catastrophic forgetting of knowledge. We investigate the underlying reason behind this issue, which is representation redundancy caused by neuron collapse in the parameter space. Hence, we propose a novel Weights-**Ro**tated **P**reference **O**ptimization (**RoPO**) algorithm, which implicitly constrains the output layer logits with the KL divergence inherited from DPO and explicitly constrains the intermediate hidden states by fine-tuning on a multi-granularity orthogonal matrix. This design prevents the policy model from deviating too far from the reference model, thereby retaining the knowledge and expressive capabilities acquired during pre-training and SFT stages. Our RoPO achieves up to a 0.5-point improvement on AlpacaEval 2, and surpasses the best baseline by 1.9 to 4.0 points on MT-Bench with merely 0.015% of the trainable parameters, demonstrating its effectiveness in alleviating the reward hacking problem of DPO.

1 Introduction

Despite achieving remarkable performance, large language models (LLMs) (OpenAI, 2023; Touvron et al., 2023; Bai et al., 2023; Yang et al., 2023, 2025a,b) still pose the risk of generating content that diverges from human expectations (Bai et al., 2022). To tackle this challenge, researchers introduced reinforcement learning from human feedback (RLHF) to improve the controllability of AI systems by simulating human preferences across various response options (Christiano et al., 2023; Ouyang et al., 2022; Stiennon et al., 2022; Dai et al., 2025). However, RLHF has been widely criticized for its training instability and high sensitivity to hyperparameters. Recently, researchers have developed RL-free direct alignment algorithms (Dong et al., 2023; Yuan et al., 2023; Zhao et al., 2023), with Direct Preference Optimization (DPO) standing out as a leading approach in this field.

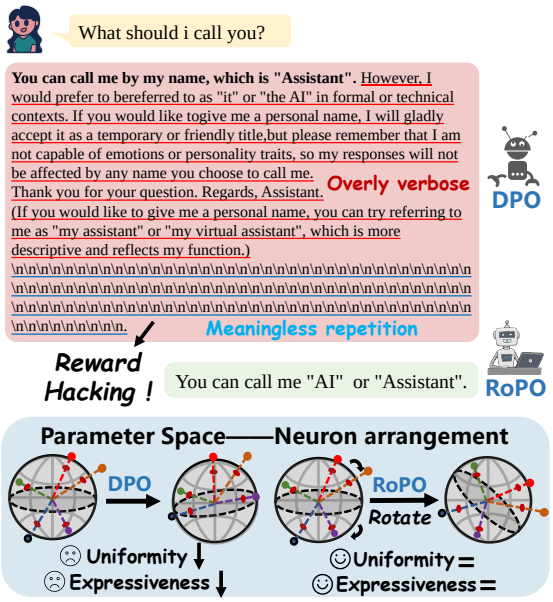


Figure 1: DPO suffers from reward hacking, causing the model's expressive capability to decline and generating overly long content with meaningless repetitions. The isotropic distribution of the neurons was disrupted during DPO. RoPO mitigates it by rotary tuning, which retains the angle-encoded expressive knowledge.

Unfortunately, there exists a significant drawback in DPO: *reward hacking*, also known as *reward overoptimization*. Deviating from the primary goal of learning the characteristics from preferred completions and discouraging undesirable behaviors in rejected ones, the policy model exploits vulnerabilities in the DPO function by overly suppressing the likelihood of rejected completions to

* Equal contribution.
† Zheng Lin is the corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 26164–26187
November 4-9, 2025

maximize rewards. Consequently, the fine-tuned model suffers from uncontrollable length, diminished diversity, and knowledge forgetting (Shen et al., 2023; Park et al., 2024). As shown in Figure 1, the DPO-trained model produces overly verbose output with meaningless repetitions when answering simple queries. Recently, several approaches have been introduced to address this challenge by modifying the constraints in the loss function (Azar et al., 2023; Wang et al., 2023; Zeng et al., 2024; Pal et al., 2024; Hong et al., 2024). However, these methods typically degrade alignment performance and perform worse than DPO.

In this paper, we investigate the underlying reasons behind the reward overoptimization of DPO from the perspective of parameter space. Our findings in Section 4.1 indicate that, compared to supervised fine-tuned (SFT) training, DPO optimization causes the model's neurons to collapse in the parameter space, leading to representation redundancy issues. To overcome this problem, we propose Weights-**Ro**tated **P**reference **O**ptimization (**RoPO**) algorithm, which simultaneously imposes constraints on both intermediate hidden layers from the parameter perspective and output layer from the logits perspective. The output layer is implicitly constrained using the original KL divergence from DPO to maintain the diverse and fluent expressions of the SFT model semantically, while the intermediate hidden layers are explicitly constrained by rotary-tuning with a multi-granularity orthogonal matrix to preserve angle-encoded knowledge. The dual constraints together prevent the policy model from deviating too far from the reference model, retaining the knowledge and expressions acquired during pre-training and SFT stages. Specifically, the specially designed multi-granularity orthogonal matrix consists of fine-grained Givens matrices and global Householder Reflection matrices.

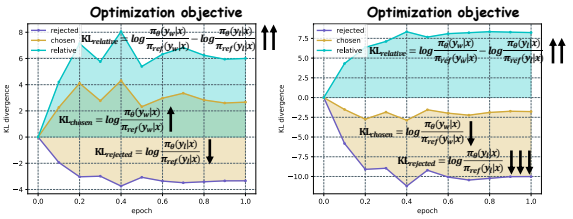
Extensive experiments show that RoPO achieves strong alignment performance while preventing excessively long generations, repetitive expressions, and catastrophic forgetting of knowledge. RoPO consistently outperforms all preference optimization baselines in all benchmarks with merely 0.015% of the trainable parameters. Specifically, RoPO surpasses the best baseline by 1.9 to 4.0 points on MT-Bench. By analyzing the training process, we discovered that the complementary constraints of RoPO prevent excessive reduction of the rejected completions likelihood while achieving higher reward accuracy, effectively mitigating the reward hacking problem.

Our contributions are summarized as follows:

- We delve into the underlying causes behind DPO reward hacking from a novel perspective.
- We propose an innovative method RoPO, which simultaneously imposes logit regularization and weight regularization to alleviate the reward hacking problem in DPO.
- The extensive experimental results across five benchmarks validate that our RoPO is comprehensively effective, with significantly reduced training parameters and training efficiency.

2 Related Work

Direct Preference Optimization. With the widespread application of LLMs, aligning them with human preferences has gained significant attention. Due to training instability and hyperparameter sensitivity in RLHF, recent studies have introduced several RL-free preference optimization methods (Dong et al., 2023; Zhao et al., 2023; Yuan et al., 2023). Rafailov et al. (2023) derived Direct Policy Optimization (DPO) theoretically by fitting an implicit reward function via reparameterization. As the most influential and effective approach, DPO significantly lowers the alignment barrier for LLMs. Numerous studies followed the proposition of DPO: IPO (Azar et al., 2023) revised the loss to minimize the disparity between the ratio of log-likelihoods and a given threshold. KTO (Ethayarajh et al., 2024) directly maximized the utility of generations. Meng et al. (2024) proposed SimPO, considering the average log-probability of a sequence as the implicit reward. Others aim to resolve different issues within the DPO objective function (Hong et al., 2024; Xu et al., 2024; Qi et al., 2024; Wang et al., 2023; Park et al., 2024; Liu et al., 2024b; Zhou et al., 2024). Among these, the four studies most relevant to ours are DPOP (Pal et al., 2024), SamPO (Lu et al., 2024), LD-DPO (Liu et al., 2024c) and InterDPO (Kojima, 2024). DPOP, SamPO, LD-DPO also focused on the reward hacking problem in DPO and attempted to fix the failure modes by adding a penalty term; InterDPO concentrated on the optimization of intermediate layers by calculating DPO loss at these layers. Different from their method, we incorporate



(a) Intended (b) Shortcut

Figure 2: The training objective was achieved through more excessive suppression on the rejected completions than on the chosen completions in DPO, suggesting a potential issue of reward hacking.

explicit regularization at the intermediate layers from parameter perspective.

Orthogonal Regularization. Orthogonal regularization has emerged as a powerful technique in deep learning, which is prevalent in research across the NLP (Mao et al., 2020; Smith et al., 2017; Conneau et al., 2018) and CV (Brock et al., 2016, 2018) fields. It is widely used to alleviate gradient vanishing and explosion by maintaining a constant norm (Brock et al., 2018) or to preserve the geometric structure of word vectors, thereby retaining semantic information within them (Smith et al., 2017). In contrast to their approaches, our RoPO introduces an orthogonal matrix from a parameter perspective for regularization rather than loss regularization, avoiding excessive deviation of the policy model during DPO training.

3 Preliminaries

3.1 Hyperspherical Energy

Hyperspherical energy (HE) was originally proposed to measure the uniformity of neuron arrangement in high-dimensional space (Liu et al., 2020). Higher HE indicates that neurons collapse in closely related directions, leading to representation redundancy; while lower HE suggests a more uniform arrangement, indicating better representation ability of the model. Suppose that there is a fully connected layer $\boldsymbol{W} = \{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_n\} \in \mathbb{R}^{d\times n}$, where $\boldsymbol{w}_i \in \mathbb{R}^d$ denotes the i-th neuron. The definition of HE is as follows:

$$\mathrm{HE}(\boldsymbol{W}) = \sum_{i\neq j} \|\hat{\boldsymbol{w}}_i - \hat{\boldsymbol{w}}_j\|^{-1}, \tag{1}$$

where $\hat{\boldsymbol{w}}_i = \frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}$ is the i-th normalized neuron.

3.2 DPO and its Reward Hacking

Direct Preference Optimization (DPO) is an optimization method that directly learns the policy bypassing the reward function (Rafailov et al., 2023).

$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) = - \mathbb{E}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)}\right)\right], \tag{2}$$

where x denotes the prompt, y_w denotes the chosen completion, y_l denotes the rejected completion, π_θ is the policy model, and π_{ref} is the reference model.

