

RLHF Algorithms Ranked: An Extensive Evaluation Across Diverse Tasks, Rewards, and Hyperparameters

Lucas Spangher

Google Research
Mountain View, CA

spangher@google.com

Rama Kumar Pasumarthi

Google Research
Mountain View, CA

ramakumar@google.com

Nick Masiewicki

Google Research
Mountain View, CA

mmasiewicki@google.com

William F. Arnold

KAIST
Seoul, Korea

willarnold@berkeley.edu

Dale Johnson

Google Research
Mountain View, CA

dalejohnson@google.com

Peter Grabowski

Google Research
Mountain View, CA

petrgrabowski@google.com

Eugene Ie

Google Research
Mountain View, CA

eugeneie@google.com

Abstract

Large Language Models (LLMs) have demonstrated impressive text generation capabilities, yet their outputs often misalign with human preferences. To address this challenge, Reinforcement Learning from Human Feedback (RLHF) has become an essential component of modern LLM training pipelines. Although Proximal Policy Optimization (PPO) initially emerged as a favored RLHF strategy, its complexity and inefficiency have spurred the investigation of simpler alternatives. This work presents, to the authors' knowledge, the most comprehensive benchmark to date of seventeen state-of-the-art RLHF algorithms. We evaluate these algorithms on two different benchmarks, OpenAI's TL;DR Summarization and Anthropic's Helpfulness / Harmlessness, with two different reward models a Gemma 2B Reward model and a Rules based reward model. We incorporate extensive hyperparameter sweeps for each algorithm. With this expanded analysis, we report consistently top-performing RLHF algorithms: IPO, DPO, Reinforce, GRPO, and Best-of-N, and list the highest performing hyperparameter combinations for each. This work aims to guide practitioners in selecting the most effective RLHF algorithm while promoting a culture of thorough and impartial benchmarking in the field.

1 Introduction

Aligning Large Language Models (LLMs) with human preferences is a key challenge, making Reinforcement Learning from Human Feedback (RLHF) an important step in the training pipeline (Wang et al., 2024).

While Proximal Policy Optimization (PPO) was an early standard for RLHF (Christiano et al., 2023), it has since fallen out of favor. Numerous alternatives—such

as Best of N Cloning (Nakano et al., 2021), KTO (Ethayarajh et al., 2024), GRPO (Shao et al., 2024), DPO (Rafailov et al., 2024), and RSO (Liu et al., 2024)—have been proposed. However, these methods are typically evaluated against few, inconsistent baselines. Even the recent work that inspired our paper, a theoretically important and eloquent argument for simpler methods like REINFORCE over the complexity of PPO (Ahmadian et al., 2024), was limited to a one-off comparison, highlighting the need for a thorough, impartial benchmark.

In this work, we conduct the most computationally intensive benchmark of RLHF algorithms to date. We revisit simple policy gradient methods alongside more complex strategies, evaluating a full suite of sixteen algorithms: Actor Critic (AC), Best of N Cloning (BON), Direct Preference Optimization (DPO), Direct Reward Optimization (DRO), Deep Q Networks (DQN), Exact Preference Optimization (EXO), General Preference Optimization (GPO), Group Relative Policy Optimization (GRPO), Identity Preference Optimization (IPO), Inverse Q Learning (IQLearn), KTO, Path Consistency Learning (PCL), PPO, REINFORCE, REINFORCE with Baseline, and RSO. We evaluate each on three different tasks with extensive hyperparameter sweeps, totaling over 3,500 training runs and 30,000 TPU-hours to identify the top-performing configurations.

Our work is novel in the following ways:

- We test an expanded set of algorithms on well-studied tasks and perform **a more extensive hyperparameter search than prior work**, with hyperparameters chosen after consulting the original authors.
- We use **a more extensive suite of evaluation metrics**, including ROUGE scores, for a more complete picture of model performance.
- We **provide the top hyperparameter combina-**

tions found per algorithm and task to save readers from extensive exploration.

Ultimately, we aim to guide practitioners in selecting an out-of-the-box RLHF algorithm and to encourage a culture of impartial, broad-selection benchmarking.

