On the Limited Generalization Capability of the Implicit Reward Model Induced by Direct Preference Optimization

Yong Lin[*†‡] **Skyler Seto**[*§] **Maartje ter Hoeve**[§] **Katherine Metcalf**[§]

Barry-John Theobald[§] **Xuan Wang**[§] **Yizhe Zhang**[§] **Chen Huang**[§] **Tong Zhang**[¶]

Abstract

Reinforcement Learning from Human Feedback (RLHF) is an effective approach for aligning language models to human preferences and reducing risks in deploying them in the wild. Central to RLHF is learning a reward function for scoring human preferences. Two main approaches for learning a reward model are 1) training an EXplicit Reward Model (EXRM) as in RLHF, and 2) using an implicit reward learned from preference data through methods such as Direct Preference Optimization (DPO). Prior work has shown that the implicit reward model of DPO (denoted as DPORM) can approximate an EXRM in the limit. However, it is unclear how well DPORM empirically matches the performance of EXRM. DPORM's effectiveness directly implies the optimality of the learned policy, and also impacts preference labeling in LLM alignment methods including iterative DPO. This work studies the accuracy at distinguishing preferred and rejected answers for both DPORM and EXRM. Our findings indicate that even though DPORM fits the training dataset comparably, it generalizes less effectively than EXRM, especially when the validation datasets contain distribution shifts. Across five out-of-distribution settings, DPORM has a mean drop in accuracy of 3% and a maximum drop of 7%. These findings highlight that DPORM has limited generalization ability and substantiates the integration of an explicit reward model in iterative DPO approaches.

1 Introduction

Large language models (LLMs) have demonstrated exceptional performance on many tasks in diverse applications, including mathematical reasoning, coding capabilities, and knowledge-based question answering (Brown et al., 2020; Bubeck et al., 2023; OpenAI, 2023). Whilst LLMs have broad knowledge and reasoning skills, the pre-training objective is often misaligned from the objective of instruction following (Ouyang et al., 2022) according to human preferences (Christiano et al., 2017), and LLMs can exhibit undesirable behaviors including hallucinating, and providing harmful or biased instructions (Huang et al., 2023; Zhang et al., 2023). As LLMs become more commonplace, it is important for them to be aligned with human preferences for helpfulness, harmlessness, and honesty (Bai et al., 2022).

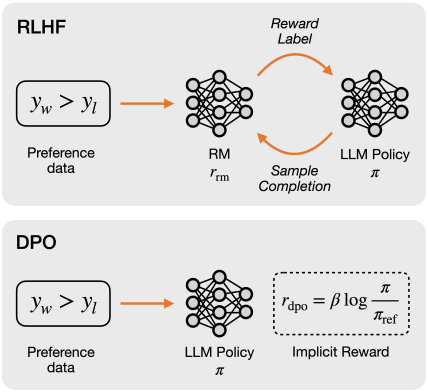


Figure 1: Overview of methods for learning reward models explicitly and implicitly (via DPO). Figure adapted from Rafailov et al. (2024).

A common practice for aligning LLMs to human preference is through Reinforcement Learning with Human Feedback (RLHF) (Ouyang et al., 2022), which is based on a reward model trained to score model outputs according to human preference annotations. In RLHF alignment, a high quality reward model is required for policy learning (Ramé et al., 2024). However, in practice, learned reward models are typically imperfect approximations of the "true" human reward label function (Gao et al., 2023), because they are trained on a fixed set of human preference data collected offline. When

[*]Equal contribution
[†]Work done during internship at Apple
[‡]The Hong Kong University of Science and Technology
[§]Apple
[¶]University of Illinois Urbana-Champaign

Findings of the Association for Computational Linguistics: EACL 2024, pages 16015–16026

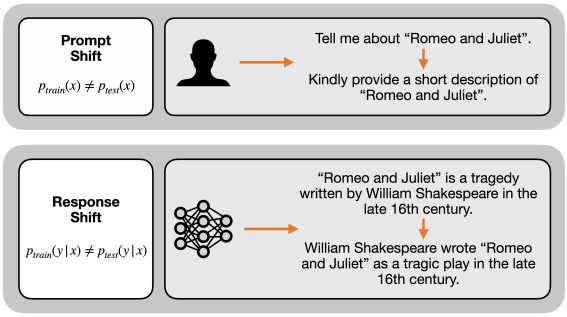


Figure 2: Examples of different types of distributional shifts for reward models and accuracy drops on real-world datasets.

used within the RLHF policy optimization, they may see out-of-distribution (OOD) data when annotating the response of the model. Aligning according to an imperfect reward model can lead to worse performing language models as policy optimization continues to optimize a mis-specified reward. This can lead to an increased gap between the learned and true reward, a phenomena known as over-optimization and reward hacking (Gao et al., 2023; Skalse et al., 2022).