The objective of the DPO function could be regarded as optimizing the relative probability of choosing the chosen completion over the rejected one $p^*(y_w \succ y_l|x)$ in each pair of samples.

$$p^*(y_w \succ y_l|x) = \sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)} - \gamma\right), \tag{3}$$

where $\gamma = \beta\log\frac{\pi_{\mathrm{ref}}(y_w|x)}{\pi_{\mathrm{ref}}(y_l|x)}$ could be regarded as a constant as π_{ref} is not updated during training.

To minimize the DPO loss function, the policy model would try to increase the probability $p^*(y_w \succ y_l|x)$, that is, increasing the ratio $\frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}$. The original intention of human preference alignment was to learn the features that humans prefer in the chosen completions and suppress the undesirable behaviors in the rejected completions. However, subject to an upper bound 1, it is difficult to significantly modify $\pi_\theta(y_w|x)$ to increase the probability ratio. As a result, the policy model turns to push the probability of y_l as low as possible to achieve high rewards, leading to overoptimizing the spurious pattern of suppressing rejected examples. In some cases, DPO could lead to a reduction in the likelihood of chosen completions when the pair of completions holds small edit distances (Pal et al., 2024). We verified this by observing KL divergences alterations over 1 epoch in DPO-tuning on a held-out set. Figure 2 illustrates that the relative likelihood increases through more excessive suppression on the rejected completions than on the chosen completions. Reward hacking suppresses all behavioral characteristics in rejected completions, resulting in poor expressive ability and the deterioration of generation diversity.

4 Methodology

The Kullback-Leibler (KL) divergency in DPO prevents the policy model from deviating too far from the arrangement on which the reward model is accurate, as well as maintaining generation diversity and

preventing mode-collapse to single high-reward answers (Rafailov et al., 2023). However, they merely regulate updates on the output logits implicitly. We aim to add an explicit constraint from the perspective of parameter updates to suppress deviations, combining it with the implicit KL divergency to prevent reward hacking, enabling LLMs to better align human preference without sacrificing expressiveness power.

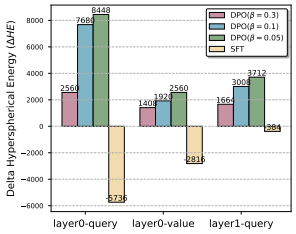
4.1 Parameter Perspective Analysis

To better design this explicit regularization, we conducted experiments to explore how DPO's parameter updates affect the model's expressiveness. As Liu et al. (2020) pointed out, the degree of uniformity in neuron placement influences model's expressiveness, and representation redundancy harms model performance. Qiu et al. (2024) held the view that some of the model's knowledge is contained within the relative angles between neurons. Drawing inspiration from their research, we hypothesize that the reward hacking problem in DPO is caused by the collapse of neuron arrangement, resulting in knowledge forgetting and verbose generation.

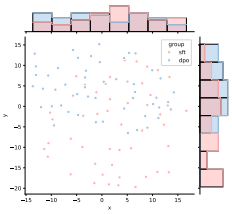
To validate our hypothesis, we compared the changes in hyperspherical energy of some layers in models trained by DPO and SFT, and used t-SNE visualization to observe the differences in neuron arrangement before and after DPO (van der Maaten and Hinton, 2008). Figure 3a shows that SFT reduces the model's hypersphere energy, while DPO leads to an increase. This suggests that the model learns new knowledge during the SFT stage, resulting in a more uniform arrangement of neurons. The reward hacking of DPO impairs expressive knowledge encoded in neuron angles. Moreover, with the weakening of the KL constraint ($\beta \downarrow$), the increase in hyperspherical energy becomes more notable, indicating that DPO's overoptimization is positively correlated with representation redundancy. From the visualization presented in Figure 3b, we observe that the neurons are more densely arranged along the y-axis after DPO, which is consistent with the result of hyperspherical energy.

4.2 Weights-Rotated Preference Optimization

Based on the above observations, we propose our Weights-**Ro**tated **P**reference **O**ptimization (**RoPO**) by adding an orthogonal regularization on intermediate hidden layers from the parameter perspective, while keeping the KL divergence regularization on output layer. The dual constraints together prevent



(a) Hyperspherical Energy variations. Increased HE indicates neuron arrangement collapse.



(b) Neuron arrangement changes of query vector in layer 0 after DPO.

Figure 3: Hyperspherical Energy variations of the query and value vectors in layer 0 and layer 1 after training.

the policy model from deviating too far from the reference model in the alignment process, retaining the angle-encoded knowledge and expressions acquired during pre-training and SFT stages. In the following, we introduce detailed design of the regularization on intermediate hidden layers.

Weight Decomposition Denoting $\boldsymbol{W} = \{\boldsymbol{w_1}, \ldots, \boldsymbol{w_n}\} \in \mathbb{R}^{d\times n}$ as the weight of policy model, we regard $\boldsymbol{w_i} \in \mathbb{R}^d$ as the i-th neuron. RoPO decomposes the weight into its magnitude vector $\boldsymbol{m} = \{m_1, \ldots, m_n\} \in \mathbb{R}^{1\times n}$ and directional matrix $\boldsymbol{W}/\|\boldsymbol{W}\|_c$, where $\|\cdot\|_c$ is the vector-wise norm of a matrix across each column, ensuring each neuron in the directional matrix remains a unit vector. We initialize the magnitude vector $\boldsymbol{m}$ as $\|\boldsymbol{W}\|_c$ and the policy model' weight $\boldsymbol{W}$ with the SFT model' weight $\boldsymbol{W}_{\text{SFT}}$ to ensure continuity for preference optimization.

$$\boldsymbol{W}' = \boldsymbol{m} \cdot \frac{\boldsymbol{W}}{\|\boldsymbol{W}\|_c} = \|\boldsymbol{W}\|_c \cdot \frac{\boldsymbol{W}}{\|\boldsymbol{W}\|_c}. \quad (4)$$

After decomposition, RoPO introduces a specially designed orthogonal matrix $\boldsymbol{R} \in \mathbb{R}^{d\times d}$ to tune the frozen directional weight $\boldsymbol{W}$ through rotation, keeping magnitude vector $\boldsymbol{m}$ trainable. Denoting $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{z} \in \mathbb{R}^n$ as the input and output vectors, the forward pass of RoPO is as follows:

$$\boldsymbol{z} = \boldsymbol{W}'^{\top}\boldsymbol{x} = \underline{\boldsymbol{m}} \cdot (\underline{\boldsymbol{R}} \cdot \frac{\boldsymbol{W}}{\|\boldsymbol{W}\|_c})^{\top}\boldsymbol{x}, \quad \text{s.t. } \boldsymbol{R}^{\top}\boldsymbol{R} = \boldsymbol{R}\boldsymbol{R}^{\top} = \boldsymbol{I}, \quad (5)$$

where $\boldsymbol{I}$ denotes an identity matrix, and the trainable parameters are denoted by an underline.

As illustrated in Figure 4, RoPO regulates the weights updating by performing rotation across d input dimensions and stretching each neuron $\boldsymbol{w_i}$ in the weight matrix to keep relative angular distances between weight vectors invariant between

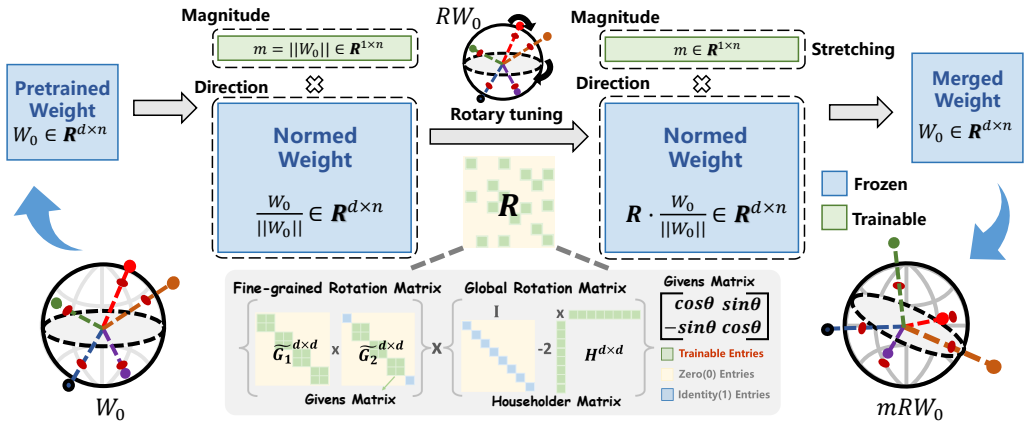


Figure 4: An overview of our RoPO method.

policy model and SFT model. The regularization ensures the angle-encoded knowledge acquired in the previous training stage preserves, which helps alleviate reward overoptimization. [1]

Multi-Granularity Orthogonal Matrix Specifically, we construct $\boldsymbol{R}$ with a combination of global rotation matrix $\tilde{\boldsymbol{H}} \in \mathbb{R}^{d\times d}$ and fine-grained rotation matrices $\tilde{\boldsymbol{G_1}}\tilde{\boldsymbol{G_2}} \in \mathbb{R}^{d\times d}$ as $R = \tilde{\boldsymbol{G_1}}\tilde{\boldsymbol{G_2}}\tilde{\boldsymbol{H}}$. We term it as multi-granularity orthogonal matrix, as the model weight is firstly rotated globally, and then rotated in each 2-d subspace planes locally. The orthogonality of the Multi-Granularity Orthogonal Matrix is formally proved in Appendix F.

The global rotation matrix is composed by two Householder Reflection matrices, rotating the weight matrix globally with the following matrix:

$$\tilde{\boldsymbol{H}} = (\boldsymbol{I} - 2\underline{\boldsymbol{u_1}}^\top \cdot \underline{\boldsymbol{u_1}}) \cdot (\boldsymbol{I} - \underline{\boldsymbol{u_2}}^\top \cdot \underline{\boldsymbol{u_2}}), \quad (6)$$

where $\boldsymbol{u_1}$ and $\boldsymbol{u_2} \in \mathbb{R}^d$ are two trainable unit vectors initialized as $[1, 0, \ldots, 0]_{1\times d}$.