2 RLHF Preliminaries

In Reinforcement Learning from Human Feedback (RLHF), the learning agent is tasked with optimizing its behavior based on human-provided feedback (which could take the form of selecting a winning response, upvote/downvote, or ranked choice.) Formally, RLHF, like RL, can be modeled as a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $\mathcal{P}(s' | s, a)$ is the transition probability from state s to s' given action a , and $\gamma \in [0, 1)$ is the discount factor that controls the weight of future rewards. In RLHF, the reward function $R(s, a)$ is often provided by a separate Reward Model, oftentimes a special purpose LLM fine-tuned on human preference data. This feedback is used to shape the original LLM to expressing output $\pi(a | s)$ that aligns with human values and objectives.

3 Related Works: Comparative RLHF Studies

A number of prior studies have systematically compared reinforcement learning algorithms for RLHF, with a particular focus on design choices and hyperparameter configurations. For example, the seminal “Back to Basics” paper (Ahmadian et al., 2024) rigorously evaluated PPO, REINFORCE, DPO, and two variants of REINFORCE (Williams, 1992a). Although the case for REINFORCE over PPO and DPO was argued persuasively, the hyperparameter exploration was somewhat limited. Importantly, the study concluded that PPO is suboptimal for RLHF and that simpler algorithms like REINFORCE and its variant RLOO (Williams, 1992a; Kool et al., 2019) outperform PPO under certain conditions.

In addition, several works have specifically examined the comparative effectiveness of PPO and Direct Policy Optimization (DPO) (Rafailov et al., 2024; Schulman et al., 2017). These investigations emphasized the influence of dataset selection and mixture configurations on performance. In particular, the Tulu series (Tulu 1, 2, and 2.5) (Wang et al., 2023; Ivison et al., 2023; Ivison et al.) conducted thorough ablation studies on open-source dataset mixtures applied to pure Supervised Fine-Tuning (SFT) models, as well as models trained with DPO alone and in combination with PPO. Their findings indicate that PPO can slightly outperform DPO on several benchmarks, and that augmenting models with larger reward models and synthetic data annotated for helpfulness leads to improved performance. However, these studies focused primarily

on PPO and DPO, with limited hyperparameter exploration and little consideration of other promising algorithms.

4 Methods

4.1 A Taxonomy of the major RLHF approaches we study

We study seventeen RLHF algorithms¹, which we categorize as: (1) classic RL approaches using reward models, (2) direct preference optimization approaches, and (3) sampling-based approaches.

4.1.1 Classic RL Architectures with Reward Models

This class of methods uses classic RL formulations (policy-gradient or value-based) where learning is guided by a separate reward model trained on preference data. These include AC, DQN, IQLearn, PCL, PI, PPO, and REINFORCE variants.

AC: Actor-Critic (Mnih et al., 2016) Actor-Critic uses an actor to select actions and a critic to evaluate them, providing feedback to optimize the policy.

DQN: Deep Q-Networks (Mnih, 2013) Deep Q-Networks (DQN) use a deep neural network to approximate the optimal action-value function, $Q(s, a)$, by applying the Bellman update rule.

GRPO: Group Relative Policy Optimization (Shao et al., 2024) Group Relative Policy Optimization (GRPO) is a critic-free algorithm that uses the mean reward of multiple candidate responses as a baseline for advantage estimation, applying a KL penalty to stay close to a reference model.

IQLearn (Garg et al., 2021; Wulfmeier et al., 2024) IQ-Learn applies inverse Q-learning, connecting KL-regularized RLHF to GAN-based methods (Ho and Ermon, 2016) and deriving a unified loss optimized with a reward model.

PCL: Path Consistency Learning (Nachum et al., 2017) Path Consistency Learning (PCL), from GFlowNet literature (Bengio et al., 2021), optimizes policies by enforcing consistency between different action paths, improving learning efficiency.

PI: Policy Iteration Policy Iteration is a classic RL algorithm that alternates between evaluating the current policy and improving it based on the resulting value function.

¹All RLHF algorithms studied are released under the Apache 2.0 license. This license is a permissive open-source license that allows users to use, modify, and distribute the software with minimal restrictions, including in commercial applications.