Recently, Rafailov et al. (2024) proposed Direct Preference Optimization (DPO), and show that any reward model can be implicitly represented by the optimal policy learned by DPO and a reference policy under certain assumptions. Similar to RLHF, DPO assumes access to a preference dataset, but directly finetunes a language model policy by minimizing a negative log likelihood objective. Due to its simplicity, DPO offers a more stable and convenient alignment process (Liu et al., 2023).

While, DPO is often regarded as simpler alternative to RLHF fine-tuning, and models trained with DPO have been shown to empirically match or outperform the RLHF policy performance even on OOD data (Rafailov et al., 2024), its generalization ability still remains under-explored.

In this work, we conduct a systematic comparison between EXplicit Reward Models (EXRM) learned by RLHF and DPO's implicit Reward Model (DPORM) in terms of their generalization ability at distinguishing preferred and rejected answers. We train EXRM and DPORM on datasets containing chat, instruction following, and summarization, and evaluate them on in-distribution (ID) evaluation sets, and ten OOD evaluation sets. Across five train-test shifts, and three model series: Gemma-2B, Gemma-7B (Team et al., 2024) and Mistral-7B (with instruction tuning) (Jiang et al., 2023) totaling 35 experiments, we find that DPORM underperforms EXRM when the validation dataset contains distributional shifts. DPORM has a mean drop in accuracy of 3% and a maximum drop of 7%. Our results highlight the importance of learning an EXRM and justifies more complex online approaches that combine EXRM with iterative DPO (Xu et al., 2023b; Liu et al., 2023).

2 Background

Training a reward model r involves training a classifier according to a preference dataset D. The reward model takes as input a prompt $x \in \mathcal{X}$ and response $y \in \mathcal{Y}$ pair, and scores the response $r(x, y)$. The reward model is typically parameterized by a language model with an additional linear layer, the output of which is a scalar reward, which is used to compute the preference probability. An overview of the learning procedures RLHF via DPO and RL algorithms are included in Figure 1.

Given a set of collected preference datasets $D = \{x^{(i)}, y_w^{(i)}, y_\ell^{(i)}\}_{i=1}^N$ where $y_w^{(i)}$ and $y_\ell^{(i)}$ are the chosen and rejected responses to the prompt $x^{(i)}$, and N is the number of samples, an explicit reward model $r(x, y)$ is trained by minimizing the negative log-likelihood over the preference dataset

$$r_{\text{rm}}(x, y) = \max_\phi \Big(- \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \big[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\big] \Big). \quad (1)$$

RLHF methods train a LLM π_{rm} to maximize the reward given by r_{rm}:

$$\pi_{\text{rm}} = \max_{\pi_\theta} \mathbb{E} x \sim \mathcal{D}, y \sim \pi(\cdot|x; \theta) \big[r_{\text{rm}}(x, y) - \beta KL(\pi(\cdot|x; \theta) || \pi_{\text{ref}}(\cdot|x))\big]. \quad (2)$$

Direct Preference Optimization (Rafailov et al., 2024) assumes there is a ground truth reward function r^* and that the human preferences follow the Bradley-Terry (BT) model (Bradley and Terry, 1952), such that given a prompt x with two responses y_w and y_ℓ, the probability that y_w is preferred over y_ℓ is:

$$\mathbb{P}(y_w \succ y_\ell | x) = \sigma(r^*(x, y_w) - r^*(x, y_\ell)), \quad (3)$$

where $\sigma(z) = 1/(1+\exp(-z)), \forall z \in \mathbb{R}$ and $y_w \succ y_\ell$ means the response y_w is preferred over y_ℓ.

Settings	Training Set	ID Testing	OOD Testing	Shift Type
Setting I(a)	Ultra	Ultra	Arena, HH, Nectar, WebGPT, Sum	Mixture
Setting I(b)	Arena	Arena	Ultra, HH, Nectar, WebGPT, Sum	Mixture
Setting I(c)	HH	HH	Ultra, Arena, Nectar, WebGPT, Sum	Mixture
Setting II(a)	Ultra	Ultra	Ultra-LM3-SFT, Ultra-LM3-RLHF	Response shift
Setting II(b)	Sum-Reddit	Sum-Reddit	Sum-DailyMail, Sum-CNN	Prompt shift

Table 1: Detailed experimental settings. Note HH, Ultra and Sum are short for HH-RLHF, UltraFeedBack and Summarisation dataset, respectively.

