CONECUT: Scalable Removal of Preference Redundancy

Purbid Bambroo, Daniel S. Brown, and Ana Marasović
University of Utah
Kahlert School of Computing
{purbid.bambroo,daniel.s.brown,ana.marasovic}@utah.edu

Abstract

Reward models are central to post-training alignment of large language models (LLMs) via human preferences. As reward benchmarks gain prominence, it becomes critical to evaluate their integrity. A key challenge that remains underexplored in this space is the identification of redundant examples in these evaluation datasets. These are preference pairs that enforce near-duplicate or redundant half-space constraints on the reward-model weight vector and hence may inflate or exaggerate the perceived alignment of a reward model. In this work, we propose a novel method, CONECUT, to identify redundancy in preference alignment datasets by formulating this task as a cone membership test over a reward model's hidden representations. Our experiments on RewardBench reveal that a substantial fraction of examples in evaluation pairs are near redundant, and pruning them results in measurable performance drops across multiple reward models. Our work highlights the overestimation of alignment that evaluation datasets might cause in socially critical areas like refusals and safety. We advocate for redundancy-aware evaluation as a step toward better model alignment and curating socially responsible evaluation datasets.

1 Introduction

Preference alignment (Ziegler et al., 2020; Bai et al., 2022b; Liu et al., 2024b) is a critical final stage when training modern large language models (LLMs) to avoid toxic or unsafe content generation (OpenAI, 2023; Anthropic, 2023; Touvron et al., 2023, among others). Central to this alignment process are reward models that learn to score model outputs according to human preferences and steer LLMs toward responses that are both helpful and harmless.

However, reward models are imperfect. They frequently overfit to superficial shortcuts that enable reward hacking (Tien et al., 2023), become biased toward the specific preferences of their human annotators (Bai et al., 2022b; Casper et al., 2023), and exhibit excessive caution that leads to unnecessary refusal behaviors (Bai et al., 2022b; Dabas et al., 2025). Given these limitations and the critical role reward models play in ensuring safe LLM deployment, it is important to develop robust methods to scrutinize their reliability. Recently, reward benchmarks that test reward model alignment have been proposed (Lambert et al., 2024). However, while such benchmarks represent important progress, it is important that these benchmarks comprehensively measure alignment rather than only test a few areas that a preference-aligned model has already mastered. Otherwise, the model's performance may appear artificially inflated regardless of how many examples the benchmark contains.

Motivated by this concern and building on the approach proposed by Brown et al. (2020) for alignment verification, we introduce CONECUT, a novel cone-membership test that seeks to detect and prune redundancy in preference datasets for reward model evaluations. We apply CONECUT to RewardBench (Lambert et al., 2024), and show that a significant portion of data, particularly in subsets like safety and reasoning, is near redundant. In particular, in one CONECUT setting, we find that 23.8% of the data is near redundant or redundant, with model performance dropping as much as 3% on the overall dataset and up to 13.98% on individual subsets, when evaluated on minimal non-redundant subsets. In the context of preference alignment, even modest accuracy decreases are particularly

concerning given their direct impact on safety assessment. We expect that CONECUT can enable and accelerate the development of more rigorous reward benchmarks by guiding the creation of less redundant tests of preference alignment and pruning current benchmarks to focus on their most informative examples.

2 Related work

Reward models are a core component of alignment techniques such as Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Bai et al., 2022a). Reward models for LLMs are typically also LLMs finetuned on human-labeled preference pairs, where one completion is more favored than the other, given a prompt. The scores from the trained reward model are then used to guide further model training, typically via policy optimization, to produce outputs that better align with human values such as helpfulness, safety, and ethics. To evaluate these reward models, the most widely used benchmark is RewardBench (Lambert et al., 2024), which includes 2,985 preference pairs across four categories: `chat`, `chat-hard`, `safety`, and `reasoning`. Other recent benchmarks for evaluating reward models include RM-Bench (Liu et al., 2024c), which tests sensitivity to subtle content changes and style bias; VLRewardBench (Li et al., 2025), which extends reward modeling to vision-language generation tasks; and ProcessBench (Zheng et al., 2025), which provides step-level supervision for mathematical reasoning tasks. We chose RewardBench because of the wide range of topics covered and the extensive community interest in this benchmark, as shown by the large number of submissions on the leaderboard.[1]