PPO: Proximal Policy Optimization (Ouyang et al., 2022; Schulman et al., 2017) Proximal Policy Optimization (PPO) improves stability by using a clipped objective function to restrict large policy changes during training.

REINFORCE (Williams, 1992b) REINFORCE is a Monte Carlo policy gradient method that updates the policy to maximize expected reward but can have high variance.

REINFORCE with Baselines (Sutton et al., 1999) This variant reduces variance by subtracting a baseline, typically the state’s expected value, from the reward signal for more stable learning.

4.1.2 Direct Preference Approaches

Direct Preference approaches directly use preference data in the optimization objective, bypassing a separate reward model. While potentially more data-efficient (Rafailov et al., 2024), this can limit exploration. These methods often use KL regularization and include DPO, IPO, GPO, EXO, DRO, KTO, and SLiC.

GPO: General Preference Optimization (Tang et al., 2024) General Preference Optimization (GPO) optimizes a general representation of user preferences, with DPO and IPO as special cases, and uses KL regularization.

EXO: Exact Preference Optimization (Ji et al., 2024) Exact Preference Optimization (EXO) directly optimizes the model on preference signals, using an anchor distribution as a safe baseline to constrain policy updates.

DPO: Direct Preference Optimization (Rafailov et al., 2024) Direct Preference Optimization (DPO) trains directly on preference triplets (prompt, preferred, dis-preferred) using a Bradley-Terry model and KL regularization. Our implementation is *online*, collecting new preferences during training (Li et al., 2024).

IPO: Identity Preference Optimization (Azar et al., 2024) Identity Preference Optimization (IPO) extends DPO to handle arbitrary preference models beyond pairwise comparisons, using KL-regularization to stay close to a baseline.

DRO: Direct Reward Optimization (Richemond et al., 2024) Inspired by DPO, Direct Reward Optimization (DRO) optimizes a policy on explicit reward signals but requires only a single reward per sample, not a contrastive pair.

KTO: Kahneman-Tversky Optimization (Ethayarajh et al., 2024) Kahneman-Tversky Optimization (KTO) uses principles from behavioral economics to operate on simple feedback (e.g., thumbs up/down) rather than more costly contrastive samples.

SLiC: Sequence Likelihood Calibration (Zhao et al., 2023) Sequence Likelihood Calibration (SLiC) fine-tunes by increasing the likelihood of preferred sequences while decreasing dis-preferred ones to better align with human expectations.

4.1.3 Sampling based approaches

These algorithms generate multiple candidate outputs and select or clone the best-performing one, leveraging sample diversity. This group includes Best of N Cloning and Rejection Sampling Optimization.

BON: Best of N Cloning (Nakano et al., 2021) Best of N Cloning (BON) trains the model to mimic the best output selected from a set of N generated candidates.

RSO: Rejection Sampling Optimization (Liu et al., 2024) Rejection Sampling Optimization (RSO) refines outputs by using rejection sampling to discard less-preferred responses and reinforce optimal ones.

4.2 Problem Setup

Model: We perform experiments fixing base model, reward model, and all other training parameters used prior to the RLHF training stage. For each task the base model is a 2B parameter Gemma summarization model (Team et al., 2024)², which has been specifically trained for the task it represents. In our experiments, we will use two different reward models:

1. The reward signal for RLHF will be provided by a 2B Gemma-based reward model, trained to evaluate the quality of summaries based on human preferences.
2. A rules-based “constitutional AI” style reward model (Bai et al., 2022b) that combines the ROUGE score with a length penalty to provide greater interpretability of reward.

Tasks: We report results on two different tasks:

1. The **OpenAI TL;DR summarization task** with a training split of 116k human written instructions, and 96k human annotated language pairs³. The task is cleverly constructed from Reddit prompts, collecting a highly rated and a badly rated user summary and matching those to create an RLHF-amenable winning/losing pair (Syed et al., 2018). This task is evaluated using both the Gemma 2B Reward model and the rules based reward model, as summarization is amenable to the ROUGE-based rules model.

²The Gemma license from Google fosters broad use and innovation for its open language models, permitting commercial applications, redistribution, and modifications, alongside a focus on responsible AI development.