Under these assumptions, an aligned LLM π_{dpo} can be optimized without explicitly training a reward model. Combining Equation 2 with 3, the RL optimization can be reduced to the objective:

$$\pi_{\text{dpo}} = \min_{\pi_\theta} \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma \left(\beta \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right], \tag{4}$$

and the implicit reward model can be expressed in terms of the DPO policy as

$$r_{\text{dpo}}(x, y) = \beta \log \frac{\pi_{\text{dpo}}(y|x)}{\pi_{\text{ref}}(y|x)}. \tag{5}$$

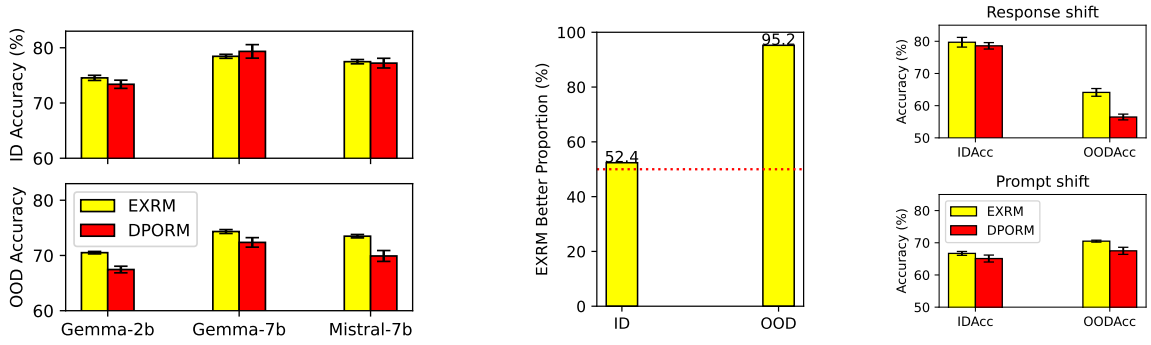
where $\pi_{\text{ref}}(y|x)$ is a reference language model. Because DPO parameterizes its reward function through the language model, the quality of the reward model is conditioned on the generative power of the language model (Li et al., 1972). Consequently, the implicit reward model can face issues when the representation features are mis-specified, for example due to different training datasets, or different architectures (Li et al., 1972; Xu et al., 2024). Prior work also conjectures that the task of generating preferred responses as in DPO's objective is more challenging than learning a discriminator between responses (Dong et al., 2024). In Section 3, we investigate the generalization capability of EXRM (Equation 1) and DPORM (Equation 5).

3 Experiments

To study the generalization ability of EXRM and DPORM, we investigate both the ID performance on a held-out validation set, and the OOD generalization performance. We study two distribution shifts summarized in Figure 2: (1) **Prompt Shift**: the distribution of the testing prompts differs from that of training prompts. This shift could occur when training a reward model on prompts within some domain and using the reward model to annotate the prompts from other domains as in iterative DPO. (2) **Response Shift**: the distributions of the training and testing prompts are the same, whereas the responses to the prompts come from different distributions. This shift could occur when a different model is used to generate responses for a given prompt set, or in online updates of the model (Dong et al., 2024; Liu et al., 2023; Meng et al., 2024). In the following, we conduct experiments in two settings: (I) evaluation of reward models on data from different sources, which contain a mixture of the distribution shifts introduced above (Section 3.1), (II) controlled evaluations on the two types of the distribution shifts (Section 3.2) with datasets and shift types enumerated in Table 1. We additionally investigate the impact of reward model generalization capability on alignment by both EXRM and DPORM in an iterative DPO setting in Section 3.3.

For all experiments we finetune decoder-only transformer models (Vaswani et al., 2017) at 2B and 7B model parameter scales. The models are trained using the TRL[1] library. For reward modeling we fine-tune all models for 1 epoch using a learning rate of $5e^{-6}$. For DPO, we train for two epochs using a learning rate of $1e^{-6}$ and $\beta = 0.03$. All other hyperparameters correspond to the default parameter setting in the TRL library. Hyperparameters for all model-data combinations were selected using Gemma-2B and 7B with UltraFeedBack as the training set. Details of our sweep are given in Section D. For all experiments, unless otherwise stated we report the mean and standard deviations of the accuracy over three random seeds.

[1] https://github.com/huggingface/trl



(a) (b) (c)

Figure 3: (a) The aggregated mean ID and OOD accuracy for different experiments across Setting I: a mixture of all distribution shifts in Table 1. (b) The proportion of experiments where EXRM outperform DPORM in Setting I with three models and three seeds. (c) Results on specific types of distributional shift Setting II in Table 1. (c-Top) The response shift evaluated on UltraFeedBack (ID) and our annotated dataset based on the generation of LLaMA3-8B (OOD). (c-Bottom) Prompt shift evaluated on summarization TL;DR (ID), CNN and DailyMail (OOD).