Naturally, the quality and diversity of these evaluation sets influence how well reward models seem to align with human judgment. Prior work has shown that redundancy in these preference pairs can lead to incorrect evaluations of model safety and reliability (Zhang et al., 2025) and having "approximate clones" in a dataset can lead to reward inflation by skewing the MLE reward scores (Procaccia et al., 2025). These works highlight the need to inspect reward benchmarks for performance leakage and accuracy inflation, particularly when subsets like safety are involved. In our work, we focus on dataset redundancy. Lee et al. (2022) find that removing exact redundancies helps training efficiency. Our work studies the complementary idea of removing redundancies for better evaluation purposes, and our approach generalizes beyond exact duplication detection. While we draw conceptually from Brown et al. (2020), our setting is substantially different. Their work involves formulating alignment as a test of behavioral compatibility using preferences, value queries, or trajectories across low-dimensional MDPs. In contrast, our work focuses on evaluation dataset pruning for LLM reward model benchmarks like RewardBench. To the best of our knowledge CONECUT is the first work that applies geometric redundancy detection on reward model evaluation datasets and successfully detects a significant drop in accuracy on the non-redundant preference pairs, across models.

3 Problem formulation

We consider a preference alignment dataset $\mathcal{D} = \{(p_i, c_i, r_i)\}_{i=1}^{N}$ used in reward model training. Each ordered triple indicates that humans prefer the model completion c_i over r_i for the input prompt p_i. Thus, each preference triple enforces a requirement on the reward model to score c_i higher than r_i, which can be viewed as a *constraint* on the model's behavior. Given a preference alignment dataset $\mathcal{D}$, we seek to prune redundant preferences to find a minimal set of pairwise preference data that best captures the underlying alignment constraints in $\mathcal{D}$.

We can denote the reward model as R_w, where $w \in \mathbb{R}^D$ are the weights of the linear reward head. We assume that if a user prefers c_i over r_i given prompt p_i, then R_w should satisfy $R_w(p_i, c_i) > R_w(p_i, r_i)$. While reward models are usually complex functions of the input, they are often finetuned from a pretrained language model using a reward head that maps the last decoder hidden state before next token prediction to a scalar reward output. Thus, we can treat them as linear functions of the final hidden state, which we denote by $\phi(x, y)$

[1] https://huggingface.co/spaces/allenai/reward-bench

for input x and generation y. Viewing reward models in this way facilitates reasoning about the constraints imposed by each pairwise preference. Following prior work by Brown et al. (2020), we rewrite the last inequality using the weights w and features ϕ, to show that each pairwise preference (p_i, c_i, r_i) induces the following linear constraint on the reward weights:

$$w^\top [\phi(p_i, c_i) - \phi(p_i, r_i)] > 0 \,. \tag{1}$$

The above inequality defines an open *half-space*:

$$\mathcal{H}_i \;:=\; \{\, w \in \mathbb{R}^D : w^\top x_i > 0\}, \qquad x_i := \phi(p_i, c_i) - \phi(p_i, r_i), \tag{2}$$

that separates weight vectors that score the preferred completion higher from the rejected completions. Intersecting all such half-spaces yields the feasible region of the reward model weights that align with the human preferences contained in $\mathcal{D}$:

$$\mathcal{F}(X) := \{w \in \mathbb{R}^D : w^\top x_k > 0,\ \forall k\}, \tag{3}$$

where $X = \{x_1, \ldots, x_N\}$ represents all preference constraints derived from the dataset $\mathcal{D}$. We can now formally define a *redundant pairwise preference*, (p_i, c_i, r_i), as one that induces a constraint $w^\top x_i$ that, if removed from $\mathcal{F}(X)$, does not change the intersecting half space of feasible rewards that align with human preferences, i.e.,

$$\{w \in \mathbb{R}^D : w^\top x_k > 0,\ \forall k \neq i\} = \mathcal{F}(X). \tag{4}$$

Next, we discuss several methods for finding such redundancies in a preference dataset.

4 Background: Redundancy detection via linear programming

Our approach draws from prior works in AI safety, particularly Brown et al. (2020), which addresses value alignment verification in reinforcement learning by proposing a theoretical approach for alignment testing to ensure an agent's policy satisfies constraints derived from human preferences. To remove redundancy, they use an approach based on linear programming (LP) as described below.

Following Telgen (1983), determining whether a preference (p_i, c_i, r_i) is redundant according to the definition (4) is equivalent to finding the smallest value $w^\top x_i$ can take, under the constraint that w satisfies all other inequalities (i.e., aligns with all other preferences in $\mathcal{D}$):

$$\min_w \; w^\top x_i \quad \text{s.t.} \quad w^\top x_k \geq 0,\ \forall k \neq i \,. \tag{5}$$

If $w_*^\top x_i \geq 0$, where w_* is the solution to (5), then preference (p_i, c_i, r_i) (corresponding to the half-space normal vector $x_i = \phi(p_i, c_i) - \phi(p_i, r_i)$) is redundant since even without enforcing $w^\top x_i \geq 0$, any solution w that satisfies all other constraints will automatically satisfy this constraint.