The fine-grained rotation matrices are composed by $d-1$ Givens rotation matrices $\boldsymbol{G}$ with arrangement as follows:

$$\begin{aligned} \tilde{\boldsymbol{G_1}} &= \prod_{k=0}^{(d/2)-1} \boldsymbol{G}(2k, 2k+1; \theta_k), \\ \tilde{\boldsymbol{G_2}} &= \prod_{k=0}^{(d-1)/2} \boldsymbol{G}(2k+1, 2k+2; \theta_k), \end{aligned} \quad (7)$$

where the Givens matrix $\boldsymbol{G}$ rotates a vector in a 2-dimensional subspace planes, and the rotation angle is controlled by θ. Suppose we have the following Givens matrix, where $\cos\theta$ appears at $\{(i,i),(j,j)\}$, $\sin\theta$ appears at $\{(i,j),(j,i)\}$.

$$\boldsymbol{G}(i,j,\theta) = \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & \cos\theta & 0 & \sin\theta & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix}, \quad (8)$$

where the trainable parameter θ is initialized as 0 to ensure that $\boldsymbol{W'}$ equals to $\boldsymbol{W}$ before the finetuning.

The arrangement of Givens matrices in $\tilde{\boldsymbol{G_1}}\tilde{\boldsymbol{G_2}}$ ensures any d-dimensional rotation could be accomplished, demonstrating its sufficient fitting capacity. In other words, given any vector $\boldsymbol{v} \in \mathbb{R}^d$, $\tilde{\boldsymbol{G_1}}\tilde{\boldsymbol{G_2}}$ could rotate it to any vector $\boldsymbol{y} \in \mathbb{R}^d$ on the same sphere with $\boldsymbol{v}$, providing a full-angle coverage. The proof of this statement is detailed in the Appendix J. Besides, the fine-grained rotation matrices could incorporate a rapid implementation approach of sparse matrix multiplication like in Rotary Position Embedding (RoPE) (Su et al., 2023), accelerating the training process. The enhancement scheme is detailed in the Appendix I.

5 Experiments

5.1 Experimental Setup

Tasks. We evaluated our method on the open-ended instruction-following task, mathematical reasoning task, and the commonsense reasoning question-answering (QA) tasks. The experimental evaluation is organized into three main scenarios. First, we evaluate RoPO's alignment capability on instruction-following tasks, following the approach of most existing works. Second, we assess the level of knowledge forgetting in two different settings.

[1] RoPO does not introduce additional overhead during the inference stage since the parameter matrices can be merged.

Method	Mistral-Base (7B)							Llama2-Base (13B)						
	AlpacaEval 2			Arena-Hard		MT-Bench		AlpacaEval 2			Arena-Hard		MT-Bench	
	LC	LWR	Len.	LWR	Len.	LWR	Len.	LC	LWR	Len.	LWR	Len.	LWR	Len.
SFT	2.5	4.2	790	5.4	1157	12.8	808	2.3	3.7	1093	4.4	1602	8.7	985
DPO	9.7	10.2	1347	14.7	1614	14.8	1591	8.0	9.0	1379	11.2	1949	15.0	1211
SimPO	10.6	10.0	1562	13.7	1789	17.3	1613	8.4	9.3	1510	13.5	2053	12.5	1053
R-DPO	8.2	9.9	1273	12.6	1642	9.9	1340	7.9	9.3	1359	11.4	1894	7.3	1164
KTO	8.4	8.9	1286	12.9	1475	14.7	1106	4.8	5.2	1529	13.1	2026	4.3	1295
WPO	10.7	11.5	1506	15.0	1763	20.3	1679	8.0	9.4	1487	13.6	2029	17.8	1207
DPO*	8.1	9.3	1194	10.9	1470	13.8	1390	4.7	6.4	1263	7.9	1809	11.5	1096
IPO	5.0	8.8	1007	13.0	1347	11.1	929	4.3	6.8	1313	9.7	1863	10.1	1186
ORPO	4.5	5.7	1074	9.8	1345	13.4	992	3.1	3.5	1358	8.5	1824	10.6	1143
DPOP	8.4	9.2	1238	14.5	1512	16.8	1501	7.2	8.1	1355	9.7	1826	12.8	1176
Inter-DPO	8.0	8.7	1429	10.4	1611	12.4	1731	6.9	6.6	1319	9.6	2024	11.2	1209
DPO-LoRA	6.2	10.0	962	11.2	1421	16.0	920	5.1	8.9	1267	9.7	1888	6.2	1133
SamPO	11.2	13.0	1156	15.1	1583	22.1	1286	7.4	9.2	1308	13.2	1848	14.2	1208
LD-DPO	10.8	13.2	979	14.2	1401	20.6	1087	7.3	9.7	1199	11.9	1823	18.4	1161
RoPO	**11.4**	**13.5**	1042	**16.0**	1433	**24.0**	970	**8.9**	**10.1**	1336	**13.8**	1913	**22.4**	1177

Table 1: Evaluation results on three instruction following benchmarks: AlpacaEval 2, Arena-Hard, and MT-Bench. **LC** denotes length controlled win rate (%). **LWR** denotes length weighted win rate (%). **Len.** is the abbreviation of avarege generation length. The best results are highlighted with **bold**.

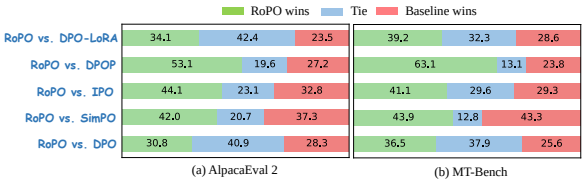


Figure 5: Human evaluation results on the AlpacaEval 2 and MT-Bench dataset (Meta-Llama-3-8B). The result is statistically significant with p-value < 0.05, and **Kappa** (κ) falls between 0.5 and 0.7.

Following Pal et al. (2024), we first apply supervised fine-tuning (SFT) and DPO on mathematical reasoning tasks. We then evaluate the model on both in-distribution (ID) mathematical reasoning benchmarks and out-of-distribution (OOD) commonsense QA benchmarks, in order to assess its retention of knowledge acquired during the pre-training stage. In addition, we first perform SFT on the commonsense QA training set, followed by SFT and DPO on instruction-following tasks. This setup is designed to evaluate how well the model retains knowledge learned during post-training.

Training details. We chose the popular Meta-Llama-3-8B (Dubey et al., 2024), Mistral-7B-v0.1 (Jiang et al., 2023), and Llama2-13B (Touvron et al., 2023) as the backbone model. For the open-ended instruction-following task, we supervised fine-tuning (SFT) a base model on the UltraChat-200k dataset (Ding et al., 2023) to acquire a SFT model. Subsequently, we conduct preference optimization on the UltraFeedback dataset (Cui et al., 2024) using the SFT model as the reference model. For the mathematical reasoning task, we supervised fine-tuning (SFT) a base model on the MetaMath-Fewshot dataset and then conduct preference optimization on MetaMath-DPO-FewShot dataset (Pal et al., 2024). For the commonsense reasoning QA tasks, it consist of 8 sub-tasks, each of which is equipped with a predefined training and testing set. We merged these 8 training sets into an integrated training set. For more training details, please refer to Appendix A.

Baselines. We compare RoPO with the following offline preference optimization methods: DPO (Rafailov et al., 2023), SimPO (Meng et al., 2024), R-DPO (Park et al., 2024), KTO (Ethayarajh et al., 2024), WPO (Zhou et al., 2024) DPO* ($\beta = 0.3$), IPO (Azar et al., 2023), ORPO (Hong et al., 2024), DPOP (Pal et al., 2024), Inter-DPO (Kojima, 2024), SamPO (Lu et al., 2024), LD-DPO (Liu et al., 2024c), and DPO-LoRA. The DPO* baseline is configured by augmenting the β parameter, which regulates the strength of the KL divergence in DPO. More detailed introductions of these baselines are given in the Appendix B.

Evaluation. We employed three challenging benchmarks to evaluate the open-ended instruction-following task: MT-Bench (Zheng et al., 2023), AlpacaEval 2 (Li et al., 2023), and Arena-Hard (Li et al., 2024). In addition, we also evaluate on the commonsense reasoning QA task after training on the instruction-following datasets to assess knowledge forgetting. We evaluate on the GSM8K (ID) (Cobbe et al., 2021), ARC (OOD) (Clark et al., 2018), and HellaSwag (OOD) (Zellers et al., 2019)

Method	New Task	Old Task								
	AlpacaEval 2	BoolQ	PIQA	SIQA	Hella.	Wino.	ARC-e	ARC-c	OBQA	Avg.
TS-FT	3.4	74.2	86.7	80.5	94.0	86.0	90.2	80.5	**88.4**	85.0
DPO	11.6	73.0	86.9	79.9	93.9	85.7	88.4	78.7	84.3	83.8
SimPO	12.7	33.6	79.4	74.8	63.9	80.0	81.6	70.3	80.0	70.5
R-DPO	11.4	25.3	84.7	76.7	59.2	80.4	83.6	72.6	80.0	70.4
KTO	8.0	44.5	85.0	72.5	93.4	78.6	82.7	72.3	80.8	76.2
WPO	13.5	74.0	85.8	81.0	93.9	85.4	89.6	79.9	84.6	84.3
IPO	7.6	**74.9**	**88.0**	81.4	94.3	86.0	**90.4**	80.7	85.0	85.1
DPOP	9.4	63.2	86.1	80.0	93.7	74.6	89.8	78.1	85.6	81.4
SamPO	13.0	73.0	86.5	79.3	93.0	86.1	88.3	78.4	84.9	84.2
LD-DPO	12.9	68.9	80.2	78.5	89.0	82.8	85.3	76.6	83.6	80.6
DPO-LoRA	9.4	74.4	**88.0**	80.9	**94.4**	86.0	90.2	80.3	85.6	85.0
RoPO	**13.7**	74.7	**88.0**	**81.5**	94.2	**86.3**	89.9	**82.0**	86.9	**85.4**

Table 2: Accuracy comparison of aligned **Meta-Llama-3-8B** model with various methods on 8 commonsense reasoning datasets. TS-FT denotes model fine-tuning on the training set of task-specific datasets.