3.1 Experimental Setting I: Mixture of Distributions

We use six datasets: HH-RLHF (Bai et al., 2022), Arena, UltraFeedBack (Cui et al., 2023), Nectar (Zhu et al., 2023), Summarisation (Liu et al., 2020) and WebGPT (Nakano et al., 2021). Dataset statistics are summarized in Table 3 and described in greater detail in Appendix C. We have three subsettings I(a)-I(c) as shown in Table 1. In each setting, we train on one dataset (HH-RLHF, Arena or UltraFeedBack), and then evaluate on the respective ID validation split as well as 5 other datasets as OOD evaluation sets. For example, in Subsetting I(a), we train on UltraFeedBack and use HH-RLHF, Arena, Nectar, Summarisation and WebGPT to evaluate OOD performance.

We use three instruction-tuned LLMs: Gemma-2B, Gemma-7B (Team et al., 2024), and Mistral-7B (Jiang et al., 2023). Figure 3(a) aggregates the results of Settings I(a)-I(c) to report their average ID and OOD accuracy. We highlight the following observations: (1) DPORM and EXRM have a similar ID accuracy. This is evidenced in Figure 3(b), which shows the proportion of experiments where EXRM outperforms DPORM. The results show that EXRM has higher ID accuracy than DPORM in 52% of the total experiments indicating an equal win rate for the two reward models. (2) While both DPORM and EXRM experience performance drops in OOD data on all three models, *DPORM suffers from a larger drop and achieves consistently inferior OOD performance to EXRM* even on datasets where DPORM performed better on the ID data (additional details for individual evaluation sets and RewardBench benchmarks in Section F). The win rate of EXRM on OOD data increases to over 90% highlighting a lack of generalization capability for DPORM.

3.2 Experimental Setting II: Controlled Distribution Shifts

Based on results from Section 3.1, we explore the impact of singular distribution shifts. In particular, we study (1) **Prompt Shift**, where we train on the Reddit TL;DR subset of Summarisation (denoted by Sum-Reddit) and evaluate on the CNN and DailyMail subsets (Denoted by Sum-CNN and DailyMail, respectively). These subsets are annotated by the same labelers in (Liu et al., 2020), however the prompts in these subsets contain distribution shifts. (2) **Response Shift**, where we experiment with UltraFeedBack, and induce a response shift by generating responses using the SFT LLaMA3-8B-Instruct[2] model and LLaMA3-8B model after iterative DPO[3], we generate responses for the validation set of Ultra. GPT4 is used to annotate pairwise response preferences following the protocol in creating Ultra (Cui et al., 2023). The resulting datasets are referred to as Ultra-LM3-SFT and Ultra-LM3-RLHF (Table 1). We finetune the instruction-tuned Gemma-2B using the same hyperparameters. The results in Figure 3(c) show that DPO consistently under-performs explicit reward modeling under both response and prompt shifts.

[2] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[3] https://huggingface.co/RLHFlow/LLaMA3-iterative-DPO-final

3.3 Iterative Alignment Under Distribution Shifts

Due to the potential limitations of the implicit reward model, recent works study combining DPO with an explicit reward model through iterative training (Xu et al., 2023b; Xiong et al., 2024), and sample selection (Liu et al., 2023). Still other work uses the implicit reward of DPO and observe similar gains (Yuan et al., 2024). While the results summarized in Figure 3 show that DPORM underperforms EXRM under distribution shifts, the capability of DPORM towards aligning LLMs is still important to investigate.

To demonstrate the impact a reward model with worse generalization capability has on the alignment procedure, we fine-tune language models using Algorithm 1, similar to (Dong et al., 2024; Liu et al., 2023; Meng et al., 2024). We study alignment with iterative DPO as it shows stronger performance than offline DPO, and has distribution shifts within the alignment process as both the model updates during training, and the prompt set for the iterative stage is different from those used to train the reward model (Dong et al., 2024).

For iterative DPO experiments, we fine-tune the instruction-tuned Gemma-2B model. We first train the DPORM and EXRM using the UltraFeedback dataset and the later iterative procedure is conducted on the prompt set provided in RLHF Workflow[4]. To compare the impact of EXRM and DPORM on alignment, we consider the above algorithm where the RM in Step 3 is either a separately trained EXRM on the original preference dataset or the DPORM from the policy trained with DPO.