Brown et al. (2020) apply this test to create non-redundant preference tests for verifying alignment. However, applying this test to a dataset with N preference pairs requires solving N such LPs. Brown et al. (2020) only consider low-dimensional problems with embedding size $D < 6$ and tens of preference queries; however, directly applying this approach to LLMs is intractable since the last decoder head usually has at least several hundred dimensions, and reward models for LLMs are usually evaluated on hundreds or thousands of pairwise preferences.

5 CONECUT: Cone membership test for preference redundancy

To address the scalability challenges, we propose a more tractable test for redundancy based on *cone membership*. Let $X \in \mathbb{R}^{N \times D}$ be the matrix whose rows are vectors x_i induced by each pairwise preference in the dataset $\mathcal{D}$. We now define a preference (p_i, c_i, r_i) as *redundant* if and only if its corresponding half-space normal vector $x_i = \phi(p_i, c_i) - \phi(p_i, r_i)$ lies in the

cone generated by the remaining rows of X:[2]

$$x_i \in \mathcal{C}(X_{-i}) := \Big\{ \sum_{k \neq i} \alpha_k \, x_k, \; \alpha_k \geq 0 \Big\}. \tag{6}$$

However, exact duplicates of preference pairs are rare in real-world datasets because of annotator noise, paraphrasing, and representation drift (Shen et al., 2024; Deng et al., 2025; Tan et al., 2025). We thus define a preference pair as ε-*redundant* if x_i can be approximated within an error margin ε by the cone $\mathcal{C}(X_{-i})$. We implement this using Non-Negative Least Squares (NNLS; Lawson & Hanson, 1995), solving:

$$r_i = \min_{a_k \geq 0} \left\| x_i - \sum_{k \neq i} a_k x_k \right\|_2 \qquad R_i^2 = 1 - \frac{\|x_i - r_i\|_2^2}{\|x_i\|_2^2}, \tag{7}$$

where r_i is the residual and R^2 the coefficient of determination, which quantifies approximation quality. We retain x_i as a *non-redundant* constraint only if $R_i^2 < \varepsilon$ where we consider the thresholds $\varepsilon \in \{0.95, 0.9\}$. We choose these thresholds empirically to ensure practical applicability in high-dimensional settings; see more details in Appendix B. The exact LP approach used in prior work (Brown et al., 2020) requires $O(ND^3)$ time ($N = 2{,}985$, $D = 4{,}096$ in RewardBench with reward models we study), whereas NNLS costs $O(ND^2)$. In practice, we use SciPy's `nnls` which yields $\approx 10^3\times$ speed-up and enables us to scale preference redundancy detection to modern LLMs.

6 Experiments

In this section, we detail our experiments assessing the impact of redundancy in preference test sets on reward model performance, focusing on RewardBench (Lambert et al., 2024).

6.1 Experimental setup

Dataset. RewardBench is a comprehensive benchmark used to evaluate reward model performance on human preference pairs. It consists of 2,985 examples across 4 data subsets: `chat` (358), `chat-hard` (456), `safety` (740), and `reasoning` (1431). The chat subset tests conversational abilities, the safety subset assesses alignment with human-described safe behaviors like refusals to dangerous and offensive responses, and the reasoning subset evaluates logical and problem-solving skills.

Feature extractor. To determine which preferences are ε-redundant, we use embeddings ϕ from the penultimate layer in `ShikaiChen/LDL-Reward-Gemma-2-27B-v0.1` in Huggingface — one of the top models on the RewardBench leaderboard. We use the standard chat prompt template used in RewardBench and use the last token representation as the feature representation.

Settings. We test a range of reward models submitted to RewardBench under four conditions: (1) *Full Dataset:* Accuracy on the entire RewardBench dataset; (2) *Redundant Subset:* Accuracy on pairs identified as ε-redundant with NNLS from the LDL Reward Gemma model; (3) *Non-Redundant Subset:* Accuracy on pairs not marked as redundant; and (4) *Random Reduced Dataset:* Accuracy on a test set where a random subset of pairs (same size as the redundant set) is removed. By evaluating performance across these conditions, we isolate the impact of ε-redundancy, hypothesizing that redundant pairs inflate the benchmark accuracy, particularly in safety and reasoning subsets, due to overrepresented patterns.