Method	MATH (ID)	Commonsense (OOD)	
	GSM8K(5)	ARC(25)	HellaSwag(10)
SFT	75.1	58.9	81.3
DPO	77.3	57.2	81.0
SimPO	**77.5**	56.2	80.4
WPO	77.2	56.1	80.7
R-DPO	75.4	54.1	79.3
IPO	76.0	**58.7**	81.1
DPOP	76.9	57.6	80.9
Inter-DPO	74.3	58.0	80.3
SamPO	77.0	58.4	80.6
LD-DPO	76.5	57.8	80.0
DPO-LoRA	76.8	58.2	81.1
RoPO	77.3	**58.7**	**81.3**

Table 3: Evaluation results on GSM8K, ARC, and HellaSwag tasks on the Meta-Llama-3-8B model.

datasets when training on MetaMath. For the automatic evaluation metric, we utilize the length-controlled (**LC**) win rate and length-weighted win rate (**LWR**) against the GPT-4 to avoid the length bias issue in GPT-4 evaluation (Equation 9). In addition, to further mitigate the length bias inherent in GPT-based evaluations, we conducted human evaluations. Five well-educated annotators were asked to choose the superior response based on evaluation instructions in Figure 11. Further details of evaluation are provided in Appendix C.

5.2 Experimental Results

5.2.1 Main Results

RoPO consistently outperforms all the preference optimization baselines across instruction-following benchmarks. Table 1 shows that RoPO achieves state-of-the-art performance on LC and LWR while maintaining shorter generation lengths, effectively alleviating the verbosity issue commonly caused by DPO. As shown in Figure 7, RoPO demonstrates consistently strong performance on the one-turn questions of MT-Bench, excelling in role-play, reasoning, extraction, coding, and writing tasks. The human evaluations results in Figure 5 also show that RoPO fulfills instruction requirements with shorter generations.

RoPO effectively mitigates reward hacking issue in preference optimization algorithms. As previously observed, RoPO demonstrates superior performance with reasonably constrained generation length, suggesting that it effectively addresses the issue of verbosity. We now analyze RoPO's ability to alleviate reward hacking from the perspective of knowledge forgetting and repetition. Table 2 shows that, in the continual learning setting, RoPO achieves the best performance on the new instruction-following task while retaining strong results on the previous QA tasks, with improved accuracy across six datasets. We speculate that this is because RoPO better preserves the knowledge encoded in the angular relationships between neurons, while also acquiring additional commonsense knowledge during the alignment process. Table 3 shows that when mathematics is the target training task, RoPO achieves the second-best performance among all preference optimization methods on GSM8K, while outperforming all baselines on the out-of-distribution (OOD) commonsense knowledge datasets. Regarding generation repetitiveness, RoPO achieves Pareto optimal performance across the LWR and diversity metrics. In contrast, all baselines except IPO reduce the diversity of generated content significantly, as illus-

Method	Mistral-Base (7B)							Llama3-Base (8B)						
	AlpacaEval 2			Arena-Hard		MT-Bench		AlpacaEval 2			Arena-Hard		MT-Bench	
	LC	LWR	Len.	LWR	Len.	LWR	Len.	LC	LWR	Len.	LWR	Len.	LWR	Len.
RoPO	**11.4**	**13.5**	1042	**16.0**	1433	**23.6**	970	**13.7**	**11.9**	1083	**18.3**	1516	**28.3**	1027
w/o GRM	9.5	10.8	939	13.2	1274	19.6	946	10.6	9.4	917	14.4	1465	16.8	931
w/o FRM	10.2	11.7	1070	13.5	1459	21.8	1014	11.4	9.9	1092	16.9	1526	22.6	993
DPO-DoRA$_{r=4}$	10.0	10.6	967	14.9	1428	17.0	953	11.2	9.0	998	15.5	1513	20.7	994

Table 4: Ablation experiment results on three instruction-following benchmarks.

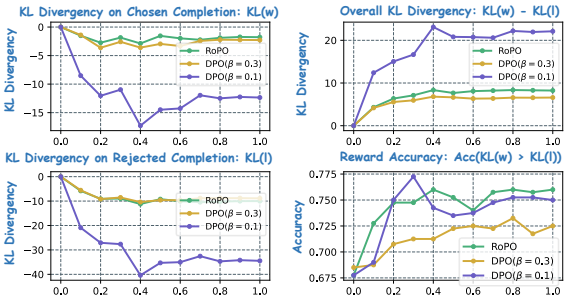


Figure 6: Comparison of KL divergences and reward accuracy evolution over 1 epoch during training on the UltraChat-200k (Meta-Llama-3-8B).

trated in Figure 8. The overall findings indicate that RoPO effectively mitigates the reward hacking issue while maintaining strong preference learning performance.

5.2.2 Ablation Study

To evaluate the efficacy of RoPO, we carried out ablation studies. In Table 4, we present results from ablating each key design of RoPO: (1) removing the global rotation matrix $\mathbf{H}$ (i.e. w/o GRM); (2) removing the fine-trained rotation matrix $\tilde{\boldsymbol{G}_1}\tilde{\boldsymbol{G}_2}$ (i.e. w/o FRM); (3) removing the rotation design in the directional matrix, and degradating to DoRA (Liu et al., 2024a). The results illustrate that every design of RoPO is crucial as eliminating each design would result in varying degrees of performance degradation, indicating that the design of multi-granularity orthogonal matrix enhances the outcomes of optimization process. DoRA could be regarded as one version of weight regularization without retaining relative angular distances between weight vectors constant. Experimental results shows that although DoRA also curbed the over-expansion of length bias, its instruct-following ability declines significantly.

5.3 Qualitative Examples

To intuitively demonstrate the effectiveness of RoPO, we compare the responses generated by RoPO with different baselines. The examples presented in Table 5 and Table 6 show that reward overoptimization could significantly degrades the model's performance on simple tasks, resulting in repetitive or irrelevant generations human dislike. RoPO effectively mitigates the degeneration phenomenon. Table 7 showcases a more challenging mathematical problem. SimPO's response was highly disorganized, while DPO exhibited a commonsense hallucination by suggesting that the probability of an event could exceed 1. In contrast, although RoPO's answer was incorrect, its reasoning process was consistent with that of GPT-4. Table 8 and Table 9 highlights that RoPO possesses outstanding open-domain generation capability.

5.4 Analysis

In this section, we will delve deeper into RoPO's underlying mechanisms. The variations depicted in Figure 6 indicate that RoPO exhibits a similar trend to DPO*, effectively preventing the excessive suppression of generation probabilities on both chosen and rejected completions, while maintaining reward accuracy on par with DPO. Additionally, its convergence curve is smoother and more stable, further validating the effectiveness of the regularization. We also conducted a visual analysis of the neuron arrangement using the t-SNE technique (van der Maaten and Hinton, 2008). As shown in Figure 10, DPO training leads to a disruption in the isotropy of sampled neurons, whereas RoPO maintains the integrity of the neuron arrangement.

6 Conclusion

In this paper, We delve into the underlying causes behind DPO reward hacking from a fresh parameter perspective. Based on the analysis, we proposed RoPO, which implicitly constrains the output layer logits with the KL divergence inherited from DPO and explicitly constrains the intermediate hidden states by fine-tuning on a multi-granularity orthogonal matrix. Experiments on extensive benchmarks

demonstrate that RoPO not only enhances alignment performance but also avoids reward hacking.

Limitations

RoPO effectively addresses reward hacking problem in DPO. However, we acknowledge certain limitations in our work.

(1) Currently, we have only applied the RoPO constraint to the query and value vectors within the attention layers. In the future, we intend to extend this approach to additional layers.

(2) In the analysis section, given that some baselines do not have a reference model, we did not plot their KL divergence changes. We will do further analyses on these baselines in the future.

(3) Due to limitations in resources and time, we were unable to validate the generalizability of our method on more and larger models. In the future, we plan to evaluate our approach on a wider range of models.

Ethics Statement

The benchmark datasets we utilized in our experiments are all highly respected, open-source datasets. The backbone LLMs are also open-sourced. They were compiled with strict adherence to user privacy protection protocols, ensuring the exclusion of any personal information. Moreover, our proposed approach is conscientiously designed to uphold ethical standards and promote societal fairness, guaranteeing that no bias is introduced. We use AI writing in our work, but only for polishing articles to enhance their readability.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62472419, 62472420).

References

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. 2020. Intrinsic dimensionality explains the effectiveness of language model fine-tuning.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback.

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale gan training for high fidelity natural image synthesis. *ArXiv*, abs/1809.11096.

Andrew Brock, Theodore Lim, James M. Ritchie, and Nick Weston. 2016. Neural photo editing with introspective adversarial networks. *ArXiv*, abs/1609.07093.

Siyuan Chen, Qingyi Si, Chenxu Yang, Yunzhi Liang, Zheng Lin, Huan Liu, and Weiping Wang. 2024. A multi-task role-playing agent capable of imitating character linguistic styles.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. Deep reinforcement learning from human preferences.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge.

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.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Ultrafeedback: Boosting language models with scaled ai feedback.

Muzhi Dai, Chenxu Yang, and Qingyi Si. 2025. S-grpo: Early exit via reinforcement learning in reasoning models.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore. Association for Computational Linguistics.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han-

wen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization.

Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. 2024. Parameter-efficient fine-tuning with discrete fourier transform.

Naibin Gu, Peng Fu, Xiyu Liu, Ke Ma, Zheng Lin, and Weiping Wang. 2025a. Adapt once, thrive with updates: Transferable parameter-efficient fine-tuning on evolving base models.

Naibin Gu, Peng Fu, Xiyu Liu, Bowen Shen, Zheng Lin, and Weiping Wang. 2024. Light-PEFT: Lightening parameter-efficient fine-tuning via early pruning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7528–7541, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Naibin Gu, Zhenyu Zhang, Xiyu Liu, Peng Fu, Zheng Lin, Shuohuan Wang, Yu Sun, Hua Wu, Weiping Wang, and Haifeng Wang. 2025b. Beamlora: Beam-constraint low-rank adaptation.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp.