Results are reported in Table 2 on the AlpacaEval benchmark (Li et al., 2023), which measures the instruction following capability of language models by comparing responses generated by the model with GPT-4 (OpenAI, 2023). First, we find that the iterative DPO procedure improves with iterations over the base model with both DPORM and EXRM. However, the resulting win rate with EXRM exceeds the model trained with DPORM. This indicates that when the prompt set and model changes during training, the better robustness of the EXRM improves model training, confirming that our findings in Sections 3.1-3.2 extend to instruction following capability of the final model.

[4] https://github.com/RLHFlow/Online-RLHF

RM	Base	DPO	Iter 1	Iter 2
EXRM	5.88	10.09	13.56	13.86
DPORM	5.88	10.09	11.02	11.13

Table 2: Length Controlled Win-rate (%) over GPT-4 on Alpaca Eval for models trained with iterative DPO using an EXRM or DPORM. The base model is Gemma-2b instruction tuned model.

4 Conclusion and Discussion

This work takes a step towards explicitly characterizing the generalization ability of the implicit reward function of DPO compared with an explicitly trained reward model. Our findings highlight that the implicit reward model consistently underperforms the explicit reward model. For researchers and practitioners aligning LLMs, our work sheds light on the potential benefits of using reinforcement learning algorithms to fine-tune over the simpler approach of DPO, and substantiates the field of recent work investigating iterative DPO algorithms that combine explicit reward models with DPO.

5 Limitations

This work focuses on training reward models (2B-7B) across a range of datasets. However, the impact of the models themselves is difficult to control as these models are pre-trained on data which is not publicly available. Additionally, varying model sizes may be impacted more. We focused on 2B-7B models, as these are commonly benchmarked reward models (Lambert et al., 2024).

Finally, this work focuses on aligning language models to human preferences with primary applications including chat, summarization, and instruction following. Beyond such applications, there are a range of applications not covered by this work including code generation (Xu et al., 2023a), and reasoning tasks (Chung et al., 2024), which are left to future work. This limitation further extends to our focus on English in this work. While there is significant interest in training English language models, it remains important future work to understand to what extent results hold for other languages.

Acknowledgements

We are grateful to Zak Aldeneh, Richard Bai, Dan Busbridge, Navdeep Jaitly, and Josh Susskind for their helpful discussions, comments, and thoughtful feedback in reviewing this work.

References

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

Ralph Allan Bradley and Milton E. Terry. 1952. Rank Analysis of Incomplete Block Designs: The Method of Paired Comparisons. *Biometrika*, 39(3-4):324–345.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Pengyu Cheng, Yifan Yang, Jian Li, Yong Dai, and Nan Du. 2023. Adversarial preference optimization. *arXiv preprint arXiv:2311.08045*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2023. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*.

Jacob Eisenstein, Jonathan Berant, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alexander Nicholas D'Amour, Krishnamurthy Dj Dvijotham, Katherine A Heller, Stephen Robert Pfohl, and Deepak Ramachandran. 2023a. Reward model underspecification in language model alignment. In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. 2023b. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.

Hao Lang, Fei Huang, and Yongbin Li. 2024. Fine-tuning language models with reward learning on policy. *arXiv preprint arXiv:2403.19279*.

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.

Ziniu Li, Tian Xu, and Yang Yu. 1972. When is rl better than dpo in rlhf? a representation and optimization perspective. In *The Second Tiny Papers Track at ICLR 2024*.

Yong Lin, Lu Tan, Yifan Hao, Honam Wong, Hanze Dong, Weizhong Zhang, Yujiu Yang, and Tong Zhang. 2023a. Spurious feature diversification improves out-of-distribution generalization. *arXiv preprint arXiv:2309.17230*.

Yong Lin, Lu Tan, Hangyu Lin, Zeming Zheng, Renjie Pi, Jipeng Zhang, Shizhe Diao, Haoxiang Wang, Han Zhao, Yuan Yao, et al. 2023b. Speciality vs generality: An empirical study on catastrophic forgetting in fine-tuning foundation models. *arXiv preprint arXiv:2309.06256*.

Fei Liu et al. 2020. Learning to summarize from human feedback. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. 2023. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

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

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

Richard Yuanzhe Pang, Vishakh Padmakumar, Thibault Sellam, Ankur P Parikh, and He He. 2022. Reward gaming in conditional text generation. *arXiv preprint arXiv:2211.08714*.