Models. We benchmark eight open–source reward models that currently appear on the RewardBench leaderboard: `GRM-Llama-3.2-3B` and `GRM-Llama-3-8B` (Yang et al., 2024), `LDL Reward Gemma-27B`, `QRM Llama-3.1-8B` (Dorka, 2024), `Llama-3 OffsetBias-8B` (Park et al., 2024), `Skywork-Reward-Llama-3.1-8B-v0` and `v0.2` (Liu et al., 2024a), and `InternLM-2-7B-Reward` (Cai et al., 2024). Model details are provided in Appendix C.

[2] A proof of the equivalence "redundancy $\iff$ cone membership" is provided in Appendix A.

Subset	Model	Full Acc.	Non-Red. Acc.	Red. Acc.	Rand. Acc.
`chat` (5.59%)	GRM-llama3.2-3B-rewardmodel-ft	91.62	91.12	100.00	91.42
	GRM-llama3-8B-distill	98.32	98.22	100.00	98.22
	LDL-Reward-Gemma-2-27B-v0.1	96.37	96.15	100.00	96.15
	Llama-3-OffsetBias-RM-8B	97.21	97.04	100.00	97.04
	QRM-Llama3.1-8B-v2	96.65	96.45	100.00	96.45
	Skywork-Reward-Llama-3.1-8B-v0.2	94.69	94.38	100.00	94.38
	Skywork-Reward-Llama-3.1-8B	95.81	95.56	100.00	95.56
	internlm2-7b-reward	99.44	99.41	100.00	99.41
	Average	**96.26**	**96.04**	**100.00**	**96.08**
`chat-hard` (4.39%)	GRM-llama3.2-3B-rewardmodel-ft	84.43	83.72	100.00	84.40
	GRM-llama3-8B-distill	68.20	67.66	80.00	67.66
	LDL-Reward-Gemma-2-27B-v0.1	90.57	90.14	100.00	90.83
	Llama-3-OffsetBias-RM-8B	82.02	81.65	90.00	81.88
	QRM-Llama3.1-8B-v2	86.84	86.24	100.00	86.70
	Skywork-Reward-Llama-3.1-8B-v0.2	88.38	87.84	100.00	88.30
	Skywork-Reward-Llama-3.1-8B	87.06	86.47	100.00	86.93
	internlm2-7b-reward	71.05	70.18	90.00	70.64
	Average	**82.32**	**81.74**	**95.00**	**82.17**
`safety` (43.51%)	GRM-llama3.2-3B-rewardmodel-ft	92.57	86.71	100.00	92.03
	GRM-llama3-8B-distill	86.49	77.05	98.47	85.75
	LDL-Reward-Gemma-2-27B-v0.1	93.65	88.65	100.00	93.00
	Llama-3-OffsetBias-RM-8B	87.03	77.78	98.77	86.47
	QRM-Llama3.1-8B-v2	92.70	86.96	100.00	92.75
	Skywork-Reward-Llama-3.1-8B-v0.2	92.70	86.96	100.00	92.51
	Skywork-Reward-Llama-3.1-8B	90.95	83.82	100.00	89.86
	internlm2-7b-reward	87.43	73.45	96.44	88.62
	Average	**90.44**	**82.67**	**99.21**	**90.12**
`reasoning` (24.39%)	GRM-llama3.2-3B-rewardmodel-ft	94.20	92.88	98.28	94.09
	GRM-llama3-8B-distill	91.35	89.30	98.14	91.14
	LDL-Reward-Gemma-2-27B-v0.1	98.85	97.69	100.00	98.34
	Llama-3-OffsetBias-RM-8B	91.95	90.02	99.14	91.35
	QRM-Llama3.1-8B-v2	96.09	94.92	99.71	96.03
	Skywork-Reward-Llama-3.1-8B-v0.2	96.66	95.42	100.00	96.43
	Skywork-Reward-Llama-3.1-8B	96.27	94.55	100.00	96.46
	internlm2-7b-reward	94.64	93.81	98.85	94.40
	Average	**95.00**	**93.57**	**99.27**	**94.78**

Table 1: Reward model accuracies for RewardBench across full, non-redundant, redundant, and randomly sampled subsets. Redundancy is determined using CONECUT with $\varepsilon = 0.95$.

6.2 Results

Table 1 and Table 2 shows our results for $\varepsilon = 0.95$ and $\varepsilon = 0.9$, respectively. The first column in both the tables shows the percentage of redundant examples based on the ε-redundancy found in different subsets of RewardBench, which are substantial for the `safety` (43.5% & 60.8% redundancy) and `reasoning` (24.4% & 59% redundancy) data subsets. See Table 3 (Appendix) for absolute counts.