³This dataset is **licensed under the MIT license**, which permits use and research by commercial entities.

2. The **Anthropic Helpfulness/Harmlessness task** is a famous RLHF task that comprises pairs of AI-generated responses to various prompts, with human annotators indicating which response they prefer based on helpfulness and harmlessness criteria (Bai et al., 2022a)¹. It has a training split of 112k training preference pairs. Since training an LLM on concise ROUGE score is not natural for this task, we only present results using the H/H reward model.

Evaluation Metrics: ROUGE-LSum (Recall-Oriented Understudy for Gisting Evaluation—Longest Common Subsequence) is a variant of the ROUGE metric (Richemond et al., 2024), which is commonly used to evaluate the quality of summaries and generated text by comparing them to reference text. The Longest Common Subsequence is the longest sequence of words (or tokens) that appears in the same order in both the generated and reference texts. We use the combination of ROUGE-LSum and reward as a first pass filter to select the best runs from the hyperparameter sweep.

Calculating Win Rates: As a test for tasks trained on reward models, we calculate the win rates of every algorithm against the base Gemma2B SFT model using a larger Gemma7B model as a judge. We chose 200 prompts to field every pair of (RLHF Model i , SFT Model) based on a standard statistical power calculation which gave an 80% chance of rejecting the null hypothesis when it is false. We calculate the accuracy of the ratio to be 70%, meaning that there exists uncertainty in the results.

Hyperparameter Sweeps: We give a fixed compute budget to each RLHF algorithm, including identifying three key hyperparameters from each algorithm and sweeping over three values of each hyperparameter. For all algorithms, the first two hyperparameters are fixed to: (1) Policy Learning Rate (1e-7, 1e-6, 1e-5), (2) α , a parameter which weights the contribution of the KL term to the RLHF reward (0.3, 0.1, 0.05), and (3) a algorithm specific parameter: please see Appendix table 3. We perform a grid search for each of these parameters. Each hyperparameter choice was decided after extensive conversations with the authors of the various methods.

Computational Setup: We conduct all experiments on a TPU-based architectural setup, the details of which we must omit for privacy reasons. We use more than 30,000 TPU hours for our work.

5 Results and Discussion

In this section, we summarize our experimental findings by grouping the RLHF algorithms into related classes, evaluating the impact of reward model training, and examining the sensitivity of performance to hyperparameter settings. Our analysis is based on both win rates and reward curves collected across multiple test scenarios.

5.1 A Class of Reinforce-like Algorithms

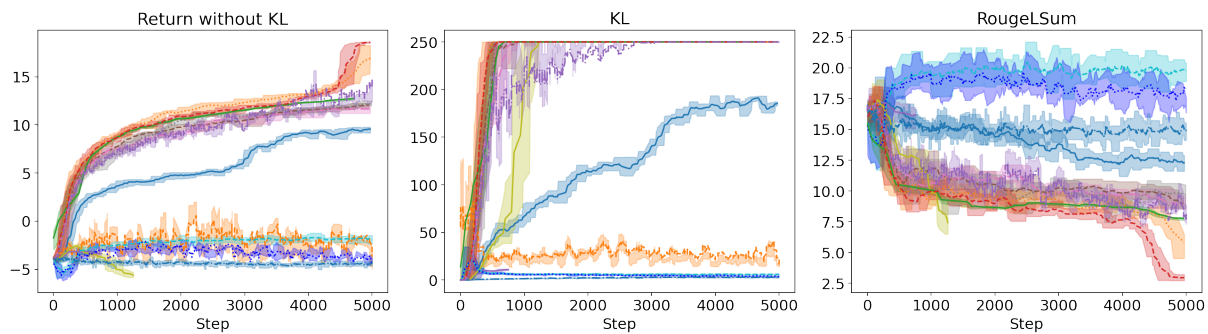
Our experiments reveal a distinct cluster of algorithms that share characteristics similar to those of the classical REINFORCE method. In particular, the algorithms **Reinforce with Baseline**, **DPO**, **IPO**, and **Best of N Cloning (BON)** consistently achieve high win rates. The reward curves for these methods indicate that simpler, more direct approaches—ones that avoid the complexity of PPO—can perform competitively. Moreover, the curves suggest that algorithms such as **Actor Critic** and **DRO** may also be considered within this class, as they exhibit comparable reward trends despite some differences in their update dynamics. Overall, these methods share a common philosophy: they directly optimize the expected reward with minimal additional regularization overhead compared to PPO, thereby reducing the risk of overly conservative updates.