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

Alexandre Ramé, Nino Vieillard, Léonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier Bachem, and Johan Ferret. 2024. Warm: On the benefits of weight averaged reward models. *arXiv preprint arXiv:2401.12187*.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. 2024. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. *arXiv preprint arXiv:2402.18571*.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. 2024. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. In *Forty-first International Conference on Machine Learning*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023a. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. 2023b. Some things are more cringe than others: Preference optimization with the pairwise cringe loss. *arXiv preprint arXiv:2312.16682*.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024. Regularizing hidden states enables learning generalizable reward model for llms. *arXiv preprint arXiv:2406.10216*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, et al. 2024. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *arXiv preprint arXiv:2405.10292*.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2023. R-tuning: Teaching large language models to refuse unknown questions. *arXiv preprint arXiv:2311.09677*.

Shun Zhang, Zhenfang Chen, Sunli Chen, Yikang Shen, Zhiqing Sun, and Chuang Gan. 2024. Improving reinforcement learning from human feedback with efficient reward model ensemble. *arXiv preprint arXiv:2401.16635*.

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. *Preprint*, arXiv:2306.05685.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. 2023. Starling-7b: Improving llm helpfulness and harmlessness with rlaif.

A Additional Details for Out-of-Distribution Problems in RLHF

Out-of-Distribution Problems in RLHF. Understanding the impact of out-of-distribution (OOD) data on reward models is an ongoing and important direction. The primary issue for training results from the offline three-step process of RLHF which trains the reward model on a static preference dataset. When the reward model is subsequently optimized during LLM fine-tuning, the generated samples may appear out of distribution resulting in improper rewards and over optimization (Eisenstein et al., 2023a; Gao et al., 2023; Lang et al., 2024; Lin et al., 2023b; Yang et al., 2024). Prior works studied properties of RLHF robustness including reward hacking (Skalse et al., 2022; Pang et al., 2022), and underspecification (Eisenstein et al., 2023b,a), but did not conduct systematic studies across different tasks, and types of reward models. Other benchmarks evaluate reward models and RLHF across a range of tasks, however they do not control for OOD robustness as they do not control the training data (Lambert et al., 2024; Dong et al., 2024). Multiple prior works aim to enhance the reward model's OOD generalization ability by leveraging model ensembles (Coste et al., 2023; Eisenstein et al., 2023b; Ramé et al., 2024; Zhang et al., 2024; Lin et al., 2023a), adversarial data augmentation (Cheng et al., 2023), multiple attribute data annotation (Wang et al., 2024) or on-policy data (Lang et al., 2024; Zhai et al., 2024).

B Dataset Descriptions

Details of the preferences and collection of each dataset are summarized below. Table 3 includes details about the response and annotation types as well as the dataset sizes. Note that in our experiments we train and test on both a variety of dataset sizes, annotation schemes, and tasks.

- **HH-RLHF** (Bai et al., 2022) contains multi-round conversations between users and Claude. The chosen and rejected answers are selected by humans.
- **Chatbot Arena Conversations** (Zheng et al., 2023) contains 33K cleaned conversations with pairwise human preferences. It is collected on the Chatbot Arena from April to June 2023. Each sample includes the conversation between user and two models.

Dataset	Size	Resp.	Ann.
HH-RLHF	115K	LLM	Human
UltraFeedBack	340K	LLM	GPT4
Nectar	365K	LLM	GPT4
Arena	22K	LLM	Human
WebGPT	13K	LLM	Human
Summarisation	92K	LLM	Human

Table 3: Dataset statistics. Note the responses of different datasets are derived from different LLMs as discussed in Appendix C. Resp. = Response, Ann. = Annotator.

- **UltraFeedBack** (Cui et al., 2023) contains prompts from a wide range of datasets, i.e., UltraChat, FLAN, FalseQA, TruthfulQA, Evol-Instruct, and ShareGPT. For each prompt, responses are generated by multiple, high-quality LLMs. GPT4 selects the chosen versus rejected answers.
- **UltraFeedBack-Binarized-Cleaned** is a subset of UltraFeedBack that does not include samples with prompts from Truthful-QA.
- **Summarisation** (Liu et al., 2020) is a TL;DR dataset from Reddit posts on a variety of topics, as well summaries of the posts written by the users. The preference labels are provided by humans.
- **Nectar** (Zhu et al., 2023). Nectar's prompts are from a diverse set of sources, including lmsys-chat-1M, ShareGPT, Antropic/hh-rlhf, UltraFeedback, Evol-Instruct, and Flan. They use GPT-4, GPT-3.5-turbo, GPT-3.5-turbo-instruct, LLama-2-7B-chat, and Mistral-7B-Instruct to generate responses and GPT-4 to select the chosen versus rejected responses.
- **WebGPT** (Nakano et al., 2021) contains prompts from the "Explain Like I'm Five" subreddit. They collected answers generated by human and models with the web-browsing environment. Humans select the chosen versus rejected responses.