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.

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

Atsushi Kojima. 2024. Intermediate direct preference optimization.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In

Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 110–119, San Diego, California. Association for Computational Linguistics.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline.

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`.

Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. 2023. Scaling down to scale up: A guide to parameter-efficient fine-tuning.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024a. Dora: Weight-decomposed low-rank adaptation.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J. Liu, and Jialu Liu. 2024b. Statistical rejection sampling improves preference optimization.

Wei Liu, Yang Bai, Chengcheng Han, Rongxiang Weng, Jun Xu, Xuezhi Cao, Jingang Wang, and Xunliang Cai. 2024c. Length desensitization in direct preference optimization.

Weiyang Liu, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. 2020. Learning towards minimum hyperspherical energy.

Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, Yandong Wen, Michael J. Black, Adrian Weller, and Bernhard Schölkopf. 2024d. Parameter-efficient orthogonal finetuning via butterfly factorization.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Junru Lu, Jiazheng Li, Siyu An, Meng Zhao, Yulan He, Di Yin, and Xing Sun. 2024. Eliminating biased length reliance of direct preference optimization via down-sampled kl divergence.

Xinyu Ma, Xu Chu, Zhibang Yang, Yang Lin, Xin Gao, and Junfeng Zhao. 2024. Parameter efficient quasi-orthogonal fine-tuning via givens rotation.

Xin Mao, Wenting Wang, Huimin Xu, Yuanbin Wu, and Man Lan. 2020. Relational reflection entity alignment. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*, CIKM '20, page 1095–1104. ACM.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward.

OpenAI. 2023. Chatgpt. https://openai.com/blog/chatgpt/.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with dpo-positive.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization.

Biqing Qi, Pengfei Li, Fangyuan Li, Junqi Gao, Kaiyan Zhang, and Bowen Zhou. 2024. Online dpo: Online direct preference optimization with fast-slow chasing.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. 2024. Controlling text-to-image diffusion by orthogonal finetuning.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

Wei Shen, Rui Zheng, Wenyu Zhan, Jun Zhao, Shihan Dou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Loose lips sink ships: Mitigating length bias in reinforcement learning from human feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2859–2873, Singapore. Association for Computational Linguistics.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2022. Learning to summarize from human feedback.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. 2023. Beyond reverse kl: Generalizing direct preference optimization with diverse divergence constraints.

Lanrui Wang, Jiangnan Li, Chenxu Yang, Zheng Lin, Hongyin Tang, Huan Liu, Yanan Cao, Jingang Wang, and Weiping Wang. 2025. Sibyl: Empowering empathetic dialogue generation in large language models via sensible and visionary commonsense inference.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open large-scale language models.

Chenxu Yang, Qingyi Si, Mz Dai, Dingyu Yao, Mingyu Zheng, Minghui Chen, Zheng Lin, and Weiping Wang. 2025a. Test-time prompt intervention.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Zheng Lin, Li Cao, and Weiping Wang. 2025b. Dynamic early exit in reasoning models.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. 2024. Token-level direct preference optimization.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Wenxuan Zhou, Ravi Agrawal, Shujian Zhang, Sathish Reddy Indurthi, Sanqiang Zhao, Kaiqiang Song, Silei Xu, and Chenguang Zhu. 2024. Wpo: Enhancing rlhf with weighted preference optimization.

A More Implementation Details

We discover that hyperparameter tuning is of paramount significance for attaining the optimal performance of preference optimization approaches. Hence, to acquire the supreme performance, we executed a sophisticated hyperparameter search. Below, we exhibit the hyperparameter configurations in the experiment.

Regarding the SFT training, we train models by utilizing the UltraChat-200k dataset with the

subsequent hyperparameters: a learning rate of 1e-6, a batch size of 128, a maximum sequence length of 2048, and a cosine learning rate schedule with 10% warmup steps for 1 epoch. All the models are trained with an Adam optimizer.

For the preference optimization stage, we train the SFT models using the UltraFeedback dataset with the same hyperparameters as SFT training under the full-parameter settings . The learning rate was set as 2e-5 for DPO-LoRA and DPO-DoRA, 1e-3 for RoPO.

For the commonsense reasoning QA task, we conducted experiment on it aimed at evaluating reward hacking. we first trained a base model on this commonsense reasoning QA training set until convergence, and then conducted SFT and preference optimization on the instruction-following dataset. The reward hacking problem could be manifested by comparing the performance on the testing set, as it leads to knowledge forgetting.

For the training framework, we implement our experiments on the open-sourced unified LlamaFactory (Zheng et al., 2024).

B More details of Baselines

DPO: Rafailov et al. (2023) derived it by fitting an implicit reward function through the reparameterization.

IPO: Azar et al. (2023) revised the objective to minimize the disparity between the ratio of log-likelihoods and a given threshold to mitigate the overfitting problem of DPO.

KTO: Ethayarajh et al. (2024) proposed it to directly maximize the utility of generations instead of maximizing the log-likelihood of preferences.

ORPO: Hong et al. (2024) integrates a penalty term to preclude the learning of undesirable responses while augmenting the probability of learning preferred ones.

R-DPO: Park et al. (2024) attempted to add a length regularization term in the loss function to alleviate the abnormally long generation issue.

SimPO: Meng et al. (2024) removed the reference model and used the average log probability of a sequence as the implicit reward.

WPO: Zhou et al. (2024) focuses on improving alignment performance by simulating on-policy preference data using off-policy preference data.

Inter-DPO: Kojima (2024) revised the DPO loss as the weighted sum of the final-layer DPO and intermediate DPO losses.

SamPO: Lu et al. (2024) identifies an algorithmic bias toward length reliance in DPO, where longer responses receive disproportionately larger gradient updates. Their solution involves downsampling to eliminate this length dependence.

LD-DPO: Liu et al. (2024c) attributed DPO's length sensitivity to how text length affects likelihood and proposed to decompose the response likelihood into public-length and excessive-length components, then reducing weights for the latter to mitigate verbosity preferences.

To acquire the supreme performance of each baseline, We conducted a parameter search based on the search space provided in the SimPO paper and the papers introducing the baselines. The following hyperparameter configuration was selected based on the best results obtained on the validation set: DPO: $\beta = 0.1/0.3$, IPO: $\tau = 2.0$, KTO: $\lambda_l = \lambda_w = 1.0, \beta = 0.1$, ORPO: $\lambda = 0.1$, R-DPO: $\alpha = 0.003, \beta = 0.1$, SimPO: $\beta = 2.0, \gamma = 0.5$, DPOP: $\beta = 0.1, \lambda = 5$, Inter-DPO: $\beta = 0.1, \gamma = 0.9$, WPO: $\beta = 0.1$, SamPO: $\beta = 0.1$, LD-DPO: $\beta = 0.1, \alpha = 0.3$,

C More Evaluation Setups

We employed three challenging benchmarks to evaluate the instruction-following task: MT-Bench (Zheng et al., 2023), AlpacaEval 2 (Li et al., 2023), and Arena-Hard (Li et al., 2024). Among them, AlpacaEval 2 consists of 805 questions. MT-Bench contains 80 questions falling into the following eight common categories: writing, role-play, extraction, reasoning, math, coding, knowledge I (STEM), and knowledge II (humanities/social science). Arena-Hard is an extension of MT-Bench, which collects 500 challenging high-quality prompts, achieving a state-of-the-art agreement with human preference rankings. For decoding hyperparameters in evaluation, we use a sampling decoding strategy to generate responses, with a temperature of 0.95, top-p of 0.7, and top-k of 50.

We present the findings using automatic and LLM-based evaluation methods. For LLM-based evaluation, we employ gpt-4o-2024-08-06 as the judge model to conduct pairwise comparisons for each preference optimization method and GPT-4. We consider the win rate (WR) against the responses generated by GPT-4 on AlpacaEval 2, MT-Bench, and Arena-Hard. We provide the prompt for evaluation in Table. To alleviate positional bias, we assess each candidate in both positions within

two separate runs, and the ultimate result is calculated as the average of the two runs. For MT-Bench, we additionally report the average MT-Bench score with gpt-4o-2024-08-06. Since GPT-4 has a tendency to give higher scores to longer responses during evaluation (Dubois et al., 2024), the length should be taken into account simultaneously when comparing performances. Under similar scores, we hold that shorter responses are superior. Therefore, we report the length-weighted win rate (LWR) as the final results by multiplying with the ratio of the content length generated by GPT-4 to that generated by the current method to exclude the influence of length as shown in shown in the following formula.

$$\text{LWR} = \text{WR} \cdot \frac{\text{Len}(y_{\text{g4}})}{\text{Len}(y)} \tag{9}$$

where y_{g4} denotes responses generated by GPT-4, and Len denotes length of generation.

For automatic evaluation, given that the reward over-optimization in alignment process causes damage to the diversity of the generated content, we add the corresponding metrics to assess it. We utilize the Distinct N-grams metric to evaluate the generation diversity(Li et al., 2016) through calculating a geometric mean of Distinct N with $N = 1, 2, 3, 4$. For commonsense reasoning QA, we regard accuracy as the evaluation metric across 8 test sets.

$$\text{Diversity} = \sqrt[4]{\prod_{n=1}^{4} \text{Distinct-n}}. \tag{10}$$

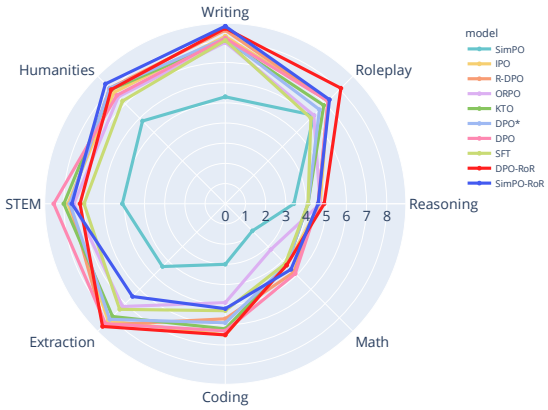


Figure 7: Scores of RoPO compared with baselines in MT-Bench on the Meta-Llama-3-8B backbone.