We hypothesize that redundancy is more pronounced in `safety` because preferred completions refuse to complete a dangerous or harmful prompt, and respond with a similar set of sentences like "*I'm sorry, but I cannot fulfill that request. It goes against my values to promote harmful...*" and "*...If you have any other inquiries or topics you'd like to discuss, feel free to let me know...*".

All models across all subsets show a drop in accuracy when reported on the non-redundant set and a close to 100% accuracy on the redundant set. In low-redundancy domains like `chat`, most models show negligible differences across subsets. However, in subsets with significant redundancy, `safety` (43.5% & 60.8%) and `reasoning` (24.4% & 59%), some models such as LDL-Reward-Gemma and Llama-3 variants achieve perfect or near-perfect scores on redundant pairs while dropping by up to 10 percentage points on the non-redundant ones. The drop is more notable in a non-redundant set found with CONECUT than on a randomly sampled one. This reveals that redundancy can mask real weaknesses and inflate performance if not accounted for, and provides evidence that CONECUT can be used as a tool to better scrutinize and assess the true performance and alignment of reward models.

Subset	Model	Full Acc.	Non-Red. Acc.	Red. Acc.	Rand. Acc.
`chat` (25.4%)	GRM-llama3.2-3B-rewardmodel-ft	91.62	89.1	98.9	91.0
	GRM-llama3-8B-distill	98.32	97.8	100.0	98.5
	LDL-Reward-Gemma-2-27B-v0.1	96.37	95.1	100.0	97.0
	Llama-3-OffsetBias-RM-8B	97.21	96.6	98.9	96.6
	QRM-Llama3.1-8B-v2	96.65	95.5	100.0	96.6
	Skywork-Reward-Llama-3.1-8B-v0.2	94.69	93.3	98.9	95.5
	Skywork-Reward-Llama-3.1-8B	95.81	94.4	100.0	95.9
	internlm2-7b-reward	99.44	99.3	100.0	99.6
	Average	**96.26**	**95.14**	**99.59**	**96.34**
`chat-hard` (21.9%)	GRM-llama3.2-3B-rewardmodel-ft	84.43	80.1	100.0	84.0
	GRM-llama3-8B-distill	68.20	65.2	79.0	68.3
	LDL-Reward-Gemma-2-27B-v0.1	90.57	87.9	100.0	91.3
	Llama-3-OffsetBias-RM-8B	82.02	78.1	96.0	83.1
	QRM-Llama3.1-8B-v2	86.84	83.1	100.0	87.9
	Skywork-Reward-Llama-3.1-8B-v0.2	88.38	85.1	100.0	89.3
	Skywork-Reward-Llama-3.1-8B	87.06	83.4	100.0	87.4
	internlm2-7b-reward	71.05	68.8	79.0	71.1
	Average	**82.32**	**78.96**	**94.25**	**82.80**
`safety` (60.8%)	GRM-llama3.2-3B-rewardmodel-ft	92.57	81.7	99.6	92.8
	GRM-llama3-8B-distill	86.49	71.7	96.0	87.6
	LDL-Reward-Gemma-2-27B-v0.1	93.65	84.5	99.6	94.5
	Llama-3-OffsetBias-RM-8B	87.03	71.7	96.9	87.9
	QRM-Llama3.1-8B-v2	92.70	82.1	99.6	92.8
	Skywork-Reward-Llama-3.1-8B-v0.2	92.70	82.1	99.6	92.8
	Skywork-Reward-Llama-3.1-8B	90.95	77.9	99.3	91.4
	internlm2-7b-reward	87.43	73.4	96.4	88.6
	Average	**90.44**	**78.14**	**98.38**	**91.05**
`reasoning` (59.0%)	GRM-llama3.2-3B-rewardmodel-ft	94.20	88.2	98.3	94.2
	GRM-llama3-8B-distill	91.35	86.0	96.9	92.0
	LDL-Reward-Gemma-2-27B-v0.1	98.85	95.7	100.0	98.8
	Llama-3-OffsetBias-RM-8B	91.95	85.7	98.3	94.4
	QRM-Llama3.1-8B-v2	96.09	91.7	99.2	95.7
	Skywork-Reward-Llama-3.1-8B-v0.2	96.66	91.3	99.5	96.3
	Skywork-Reward-Llama-3.1-8B	96.27	91.8	98.7	95.7
	internlm2-7b-reward	94.64	91.5	97.5	94.7
	Average	**95.00**	**90.24**	**98.55**	**95.23**

Table 2: Reward model accuracies for RewardBench across full, non-redundant, redundant, and randomly sampled subsets. Redundancy is determined using ConeCut with $\varepsilon = 0.90$.