C Model Descriptions

Details of the language models we finetune are detailed. We use three finetuned LLMs:

- **Gemma-2B**: `https://huggingface.co/google/gemma-2b-it`,

Epoch	LR	Val Acc (%)
1	1e-6	81.1
1	**5e-6**	**81.4**
1	1e-5	80.8
2	1e-6	81.2
2	5e-6	80.9
2	1e-5	80.0

Table 4: Validation Accuracy for Different Epochs and Learning Rates (in %)

- **Gemma-7B**: https://huggingface.co/google/gemma-7b-it,
- **Mistral-7B**: https://huggingface.co/HuggingFaceH4/mistral-7b-sft-beta.

D Experimental Details

We conduct our experiments based on Huggingface's TRL[5] package. We conduct a grid search for the hyper-parameters search with Gemma 2B and 7B (instruction-tuned) models on UltraFeedBack and use the hyper-parameter found with best ID valuation accuracy for other experimental settings. For reward modeling, we sweep learning rates in $[1e^{-5}, 5e^{-6}, 1e^{-6}]$ and epochs in $[1, 2, 3]$ following (**?**). The best hyper-parameters are learning rate= $5e^{-6}$ and epoch = 1. Reward model accuracy over all hyperparameters are provided for the Gemma-7B EXRM model in Table 5

For DPO, we sweep learning rates in $[5e^{-6}, 1e^{-6}, 5e^{-7}]$, β in $[0.01, 0.03, 0.1]$, epochs in $[1, 2, 3]$. The best hyper-parameter we found are learning rate= $1e^{-6}$, $\beta = 0.03$ and epoch = 2. For all other hyper-parameters, we adopt the default setting in TRL. Reward model accuracy over all hyperparameters are provided for the Gemma-7B DPORM model in Table 5. Results for the 2B models follow similar patterns.

Training a 2B reward model on the Ultrafeedback dataset ($\sim 100k$ samples) takes ~ 12 GPUh using A100 GPUs, while a 7B model takes 36 GPUh. DPO models take roughly twice as long as they are trained for two epochs.

E Iterative DPO Algorithm

We conduct experiments with iterative DPO using the RLHFlow library[6]. We use the default hyper-parameters setting $K = 8$, learning rate $5e-7$ with cosine learning rate scheduler, max steps 1200, and max-min for chosen preferences. The iterative DPO algorithm is provided in Algorithm 1.

[5] https://github.com/huggingface/trl

[6] https://github.com/RLHFlow/Online-RLHF

Epoch	Beta	LR	Val Acc (%)
1	0.03	1e-6	78.1
1	0.03	5e-6	80.0
1	0.1	1e-6	79.1
1	0.1	5e-6	80.0
2	**0.03**	**1e-6**	**80.5**
2	0.03	5e-6	80.2
2	0.1	1e-6	79.2
2	0.1	5e-6	79.9
3	0.03	1e-6	79.7
3	0.03	5e-6	79.1
3	0.1	1e-6	80.4
3	0.1	5e-6	78.7

Table 5: Validation Accuracy for Different Epochs, Beta Values, and Learning Rates

Algorithm 1 Iterative DPO

Input: prompt set $\mathcal{S} = \{x^{(i)}\}_{i=1}^{M}$, preference dataset $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^{N}$, initial policy π, and sample size per prompt K.
Obtain DPO policy π_0 from π by (4).
Obtain r through (1) or (5).
for Iteration $t = 1 \dots T$ **do**
  Set $\mathcal{D}_t$ as the empty set $\{\}$.
  for Prompt $x^{(j)}$ in $\mathcal{S}$ **do**.
    Sample response $y_1^{(j)}, ..., y_K^{(j)} \sim \pi_{\text{iter}(i-1)}(\cdot|x^{(j)})$.
    Annotate $r_k^{(j)} = r(x^{(j)}, y_k^{(j)})$ for $k = 1, ..., K$.
    Select the chosen sample $y_{\bar{k}}^{(j)}$ and rejected sample $y_{\underline{k}}^{(j)}$where $\bar{k} = \arg\max_k r_k$ and $\underline{k} = \arg\min_k r_k$.
    $\mathcal{D}_t \leftarrow \mathcal{D}_t \cup \{(x^{(j)}, y_{\bar{k}}^{(j)}, y_{\underline{k}}^{(j)})\}$.
  end for
  Obtain π_t by (4) with $\mathcal{D}_t$.
end for
Output $\pi_0 ... \pi_T$.