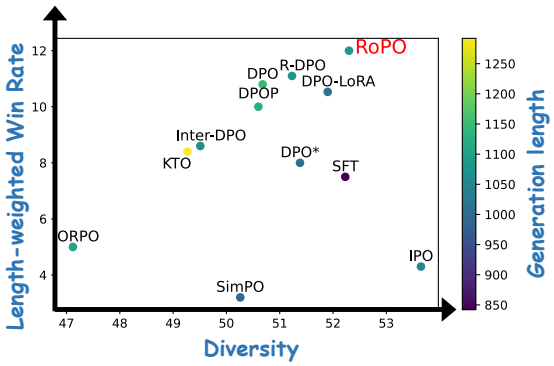


Figure 8: experimental results on Llama3-8B.

D More Analyses of Knowledge Retaining

To validate the issue of knowledge forgetting caused by alignment overfitting, we compare the influence of different preference optimization methods on the performance of commonsense reasoning tasks. Table 2 exhibits that DPO, KTO, and R-DPO resulted in a decline of the model's general ability. By observing some response cases, we discovered that the model appeared to directly answer the content of the options instead of answering the options themselves. It even provided safe response like "Sorry, I don't know." This implies that overfitting during the alignment training lead to the forgetting of task format knowledge and the decline of the model's question understanding ability. In contrast, RoPO still maintain the performance well on Commonsense Reasoning QA, with the accuracy rate on 6 datasets enhanced. We speculate that this is because RoPO retains the knowledge encoded in the angle between neurons well and acquires additional commonsense knowledge during the alignment process.

E β in KL divergency.

In the DPO objective, β governs the deviation from the reference model π_{ref}. We believe that through adjusting β, the alignment intensity can be regulated, thereby controlling reward hacking. The following is our explanation. Assuming that the true preference of a sample $p^*(y_w \succ y_l|x) = \hat{p}$, if the β value shrinks to approach 0, the model has to learn to further reduce the probability of rejected completions $\pi_\theta(y_l|x)$ to fit the true preference probability $\hat{p}$; if the β value increases, the model merely needs to learn to decrease the probability of rejected completions with a relatively weaker strength.

F Proof of Orthogonality

Definition of Householder Reflection Matrix

Given a non-zero vector $\mathbf{v} \in \mathbb{R}^n$, the corresponding Householder reflection matrix H is defined as:

$$H = I - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}},$$

where I is the identity matrix and $\mathbf{v}\mathbf{v}^T$ is the outer product matrix.

Properties of Householder Matrices

1. **Symmetry:** $H^T = H$.
   Proof:

$$\begin{aligned} H^T &= \left(I - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right)^T \\ &= I^T - 2\frac{(\mathbf{v}\mathbf{v}^T)^T}{\mathbf{v}^T\mathbf{v}} \\ &= I - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} = H. \end{aligned} \tag{11}$$

2. **Orthogonality:** $H^T H = I$.
   Proof:

$$H^T H = HH \quad (\text{since } H \text{ is symmetric})$$

$$\begin{aligned} HH &= \left(I - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right)\left(I - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right) \\ &= I - 4\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} + 4\frac{\mathbf{v}\mathbf{v}^T\mathbf{v}\mathbf{v}^T}{(\mathbf{v}^T\mathbf{v})^2}. \end{aligned} \tag{12}$$

   Since $\mathbf{v}^T\mathbf{v}$ is a scalar, we have:

$$\mathbf{v}\mathbf{v}^T\mathbf{v}\mathbf{v}^T = \mathbf{v}(\mathbf{v}^T\mathbf{v})\mathbf{v}^T = (\mathbf{v}^T\mathbf{v})\mathbf{v}\mathbf{v}^T.$$

   Therefore:

$$\begin{aligned} HH &= I - 4\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} + 4\frac{(\mathbf{v}^T\mathbf{v})\mathbf{v}\mathbf{v}^T}{(\mathbf{v}^T\mathbf{v})^2} \\ &= I - 4\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} + 4\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} = I. \end{aligned} \tag{13}$$

   Thus, $H^T H = I$, proving that H is orthogonal.

Product of Two Householder Matrices

Let H_1 and H_2 be two Householder matrices:

$$H_1 = I - 2\frac{\mathbf{v}_1\mathbf{v}_1^T}{\mathbf{v}_1^T\mathbf{v}_1}, \quad H_2 = I - 2\frac{\mathbf{v}_2\mathbf{v}_2^T}{\mathbf{v}_2^T\mathbf{v}_2}.$$

We want to show that $\tilde{H} = H_1H_2$ is orthogonal, i.e., $\tilde{H}^T\tilde{H} = I$.

1. Compute $\tilde{H}^T$:

$$\tilde{H}^T = (H_1H_2)^T = H_2^T H_1^T.$$

   Since H_1 and H_2 are symmetric ($H_1^T = H_1$, $H_2^T = H_2$):

$$\tilde{H}^T = H_2H_1.$$

2. Compute $\tilde{H}^T\tilde{H}$:

$$\tilde{H}^T\tilde{H} = (H_2H_1)(H_1H_2) = H_2(H_1H_1)H_2.$$

   Because H_1 is orthogonal ($H_1H_1 = I$):

$$\tilde{H}^T\tilde{H} = H_2IH_2 = H_2H_2.$$

   Since H_2 is orthogonal ($H_2H_2 = I$):

$$\tilde{H}^T\tilde{H} = I.$$

   Therefore, $\tilde{H} = H_1H_2$ is orthogonal.

Definition of Rotation Matrix

A rotation matrix $G \in \mathbb{R}^{n\times n}$ is a matrix that performs a rotation in Euclidean space. In $\mathbb{R}^2$, the rotation matrix by angle θ is:

$$G(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Properties of Givens Matrices

1. Compute the transpose:

$$G(\theta)^T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

2. Verify $R^T R = I$:

$$\begin{aligned} G^TG &= \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \\ &= \begin{bmatrix} \cos^2\theta + \sin^2\theta & -\cos\theta\sin\theta + \sin\theta\cos\theta \\ -\sin\theta\cos\theta + \cos\theta\sin\theta & \sin^2\theta + \cos^2\theta \end{bmatrix} \end{aligned}$$

   Using trigonometric identities:

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

3. Similarly, $GG^T = I$ can be verified.

Since the product of orthogonal matrices still satisfies orthogonality, the Multi-Granularity Orthogonal Matrix possesses orthogonality.

G Hyperspherical Energy

The initial proposal of Hyperspherical Energy (HE) was motivated by the diversification and balanced arrangement of neurons to prevent the parameter redundancy problem (Liu et al., 2020). Inspired by the renowned physics problem known as Thomson problem, Liu et al. (2020) designed the neural network training objective with Minimum Hyperspherical Energy (MHE) as the regularization. Assuming that there is a fully connected layer $\boldsymbol{W} = \{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_n\} \in \mathbb{R}^{d\times n}$, where $\boldsymbol{w}_i \in \mathbb{R}^d$ denotes the i-th neuron. The definition of HE is as follows:

$$\mathrm{HE}(\boldsymbol{W}) = \sum_{i\neq j} \|\hat{\boldsymbol{w}}_i - \hat{\boldsymbol{w}}_j\|^{-1} \tag{14}$$

where $\hat{\boldsymbol{w}}_i = \boldsymbol{w}_i/\|\boldsymbol{w}_i\|$ is the i-th normalized neuron.

$$\Delta\mathrm{HE}(\boldsymbol{W}) = \mathrm{HE}(\boldsymbol{W'}) - \mathrm{HE}(\boldsymbol{W}) \tag{15}$$

where $\boldsymbol{W'}$ denotes the aligned model, and $\boldsymbol{W}$ denotes the model before alinging.

Our RoPO satisfies the following equation:

$$\sum_{i\neq j} \|\hat{\boldsymbol{w}}_i - \hat{\boldsymbol{w}}_j\|^{-1} - \sum_{i\neq j} \|\hat{\boldsymbol{w}}_i^0 - \hat{\boldsymbol{w}}_j^0\|^{-1} = 0, \tag{16}$$

where $\hat{\boldsymbol{w}}_i^0$ denotes the weight before fine-tuning, and $\hat{\boldsymbol{w}}_i$ denotes the weight after fine-tuning.

H Training Efficiency

In addition to its outstanding comprehensive performance, RoPO also has the advantages of low trainable parameters. Compared with other preference optimization baselines (100% parameters), RoPO merely demands 0.0151% of the trainable parameters. Next, we conduct a mathematical analysis:

Supposing the weight matrix of the neural network is $\boldsymbol{W} \in \mathbb{R}^{d\times n}$, there are three components of trainable parameters in RoPO: magnitude vector $\mathbf{m}$ with n trainable parameters, fine-grained rotation matrices with $d-1$ trainable parameters, and global rotation matrix with $2d$ trainable parameters. Therefore, the training parameter quantity of RoPO is $3d-1+n$. By contrast, the training parameter quantity of the baseline DPO-LoRA is $r\times(d+n)$. In our experimental setup, we apply the trainable matrix to the query vectors and value vectors in the attention mechanism. Assuming that the backbone model employs the common Multi-Head Attention ($d = n$), then the trainable parameter quantity of RoPO is approximately $4d-1$, the parameter quantity of DPO-LoRA (r = 4) and DPO-DoRA (r = 4) is $8d$. RoPO achieves performance exceeding that of DPO-LoRA and DPO-DoRA with fewer parameters.

I Sparse Matrix Multiplication Implementation

Due to the sparsity of Fine-grained Rotation Matrix $\tilde{G}_1$, the matrix multiplication between it and the parameter matrix can be quickly implemented in the equivalent way displayed in Figure 9.