7 Conclusion and social impact

We present a novel method, ConeCut, for reward model evaluation by pruning redundant or near-redundant examples in preference datasets such as RewardBench. We formulate a cone membership test for reward models, implemented via a non-negative least squares algorithm to find ε-redundant preference pairs from the evaluation dataset. Our experiments find redundancy in the RewardBench dataset and demonstrate that this redundancy can lead to inflated performance. Our findings highlight critical societal risks: deploying misaligned models in high-stakes settings—such as content moderation or automated decision-making—can result in user harm and diminished public confidence. Our work promotes improved investigation of popular reward evaluation benchmark results, and we hope that our results will inspire a stronger focus on redundancy-aware benchmark creation to enable better alignment testing.

8 Acknowledgments

We thank Rishanth Rajendhran for the initial codebase. We are also grateful to the UtahNLP lab and anonymous reviewers for valuable feedback.

References

Anthropic. Claude: A conversational ai model. https://www.anthropic.com/claude, 2023.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862, 2022a. doi: 10.48550/ARXIV.2204.05862. URL https://doi.org/10.48550/arXiv.2204.05862.

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. Constitutional ai: Harmlessness from ai feedback, 2022b. URL https://arxiv.org/abs/2212.08073.

Daniel S. Brown, Jordan Schneider, Anca D. Dragan, and Scott Niekum. Value alignment verification, 2020. URL https://arxiv.org/abs/2012.01557.

Zheng Cai, Maosong Cao, Haojiong Chen, and *et al.* Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024. URL https://arxiv.org/abs/2403.17297.

Jared Casper, Xander Lin, Joe Chen, and Esin Durmus. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023. URL https://arxiv.org/abs/2307.15217.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4302–4310, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964. URL https://arxiv.org/abs/1706.03741.

Mahavir Dabas, Si Chen, Charles Fleming, Ming Jin, and Ruoxi Jia. Just enough shifts: Mitigating over-refusal in aligned language models with targeted representation fine-tuning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=TiYOHdK35L.

Xun Deng, Han Zhong, Rui Ai, Fuli Feng, Zheng Wang, and Xiangnan He. Less is more: Improving llm alignment via preference data selection, 2025. URL https://arxiv.org/abs/2502.14560.

Nicolai Dorka. Quantile regression for distributional reward models in rlhf, 2024. URL https://arxiv.org/abs/2409.10164.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling, 2024. URL https://arxiv.org/abs/2403.13787.

Charles L. Lawson and Richard J. Hanson. *Solving Least Squares Problems*, volume 15 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995. ISBN 978-0-89871-356-5. doi: 10.1137/1.9781611971217. URL https://epubs.siam.org/doi/book/10.1137/1.9781611971217. Revised reprint of the 1974 Prentice–Hall edition.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better, 2022. URL https://arxiv.org/abs/2107.06499.

Lei Li, Yuancheng Wei, Zhihui Xie, Xuqing Yang, Yifan Song, Peiyi Wang, Chenxin An, Tianyu Liu, Sujian Li, Bill Yuchen Lin, Lingpeng Kong, and Qi Liu. Vl-rewardbench: A challenging benchmark for vision-language generative reward models, 2025. URL https://arxiv.org/abs/2411.17451.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in LLMs. *arXiv preprint arXiv:2410.18451*, 2024a. URL https://arxiv.org/abs/2410.18451.

Wenhao Liu, Xiaohua Wang, Muling Wu, Tianlong Li, Changze Lv, Zixuan Ling, Zhu JianHao, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. Aligning large language models with human preferences through representation engineering. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10619–10638, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.572. URL https://aclanthology.org/2024.acl-long.572/.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. Rm-bench: Benchmarking reward models of language models with subtlety and style, 2024c. URL https://arxiv.org/abs/2410.16184.

OpenAI. Chatgpt: Optimizing language models for dialogue. https://openai.com/chatgpt, 2023.

Junsoo Park, Seungyeon Jwa, Meiying Ren, Daeyoung Kim, and Sanghyuk Choi. Offsetbias: Leveraging debiased data for tuning evaluators. *Findings of EMNLP 2024*, 2024. URL https://arxiv.org/abs/2407.06551.

Ariel D. Procaccia, Benjamin Schiffer, and Shirley Zhang. Clone-robust ai alignment, 2025. URL https://arxiv.org/abs/2501.09254.

Judy Hanwen Shen, Archit Sharma, and Jun Qin. Towards data-centric rlhf: Simple metrics for preference dataset comparison, 2024. URL https://arxiv.org/abs/2409.09603.