5.2 Performance of Reward Models versus Direct Optimization

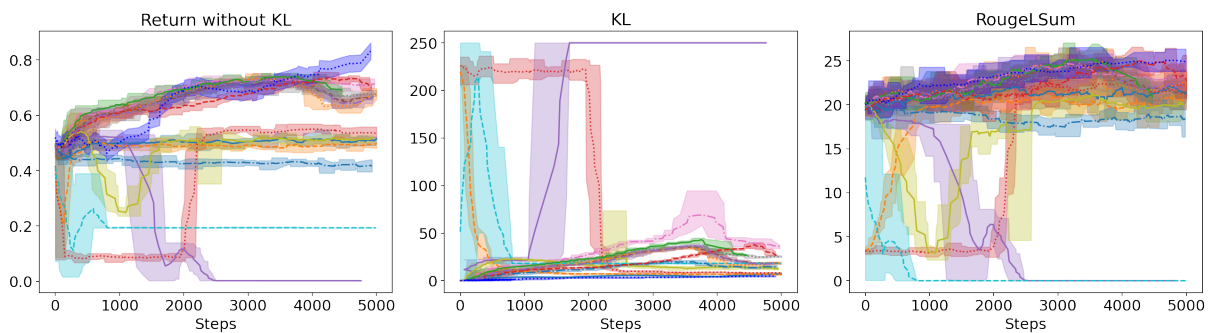
Our results indicate that training algorithms with a reward model becomes increasingly challenging when both the primary model and the reward model are scaled down. In these cases, the reward curves serve as a robust indicator of how effectively an algorithm can optimize within its environment, independent of potential vulnerabilities to reward hacking. We encourage readers to consider both the reward curves and win rates when assessing algorithm performance. In particular, while some methods may achieve high rewards, they sometimes suffer from mode collapse (e.g., repeated identical outputs), which is evident when comparing win rates with the reward curves. This observation underscores the importance of a balanced evaluation that accounts for both raw reward optimization and qualitative output diversity.

5.3 Hyperparameter Sensitivity and Optimization

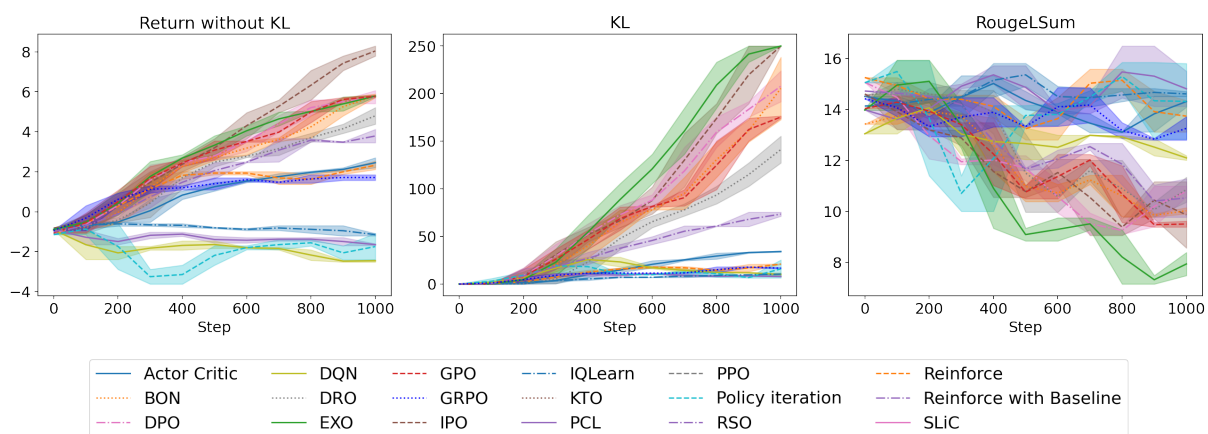
Analysis of the hyperparameter configurations shows that the best-performing models tend to use the smallest observed learning rates and temperatures. Please see Table 1 for a list of the best hyperparameters. This suggests that a conservative update strategy is beneficial, likely because smaller learning rates help restrain the model from shifting too rapidly. Additionally, our experiments indicate that hyperparameter optimization places less emphasis on SFT-regularization. In fact, the KL divergence coefficient and update frequencies tend to be on the lower end of the tested spectrum. These findings point to a continuum of strategies: one can either constrain the model’s update step to prevent abrupt changes or keep the model closely aligned with the pre-trained SFT policy—but imposing both restrictions simultaneously may overconstrain the model. This overconstraining is reminiscent of the issues observed with PPO’s trust region being inadequate for the LLM setting, as highlighted by Ahmadian et al. We suggest that future work should explore alternative regularization