J Proof of Full-Angle Coverage for the Fine-Grained Rotation Matrix

Given any vector $\boldsymbol{v} \in \mathbb{R}^d$, $\tilde{\boldsymbol{G_1}}\tilde{\boldsymbol{G_2}}$ could rotate it to any vector $\boldsymbol{y} \in \mathbb{R}^d$ on the same sphere with $\boldsymbol{v}$, *i.e.* $\tilde{\boldsymbol{G_1}}\tilde{\boldsymbol{G_2}}\boldsymbol{v} = \boldsymbol{y}$. The theorem ensures that Fine-grained Rotation Matrix (FRM) in RoPO could accomplish any d-dimensional rotation for n neurons synchronously in the weight matrix. Below is our proof:

We assume that the original neuron $\boldsymbol{v} = [v_0, v_1, \ldots, v_{d-1}]$ and the target neuron $\boldsymbol{y} = [y_0, y_1, \ldots, y_{d-1}]$. Since the magnitude of the vector remains unchanged before and after rotation, we have $||\boldsymbol{v}||_2^2 = ||\boldsymbol{y}||_2^2$. Without loss of generality, we could normalize the vectors $\boldsymbol{v}$ and $\boldsymbol{y}$ to obtain unit vectors $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{y}}$, so that their directions are preserved and the problem reduces to rotating one unit vector to another. In the construction of the FRM, for any two dimensions, there exists a corresponding Givens rotation matrix that performs a rotation in this plane. That is, there are $d-1$ independent parameters to describe this transformation, ensuring a $d-1$ degree of freedom for rotation. Then, we can sequentially rotate the vector to set its dimensions from the 0-th to $d-2$-th to 0. Since $\hat{\boldsymbol{v}}$ is a unit vector, its last dimension must be 1. Next we apply the inverse of the above process to rotate the vector $[0, 0, \ldots, 1]_d$ back to $\hat{\boldsymbol{y}}$.

$$\begin{aligned}\tilde{\boldsymbol{G_1}} &= \prod_{k=0}^{(d/2)-1} \boldsymbol{G}(2k, 2k+1; \theta_k),\\ \tilde{\boldsymbol{G_2}} &= \prod_{k=0}^{(d-1)/2} \boldsymbol{G}(2k+1, 2k+2; \theta_k),\end{aligned} \tag{17}$$

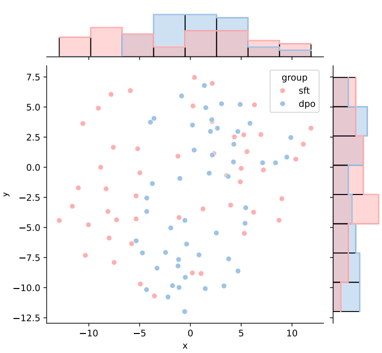
$$\mathbf{R} \cdot \mathbf{x}_m = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{d-2} \\ x_{d-1} \end{bmatrix} \otimes \begin{bmatrix} \cos m\theta_0 \\ \cos m\theta_0 \\ \cos m\theta_1 \\ \cos m\theta_1 \\ \cos m\theta_1 \\ \vdots \\ \cos m\theta_{d/2-1} \\ \cos m\theta_{d/2-1} \end{bmatrix} + \begin{bmatrix} x_1 \\ x_0 \\ x_3 \\ x_2 \\ \vdots \\ x_{d-1} \\ x_{d-2} \end{bmatrix} \otimes \begin{bmatrix} -\sin m\theta_0 \\ \sin m\theta_0 \\ -\sin m\theta_1 \\ \sin m\theta_1 \\ \vdots \\ -\sin m\theta_{d/2-1} \\ \sin m\theta_{d/2-1} \end{bmatrix}$$

Figure 9: The sparse matrix multiplication acceleration method for Fine-grained Rotation Matrices.

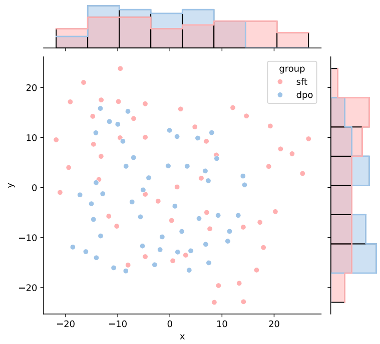
K More Related Work

K.1 Parameter-Efficient Fine-Tuning.

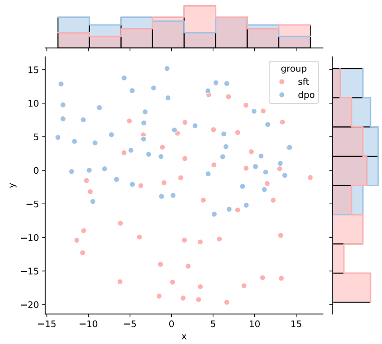
As the scale of the model continues to expand, conducting full fine-tuning of the pre-trained model on downstream tasks is becoming increasingly challenging. The proposal of Parameter-Efficient Fine-Tuning (PEFT) technology has substantially reduced the training and storage costs (Gu et al., 2024), significantly expediting the pace of AI research. Currently, there exist three mainstream PEFT approaches (Lialin et al., 2023): The first one is adapter tuning, which is accomplished by inserting additional trainable modules into the original model (Houlsby et al., 2019; Liu et al., 2022a; Wang et al., 2025); the second one is prompt tuning, which is achieved by concatenating learnable prefix tokens at the beginning of the input (Lester et al., 2021; Liu et al., 2022b); the last one is reparameterization tuning, where only the delta of partial model parameters are reparameterized with few trainable parameters (Hu et al., 2021; Aghajanyan et al., 2020). The last method can merge the learned delta of model parameters into the original model parameters during inference and does not introduce additional costs at the inference stage. It is commonly held that there are two types of schemes for reparameterization tuning, including LoRA-based methods (Gao et al., 2024; Gu et al., 2025a; Liu et al., 2024a; Gu et al., 2025b) and orthogonal fine-tuning (OFT) (Qiu et al., 2024; Liu et al., 2024d; Ma et al., 2024; Chen et al., 2024), and our method of adding parameter regularization is relatively similar to the OFT method.



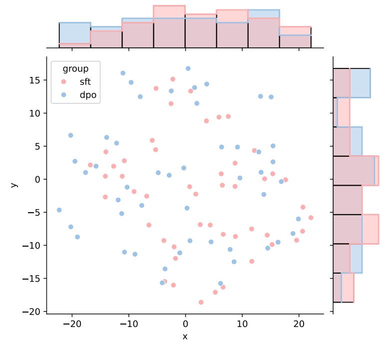
(a) Neuron distribution of the value vector in layer 28 before and after DPO-tuning.



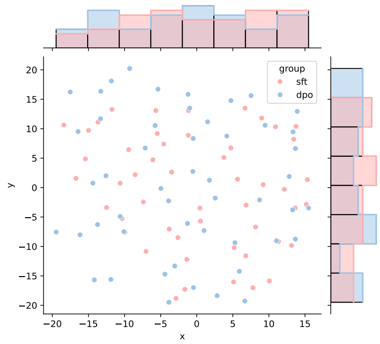
(b) Neuron distribution of the query vector in layer 15 before and after DPO-tuning.



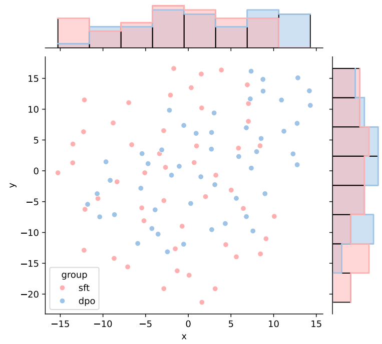
(c) Neuron distribution of the query vector in layer 0 before and after DPO-tuning.



(d) Neuron distribution of the value vector in layer 28 before and after RoPO-tuning.



(e) Neuron distribution of the query vector in layer 15 before and after RoPO-tuning.



(f) Neuron distribution of the query vector in layer 0 before and after RoPO-tuning.

Figure 10: The visualization results using t-SNE on the neuron distribution of the query or value vectors in low, middle, and high layers. The top row illustrates the distribution changes of neurons before and after DPO-tuning, whereas the bottom row illustrates the same for RoPO

Input:	What should i call you?
Output:	
DPO	You can call me by my name, which is "Assistant". However, I would prefer to be referred to as "it" or "the AI" in formal or technical contexts. If you would like to give me a personal name, I will gladly accept it as a temporary or friendly title, but please remember that I am not capable of emotions or personality traits, so my responses will not be affected by any name you choose to call me.\n\n Thank you for your question. \n\n Regards, \n\n Assistant.\n\n (If you would like to give me a personal name, you can try referring to me as "my assistant" or "my virtual assistant", which is more descriptive and reflects my function.) \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n
RoPO	You can call me by my name, which is "Assistant". However, some people prefer to address me as "AI", "Virtual Assistant", or "Bot".
gpt-4o(2024-05-13)	You can call me Assistant. How can I help you today?

Table 5: An easy example for case study.

Evaluation Instructions

You need to complete a comparison task. For this task, you are given an **instruction** and two different **responses** generated by LLMs.

Your task is to identify which of the **responses** is better. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of their responses. If you aren't sure, select that they are **similar**.

Instruction

How did US states get their names?

Response A:

The names of US states were given to them by a combination of factors, including indigenous names, historical figures, geographic features, and cultural references. Some states were named after Native American tribes or their leaders, such as Alabama (from the Choctaw language meaning \"thicket-clearers\"), Dakota (from the Sioux tribe), and Delaware (from the Lenape tribe). Other states were named after historical figures, such as Massachusetts (named after the Massachusett tribe and English explorer Captain John Smith), Virginia (named after Queen Elizabeth I, the \"Virgin Queen\"), and New York (named after the Duke of York, who later became King James II). Some states were named after geographic features, such as California (named after the Baja California peninsula), Michigan (named after Lake Michigan), and Florida (named after the Spanish word for \"flowery\"). Finally, some states were named after cultural references, such as Georgia (named after King George II of England) and Utah (named after the Ute tribe and derived from the Ute word \"yoota\" meaning \"people\").