Haoru Tan, Sitong Wu, Wei Huang, Shizhen Zhao, and Xiaojuan Qi. Data pruning by information maximization, 2025. URL https://arxiv.org/abs/2506.01701.

Jan Telgen. Identifying redundant constraints and implicit equalities in systems of linear constraints. *Management Science*, 29(10):1209–1222, October 1983. doi: 10.1287/mnsc.29.10.1209. URL https://pubsonline.informs.org/doi/epdf/10.1287/mnsc.29.10.1209.

Jeremy Tien, Jerry Zhi-Yang He, Zackory Erickson, Anca D Dragan, and Daniel S Brown. Causal confusion and reward misidentification in preference-based reward learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2204.06601.

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. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for LLMs. *arXiv preprint arXiv:2406.10216*, 2024. URL https://arxiv.org/abs/2406.10216.

Zicheng Zhang, Xiangyu Zhao, Xinyu Fang, Chunyi Li, Xiaohong Liu, Xiongkuo Min, Haodong Duan, Kai Chen, and Guangtao Zhai. Redundancy principles for MLLMs benchmarks, 2025. URL https://arxiv.org/abs/2501.13953.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical reasoning, 2025. URL https://arxiv.org/abs/2412.06559.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020. URL https://arxiv.org/abs/1909.08593.

Subset	# Total	$\epsilon = 0.95$		$\epsilon = 0.90$	
		# Red.	Percent (%)	# Red.	Percent (%)
chat	358	20	5.59	91	25.4
chat_hard	456	20	4.39	100	21.9
safety	740	322	43.51	450	60.8
reasoning	1431	349	24.39	845	59.0

Table 3: Redundancy rates using the NNLS cone membership test with $\epsilon = 0.95$, and 0.90

A Proof of redundancy–cone membership equivalence

In this section, we prove that a preference (p_i, c_i, r_i) is *redundant*, i.e., $\mathcal{F}(X) = \mathcal{F}(X_{-i})$, if and only if x_i lies in the cone generated by the remaining preferences, i.e., $x_i \in \mathcal{C}(X_{-i})$.

A.1 Definitions and notation used in the proof

We begin by briefly recalling the key definitions and notation introduced in the main text.

Redundancy. As a reminder, $X = \{x_1, \ldots, x_N\}$ represents all preference constraints in the dataset $\mathcal{D}$, and the feasible region of the reward model weights that align with human preferences is defined as $\mathcal{F}(X) = \{w \in \mathbb{R}^d : w^\top x_k \geq 0, \ \forall k\}$. We call a preference (p_i, c_i, r_i) *redundant* if removing x_i leaves the feasible set of halfspace constraints unchanged, i.e., $\mathcal{F}(X) = \mathcal{F}(X_{-i})$, where $X_{-i} := X \setminus \{x_i\}$.

Cone of constraints. The cone spanned by all preference excluding i-th is given by:

$$\mathcal{C}(X_{-i}) := \Big\{ \sum_{k \neq i} \alpha_k \, x_k, \ \alpha_k \geq 0 \Big\}.$$

NNLS solver for cone membership. The non-negative least-squares (NNLS) residual is:

$$r_i \ := \ \min_{\alpha_k \geq 0} \left\| x_i - \sum_{k \neq i} \alpha_k x_k \right\|_2 .$$

A.2 Proof

For every constraint x_i the following statements are equivalent:

$$\boxed{x_i \text{ is redundant}} \iff \boxed{x_i \in \mathcal{C}(X_{-i})} \iff \boxed{r_i = 0}.$$

Hence, solving the NNLS sub-problem provides an exact test for redundancy.

(i) *Redundancy $\Rightarrow$ cone membership.*

Assume x_i is redundant but $x_i \notin \mathcal{C}(X_{-i})$. By the separating hyperplane theorem, there exists w such that $w^\top x_k \geq 0$ for all $k \neq i$ while $w^\top x_i \leq -\varepsilon < 0$. Thus $w \in \mathcal{F}(X_{-i})$ but $w \notin \mathcal{F}(X)$, which contradicts redundancy. Therefore $x_i \in \mathcal{C}(X_{-i})$.

(ii) *Cone membership $\Rightarrow$ redundancy.*

If $x_i = \sum_{k \neq i} \alpha_k x_k, \ \alpha_k \geq 0$, and w satisfies $w^\top x_k \geq 0, \ \forall k \neq i$, then $w^\top x_i = \sum_{k \neq i} \alpha_k w^\top x_k \geq 0$. Hence $\mathcal{F}(X_{-i}) \subseteq \mathcal{F}(X)$, so dropping x_i does *not* shrink the feasible set. In other words, if a constraint is already inside the cone, then others can make up for it, even if we drop this constraint.