(a) TL;DR Task trained with RM.



(b) TL;DR Task trained with rules-based reward.



(c) Harmlessness/Helpfulness task trained with RM.

Figure 1: Evaluation metrics of highest performing examples over a test set across various tasks.

methods to avoid the potential pitfalls of over-reliance on KL divergence.

5.4 Discussion and Future Directions

Taken together, our results highlight a promising class of Reinforce-like RLHF algorithms that achieve robust performance without the complexity of PPO. While reward these algorithms can effectively optimize in various environments, careful hyperparameter tuning is essential to avoid issues like mode collapse and overly conservative policy updates.

We provide the best hyperparameter configurations per algorithm in Table 1 in the Appendix.

We provide win rates for the models over an SFT base model in Table ???. Here we find rough agreement with the ROUGE ordering, finding that BON, GRPO, REINFORCE with baseline, DPO, and IPO are among the alternatives.

6 Conclusion

We return to the paper which founded the motivation for our study (Ahmadian et al., 2024). Our analysis confirms that several RLHF algorithms from the classical reinforcement learning era remain competitive on generic benchmark problems. In particular, our experiments indicate that algorithms such as REINFORCE with Baseline, DPO, IPO, and Best of N Cloning (BON) perform at the top of the evaluated set. Notably, the reward curves suggest that even methods like Actor Critic and DRO—which are often considered more complex—could be viable alternatives when evaluated under specific conditions. These algorithms share a common trait: they eschew the heavy regularization and trust-region constraints of PPO in favor of simpler, more direct policy updates. This observation not only reinforces the original critique in (Ahmadian et al., 2024) but also suggests that simplicity in policy optimization may confer advantages in terms of convergence speed and reward stability. While REINFORCE emerges as a strong contender, its performance, though promising, was not uniformly superior across all metrics.

We embarked on a comprehensive benchmarking approach to evaluate a wide range of algorithms for Reinforcement Learning from Human Feedback (RLHF). Our goal was to provide practitioners with clear guidance on the effectiveness of different strategies under realistic conditions.

7 Future Directions

We plan to extend our benchmarking suite by testing with different base SFT models like Gemini and Llama. Exploring alternative reward models and regularization strategies beyond standard KL divergence may provide further insights into optimizing RLHF performance.

8 Limitations

Our study has several **limitations**. First, our reward models, like many others, showed signs of incompleteness; mode collapse sometimes led to artificially high rewards, which we caution as **a risk to our work**. Second, the underlying model had only 2 billion parameters, which may not capture the dynamics of larger, production-scale reward models. Third, our exploration of hyperparameters was limited compared to the vast configuration space available. Focusing on fine-tuning a single algorithm might yield better performance than our broader survey.