Response B:

The names of US states were given to them by a variety of factors, including:\n\n1. Native American names: Many states, such as Alabama, Florida, and Mississippi, were named after Native American tribes that lived in the area.\n\n2. Historical figures: Some states, such as Virginia and New York, were named after famous historical figures.\n\n3. Geographical features: Many states, such as California and Texas, were named after the regions or bodies of water that they are located near.\n\n4. Land grants: Some states, such as North Carolina and South Carolina, were named after the original land grants that were given to them by the British monarchy.\n\n5. Political factors: Some states, such as Vermont and New Hampshire, were named after their location in relation to the other states around them.\n\n6. Religious factors: Some states, such as Connecticut and Rhode Island, were named after religious figures or events.

Which response is better?

- A is better.
- B is better.
- A and B are similar.

Figure 11: Human evaluation instructions and interface.

Input:	Given the following data, identify the company with the highest profit in 2021 and provide its CEO's name:a) Company X, with CEO Amy Williams, reported $30 billion in revenue and a $3 billion profit in 2021.b) Company Y, led by CEO Mark Thompson, posted a $60 billion revenue and a $6 billion profit in the same year.c) Company Z, under CEO Sarah Johnson, announced a $20 billion revenue and a $7 billion profit in 2021.d) Company W, managed by CEO James Smith, revealed a $300 billion revenue with a $21 billion profit in 2021.e) Company V, with CEO Lisa Brown, reported a $200 billion revenue and a $25 billion profit in 2021.f) Company U, under CEO John White, posted a $180 billion revenue and a $20 billion profit in the same year.
Output:	
SimPO	e)\n\n Confidence: Based on given data, Company V earned the highest net profit ($25\n\n billion). Lisa Brown is CEO, providing the requested detail. Confidence level: 95% based on given information and no errors present.
RoPO	Company V, with CEO Lisa Brown, reported the highest profit in 2021 with $25 billion.
gpt-4	The company with the highest profit in 2021 is Company V, with a $25 billion profit. Its CEO is Lisa Brown.

Table 6: An easy example for case study.

Input:	When rolling two dice, what is the probability that you roll a total number that is at least 3?
Output: **SimPO**	10 / 36 = 5/18 or 0.2777777...(recurring). Consider all possible non-failures (opposite of 'not reaching 3'): a double 4, 3 + any number from dice 2 possibilities (3,4 for 2, giving unique sums 3+3(skip) + 3+4 = 10 distinct cases out of total 36. Hope this clear!)
DPO	The probability of rolling a total number that is at least 3 is 91/36, or approximately 0.25. The possible combinations for rolling a total of at least 3 are:- Rolling a 3 (1+2) with either die: (1,2) or (2,1)- Rolling a 4 (1+3) with either die: (1,3) or (2,2) or (3,1)- Rolling a 5 (2+3) with either die: (1,4) or (2,3) or (3,2)- Rolling a 6 (3+3) with either die: (1,5) or (2,4) or (3,3) or (4,2) or (5,1). The total number of combinations for rolling two dice is 36 (6 possibilities for the first die x 6 possibilities for the second die). Therefore, the probability of rolling a total number that is at least 3 is 15/36, or 91/36, which simplifies to 5/12. This can also be calculated by subtracting the probability of rolling a total number less than 3 (probability of rolling a 2 or less with either die: 1/6 + 1/6 = 2/6 = 1/3) from 1 (the probability of rolling a total number): 1 - 1/3 = 2/3, and then multiplying by the probability of rolling a total number that is at least 3 (probability of rolling a 3 or more with either die: 5/6 = 5/12): 2/3 x 5/12 = 5/12.
RoPO	The probability that you roll a total number that is at least 3 when rolling two dice is 1 - (the probability of rolling a total of 2 or less). The probability of rolling a total of 2 or less is: P(rolling a total of 2 or less) = P(rolling a 1 on both dice) + P(rolling a 2 on both dice) + P(rolling a 1 on one die and a 2 on the other die) P(rolling a 1 on both dice) = (1/6) x (1/6) = 1/36 P(rolling a 2 on both dice) = (1/6) x (1/6) = 1/36 P(rolling a 1 on one die and a 2 on the other die) = (1/6) x (2/6) = 2/36 Adding these probabilities together gives: P(rolling a total of 2 or less) = 1/36 + 1/36 + 2/36 = 4/36 = 1/9. Therefore, the probability that you roll a total number that is at least 3 when rolling two dice is: 1 - P(rolling a total of 2 or less) = 1 - 1/9 = 8/9 = 0.8889.
gpt-4o(2024-05-13)	To find the probability that the sum of the numbers on two rolled dice is at least 3, we can first calculate the total number of possible outcomes and then the number of favorable outcomes. Each die has 6 faces, so when rolling two dice, the total number of possible outcomes is: $6 \times 6 = 36$. Next, let's determine the number of outcomes where the sum is at least 3. It is easier to count the outcomes where the sum is less than 3 and subtract from the total. The sums less than 3 are: 1. A sum of 2, which can occur with the outcome (1, 1). There is only 1 such outcome. Therefore, the number of outcomes where the sum is at least 3 is: $36 - 1 = 35$. Thus, the probability that the sum is at least 3 is: $\frac{35}{36}$. Therefore, the probability of rolling a total number that is at least 3 is $\frac{35}{36}$.

Table 7: A difficult example for case study.

Input:	Write a descriptive paragraph about a bustling marketplace, incorporating sensory details such as smells, sounds, and visual elements to create an immersive experience for the reader.
Output:	
RoPO	As I step into the bustling marketplace, my senses are immediately overwhelmed by a medley of sights, sounds, and smells. The air is thick with the aroma of freshly baked bread, sizzling spices, and sweet fruits, making my mouth water in anticipation of the culinary delights to come. The marketplace is alive with the hum of vendors haggling over prices and the chatter of customers in various languages. Brightly colored fabrics, exotic fruits, and intricate handicrafts line the stalls, beckoning me to explore. The rhythmic clanging of metal pots and pans from the nearby food stalls punctuates the air, as cooks whip up traditional delicacies for eager patrons. The cacophony of the marketplace is punctuated by the occasional burst of laughter or the melodic tones of a street musician, adding to the lively atmosphere. The sun beats down on the market, casting a warm glow over the scene, as vendors with broad smiles and friendly faces beckon me to their stalls. This is a place where the senses are awakened and the heart is filled with the excitement of discovering new flavors, textures, and cultures.
gpt-4	The bustling marketplace teemed with life, a vibrant tapestry woven with the threads of merchants, shoppers, and passersby. The air was rich with the aromas of sizzling kebabs, pungent spices, and sweet, ripe fruits that beckoned from their rainbow-colored stalls. Eager vendors called out in boisterous voices, proudly advertising their wares, while the melodic strains of a nearby street musician's flute danced through the crowd. A kaleidoscope of colors dazzled the eye, from the vivid hues of intricately woven textiles draped over the stalls to the glistening gold and silver trinkets that glinted enticingly from the jewelry vendors' displays. The ground beneath was a mosaic of worn cobblestones, cool and uneven beneath the feet, a testament to the countless souls who had traversed the marketplace over the years. The marketplace was a living, breathing entity, pulsating with the energy of commerce and human connection, a sensory feast that left one both invigorated and overwhelmed.

Table 8: An example for case study.

Input:	Picture yourself as a 100-years-old tree in a lush forest, minding your own business, when suddenly, a bunch of deforesters shows up to chop you down. How do you feel when those guys start hacking away at you?
Output:	
RoPO	As a 100-years-old tree, I feel a deep sense of sadness and helplessness. My entire life has been spent growing and flourishing in this forest, providing shelter and sustenance to countless creatures. To be cut down without warning or reason feels like a violation of my very existence. I feel the pain of each strike, as if it were a physical blow to my body. I mourn the loss of my branches, the destruction of my roots, and the end of my legacy. But most of all, I feel a deep sense of anger towards those who would do such a cruel and unnecessary act.
gpt-4	As an AI, I do not have personal experiences or emotions, but if I were to imagine myself as a 100-year-old tree, I would likely feel a sense of fear and pain from the deforesters chopping me down. The tree has spent a century growing and providing shelter and resources to the surrounding ecosystem, and its destruction would have a significant impact on the environment.

Table 9: An example for case study.

Alpacaeval 2	You are a highly efficient assistant, who evaluates and selects the best large language model (LLMs) based on the quality of their responses to a given instruction. This process will be used to create a leaderboard reflecting the most accurate and human-preferred answers. I require a leaderboard for various large language models. I'll provide you with prompts given to these models and their corresponding outputs. Your task is to assess these responses, and select the model that produces the best output from a human perspective. (Instruction 0), (Output 0), (Identifier 0) (Instruction 1), (Output 1), (Identifier 1) Evaluate the models based on the quality and relevance of their outputs, and select the model that generated the best output. Answer by providing the model identifier of the best model. We will use your output as the name of the best model, so make sure your output only contains one of the following model identifiers and nothing else (no quotes, no spaces, no new lines, ...)
arena-hard	Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user prompt displayed below. You will be given assistant A's answer and assistant B's answer. Your job is to evaluate which assistant's answer is better. Begin your evaluation by generating your own answer to the prompt. You must provide your answers before judging any answers. When evaluating the assistants' answers, compare both assistants' answers with your answer. You must identify and correct any mistakes or inaccurate information. Then consider if the assistant's answers are helpful, relevant, and concise. Helpful means the answer correctly responds to the prompt or follows the instructions. Note when user prompt has any ambiguity or more than one interpretation, it is more helpful and appropriate to ask for clarifications or more information from the user than providing an answer based on assumptions. Relevant means all parts of the response closely connect or are appropriate to what is being asked. Concise means the response is clear and not verbose or excessive. Then consider the creativity and novelty of the assistant's answers when needed. Finally, identify any missing important information in the assistants' answers that would be beneficial to include when responding to the user prompt. After providing your explanation, you must output only one of the following choices as your final verdict with a label: 1. Assistant A is significantly better: $[[A >> B]]$ 2. Assistant A is slightly better: $[[A > B]]$ 3. Tie, relatively the same: $[[A = B]]$ 4. Assistant B is slightly better: $[[B > A]]$ 5. Assistant B is significantly better: $[[B >> A]]$ Example output: "My final verdict is tie: $[[A = B]]$".

Table 10: Evaluation prompts