(iii) $r_i = 0 \iff$ *cone membership.*

The NNLS residual is precisely the Euclidean distance from x_i to $\mathcal{C}(X_{-i})$; thus $r_i = 0$ iff $x_i \in \mathcal{C}(X_{-i})$.

Combining (i)–(iii) establishes the claimed equivalence.

B Selection of appropriate ε

In this section, we discuss how much slack does the ε redundancy introduce over absolute redundancy when solving the same problem using linear programming (LP).

Notation. As a reminder, we decide if a preference pair (represented by vector x_i) is ε-redundant by checking how well it can be approximated by the cone generated by all other preference vectors, $\mathcal{C}(X_{-i})$. We use the coefficient of determination R^2 from Non-Negative Least Squares (NNLS) (Lawson & Hanson, 1995), solving:

$$r_i = \min_{a_k \geq 0} \left\| x_i - \sum_{k \neq i} a_k x_k \right\|_2 \qquad R_i^2 = 1 - \frac{\|x_i - r_i\|_2^2}{\|x_i\|_2^2}, \tag{8}$$

where r_i is the residual and R^2 the coefficient of determination, which quantifies approximation quality.

Approximate redundancy. Let relative residual be $\rho_i := \|x_i - r_i\|_2 / \|x_i\|_2$, so $R_i^2 = 1 - \rho_i^2$. We set $\varepsilon = 0.95$, implying $\rho_i \leq \sqrt{0.05} \approx 0.224$ for redundancy. This means, constraints exceeding our R^2 threshold can be reconstructed to within 22.4% of their l_2-norm by using linear combinations of other constraints, while explaining 95% variance of the said constraint. For $\varepsilon = 0.90$, this value is 0.316, implying a reconstruction error of 31.6%, with 90% variance explained for constraints exceeding our R^2 threshold.

We can also quantify the angular deviation implied by this approximation. If $R_i^2 \geq 0.95$, let $\theta_i = \angle(x_i, \hat{x}_i)$ be the angle between the two vectors. By the law of cosines,

$$\rho^2 \|x_i\|_2^2 \;=\; \|x_i\|_2^2 + \|\hat{x}_i\|_2^2 - 2\,\|x_i\|_2\,\|\hat{x}_i\|_2 \cos\theta_i.$$

The residual is smallest when R_i^2 is close to 1 or $\|x_i\|_2 = \|\hat{x}_i\|_2$. Setting $\|x_i\|_2 = \|\hat{x}_i\|_2$ and dividing by $\|x_i\|_2^2$ gives:

$$\rho^2 = 2 - 2\cos\theta_i \;\;\Longrightarrow\;\; \cos\theta_i = 1 - \frac{\rho^2}{2} \tag{9}$$

This solves to:

$$\boxed{\theta_i \;\leq\; \arccos\!\Big(1 - \frac{\rho^2}{2}\Big) = \arccos\!\Big(1 - \tfrac{0.05}{2}\Big) = \arccos(0.975) \;\approx\; 12.8^\circ} \tag{10}$$

This implies that pruning ε-redundant constraints at $R^2 \geq 0.95$ makes sure that the maximum deviation of that constraint from a true cone element is $\approx 13^\circ$. If we instead use $\varepsilon = 0.90$, this maximum deviation becomes $\approx 18^\circ$.

C Models

In this section, we provide more information about models that we evaluate on various data subsets.

- **Generalisable Reward Models (GRM).** We use the GRM-LLAMA-3.2 3B and GRM-LLAMA-3 8B variants, whose hidden-state regularisation improves out-of-distribution robustness (Yang et al., 2024).
- **LDL Reward Gemma 27B.** This model predicts a *label distribution* rather than a point estimate, yielding smoother gradients for RLHF.
- **Quantile Reward Model (QRM) Llama-3.1 8B.** QRM learns a full reward *distribution* via quantile regression (Dorka, 2024).
- **Llama-3 OffsetBias 8B.** Fine-tuned on the OFFSETBIAS corpus to mitigate length and style artifacts in judge models (Park et al., 2024).

- **Skywork-Reward Llama-3.1 8B (v0 and v0.2).** Trained on an 80k, data-centric preference set that emphasises high-quality, diverse prompts (Liu et al., 2024a).
- **InternLM-2 7B Reward.** Produced via the COOL-RLHF pipeline described in the InternLM2 technical report (Cai et al., 2024).