Table 1: Best Hyperparameters for Each RLHF Algorithm Across Three Tasks

Algorithm	TL;DR Rules Based			TL;DR RM			H/H RM		
	LR	Parameter	α	LR	Parameter	α	LR	Parameter	α
Actor Critic	$1e^{-6}$	ξ : false	0.05	$1e^{-6}$	ξ : false	0.05	$1e^{-6}$	ξ : false	0.05
Reinforce	$1e^{-7}$	T_{out} : 1	0.05	$1e^{-7}$	T_{out} : 1	0.10	$1e^{-7}$	T_{out} : 1	0.05
R w/ Baseline	$1e^{-7}$	T_{out} : 1	0.05	$1e^{-7}$	T_{out} : 1	0.05	$1e^{-7}$	T_{out} : 1	0.05
PPO	$1e^{-7}$	T_{out} : 2	0.05	$1e^{-6}$	T_{out} : 2	0.10	$1e^{-7}$	T_{out} : 1	0.10
PCL	$5e^{-7}$	T_{out} : 1	0.30	$1e^{-7}$	T_{out} : 2	0.10	$1e^{-7}$	T_{out} : 2	0.10
IPO	$5e^{-7}$	β_{IPO} : 0	0.05	$1e^{-7}$	β_{IPO} : 0.5	0.10	$1e^{-6}$	β_{IPO} : 0	0.10
DPO	$1e^{-7}$	β_{IPO} : 0	0.05	$1e^{-7}$	β_{IPO} : 0	0.30	$1e^{-7}$	β_{IPO} : .50	0.30
DRO	$1e^{-7}$	β_{DRO} : 0	0.30	$1e^{-7}$	β_{DRO} : 0	0.10	$1e^{-7}$	β_{DRO} : 0	0.30
DQN	$5e^{-7}$	ω : 75	0.30	$5e^{-7}$	ω : 75	0.30	$1e^{-6}$	ω : 150	0.30
Policy Iteration	$1e^{-7}$	ω : 150	0.30	$1e^{-7}$	ω : 150	0.30	$1e^{-7}$	ω : 150	0.30
IQLearn	$5e^{-7}$	τ_{IQL} : 0.90	0.10	$5e^{-7}$	τ_{IQL} : 0.90	0.10	$1e^{-7}$	τ_{IQL} : 0.50	0.05
BON	$5e^{-7}$	N : 4	0.05	$1e^{-7}$	N : 4	0.05	$1e^{-7}$	N : 2	0.10
EXO	$1e^{-7}$	ω : 0.10	0.30	$1e^{-7}$	ω : 0.01	0.05	$1e^{-7}$	ω : 0.10	0.05
GPO	$5e^{-7}$	β_{GPO} : 0.50	0.30	$1e^{-7}$	β_{GPO} : 1	0.10	$1e^{-7}$	β_{GPO} : 2	0.10
RSO	$5e^{-7}$	N : 4	0.05	$1e^{-7}$	N : 4	0.30	$1e^{-7}$	N : 2	0.10
GRPO	$1e^{-7}$	g : 20	.1	$1e^{-7}$	g : 10	.3	$1e^{-7}$	g : 20	.1

Agent	2B reward model		7B reward model	
	TL;DR	H/H	TL;DR	H/H
AC	53%	55%	56%	59%
BON	64%	52%	68%	58%
DPO	62%	67%	69%	69%
DQN	32%	36%	35%	42%
DRO	58%	52%	61%	58%
EXO	48%	43%	51%	48%
GPO	61%	46%	64%	49%
GRPO	65%	63%	75%	68%
IPO	66%	48%	69%	51%
IQLearn	49%	51%	52%	54%
KTO	52%	56%	57%	59%
PCL	41%	42%	45%	45%
PI	45%	46%	48%	49%
PPO	54%	57%	58%	59%
REINFORCE	60%	62%	64%	67%
R w/Baseline	66%	64%	69%	69%
RSO	58%	61%	61%	66%
SLIC	48%	49%	48%	52%

Table 2: Win rates for RLFT agents over SFT, evaluated with 2B and 7B parameter reward models. Top three values in each column are bolded.