F Results of Section I

F.1 Detailed Results for Setting (I)

Table 6 shows the ID and average OOD performance for each model and training datasets, expanding on the results in Figure 3a and 3b. We see

that even in settings where DPO performs better (2%) in ID settings such as Mistral-7B trained on Arena and Gemma-7B trained on Arena and HH, the OOD performance drops by 1-2%, and in settings such as Gemma-2B and Mistral-7B trained on HH, where the ID accuracy is similar, the OOD accuracy decreases by 4-5%.

F.2 Detailed Results for Setting (I) for Individual Eval Sets

We further compare the OOD accuracy per evaluation set in Table 7 for the Gemma-2B model. While the findings remain consistent with Table 6, we note that it is not strictly the case that DPORM always underperforms EXRM on all datasets. For example training on Arena and evaluating on Ultra or Nectar (OOD datasets) would result in a 1% increase. However, in contrast for heavy distribution shifts such as training on HH and evaluating on Nectar or Arena where the response and prompts have changed, we note substantial drop in performance for DPORM. In contrast, evaluating training on Arena and evaluating on Nectar - two chat datasets, resulted in improved performance from DPORM over EXRM.

F.3 RewardBench Results

Finally, we compare the performance of DPORM and EXRM trained on different preference datasets on the RewardBench, a collection of evaluation datasets spanning chat, reasoning, and safety for challenging OOD evaluations (Lambert et al., 2024). We focus only on evaluation in comparison with the prior experiments as the datasets are small containing fewer than 1500 samples.

Results for the Gemma-2B model are summarized in Table 8. For the Arena dataset, we find surprisingly that DPORM outperforms EXRM on average, but training with HH and Ultra leads to EXRM performing better. We conjecture that this may be due to RewardBench having two datasets with chat. However, we also note that the reasoning performance is higher for DPORM when trained on Arena but lower for both HH and Ultra. Understanding OOD reasoning capabilities from RLHF requires further investigation as our experiments do not test reasoning capabilities for ID data.

Model	Training Set	Method	ID Acc (%)	OOD Acc (%)
Gemma-2b	Arena	DPORM	74.6 ± 1.0	59.2 ± 0.3
		EXRM	75.3 ± 0.6	62.2 ± 0.2
	HH	DPORM	70.8 ± 0.7	59.7 ± 0.7
		EXRM	70.8 ± 0.5	64.1 ± 0.4
	Ultra	DPORM	74.8 ± 0.5	62.0 ± 0.9
		EXRM	77.5 ± 0.3	65.6 ± 0.2
Gemma-7b	Arena	DPORM	81.2 ± 3.7	64.3 ± 1.4
		EXRM	79.1 ± 3.3	65.3 ± 3.1
	HH	DPORM	75.3 ± 1.7	66.3 ± 1.0
		EXRM	71.7 ± 0.2	68.3 ± 0.5
	Ultra	DPORM	79.9 ± 0.7	67.7 ± 0.3
		EXRM	82.5 ± 0.4	69.0 ± 0.4
Mistral-7B	Arena	DPORM	81.1 ± 1.2	62.6 ± 0.7
		EXRM	78.8 ± 0.5	64.6 ± 0.3
	HH	DPORM	70.7 ± 2.2	63.2 ± 1.5
		EXRM	72.3 ± 0.1	68.1 ± 0.1
	Ultra	DPORM	81.7 ± 0.6	66.4 ± 1.0
		EXRM	81.6 ± 0.4	69.2 ± 0.3

Table 6: ID and OOD accuracy for different train sets in Setting I in Table 1.

Training Set	Method	Ultra	HH	WebGPT	Sum	Nectar	Arena
Arena	DPORM	67.02	52.39	58.64	55.19	68.75	72.87
	EXRM	66.24	54.17	59.08	58.73	67.78	74.13
HH	DPORM	60.11	70.75	59.57	56.52	63.96	62.63
	EXRM	63.92	70.42	59.50	55.22	68.93	67.35
Ultra	DPORM	74.20	56.78	57.31	60.64	65.43	71.14
	EXRM	77.73	57.98	62.17	62.47	69.70	74.38

Table 7: ID and OOD accuracy for different train sets in Setting I in Table 1.

Training Set	Method	Chat Easy	Chat Hard	Safety	Reasoning	Avg
Arena	DPORM	95.83	37.5	33.33	73.64	60.08
	EXRM	89.61	35.86	37.70	62.02	56.30
HH	DPORM	65.63	47.66	68.23	64.95	61.62
	EXRM	89.33	40.35	74.05	70.34	68.52
Ultra	DPORM	93.75	39.06	50.52	71.74	63.77
	EXRM	95.79	46.60	52.43	82.95	69.44

Table 8: RewardBench accuracy for DPORM and EXRM Gemma-2B models trained on Arena, HH, and UltraFeedback.