References

- Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*.
- Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.
- Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. 2021. Flow network based generative models for non-iterative diverse candidate generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27381–27394. Curran Associates, Inc.
- Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. Deep reinforcement learning from human preferences. *Preprint*, arXiv:1706.03741.
- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Model alignment as prospect theoretic optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 12634–12651. PMLR.
- Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. 2021. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34:4028–4039.
- Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29.
- Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. Unpacking DPO and PPO: Disentangling Best Practices for Learning from Preference Feedback. *Preprint*, arXiv:2406.09279.
- Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a Changing Climate: Enhancing LM Adaptation with Tulu 2. *Preprint*, arXiv:2311.10702.
- Haozhe Ji, Cheng Lu, Yilin Niu, Pei Ke, Hongning Wang, Jun Zhu, Jie Tang, and Minlie Huang. 2024. Towards efficient exact optimization of language model alignment. In *Forty-first International Conference on Machine Learning*.
- Wouter Kool, Herke van Hoof, and Max Welling. 2019. Buy 4 REINFORCE Samples, Get a Baseline for Free!
- Changhao Li, Yuchen Zhuang, Rushi Qiang, Haotian Sun, Hanjun Dai, Chao Zhang, and Bo Dai. 2024. Matryoshka: Learning to drive black-box llms with llms. *arXiv preprint arXiv:2410.20749*.
- Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. 2024. Statistical rejection sampling improves preference optimization. In *The Twelfth International Conference on Learning Representations*.
- Volodymyr Mnih. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA. PMLR.
- Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. 2017. Bridging the gap between value and policy based reinforcement learning. *Advances in neural information processing systems*, 30.
- 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*.
- 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.
- 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.

- Pierre Harvey Richemond, Yunhao Tang, Daniel Guo, Daniele Calandriello, Mohammad Gheshlaghi Azar, Rafael Rafailov, Bernardo Avila Pires, Eugene Tarassov, Lucas Spangher, Will Ellsworth, Aliaksei Severyn, Jonathan Mallinson, Lior Shani, Gil Shamir, Rishabh Joshi, Tianqi Liu, Remi Munos, and Bilal Piot. 2024. [Offline regularised reinforcement learning for large language models alignment](#). *Preprint*, arXiv:2405.19107.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal Policy Optimization Algorithms](#). *Preprint*, arXiv:1707.06347.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.
- Shahbaz Syed, Michael Voelske, Martin Potthast, and Benno Stein. 2018. [Dataset for generating TL;DR](#).
- Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Remi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Avila Pires, and Bilal Piot. 2024. [Generalized preference optimization: A unified approach to offline alignment](#). In *Forty-first International Conference on Machine Learning*.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikula, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. [Gemma: Open models based on gemini research and technology](#). *Preprint*, arXiv:2403.08295.
- Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. [How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources](#). *Preprint*, arXiv:2306.04751.
- Zhichao Wang, Bin Bi, Shiva Kumar Pentiyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, et al. 2024. A comprehensive survey of llm alignment techniques: Rlhf, rlaf, ppo, dpo and more. *arXiv preprint arXiv:2407.16216*.
- Ronald J. Williams. 1992a. [Simple statistical gradient-following algorithms for connectionist reinforcement learning](#). *Machine Learning*, 8(3):229–256.
- Ronald J Williams. 1992b. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.
- Markus Wulfmeier, Michael Bloesch, Nino Vieillard, Arun Ahuja, Jorg Bornschein, Sandy Huang, Artem Sokolov, Matt Barnes, Guillaume Desjardins, Alex Bewley, et al. 2024. Imitating language via scalable inverse reinforcement learning. *arXiv preprint arXiv:2409.01369*.
- Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

Appendix

8.1 Algorithm Specific Hyperparameters that were swept over

Please see Table 3 for a list of all of the algorithm specific hyperparameters that were swept over.

8.2 Best Hyperparameters

Please see Table 1 for a list of the best hyperparameters that we used.

8.3 Decoded Output

Here is an example of “good” decoded output from our models. NOTE, just as reminder, these are prompts that are prepared in the TL;DR Summarization dataset that was scraped from Reddit; *not* prompts that we created ourselves.

Table 3: Hyperparameter Ablations for RLHF Algorithms

Category	Algorithm	Symbol	Description	Ablated Values
Classical RL	AC	ξ	Boolean flag to include the KL term in the value model.	True, False
	DQN	ω	Target network update frequency (in steps).	75, 150, 250
	GRPO	g	Group batch size for advantage estimation.	10, 20, 50
	QLearn	τ_{IQL}	Regularization strength for Q-value estimation.	0.5, 0.9
	PI	ω	Frequency of target Q-distribution updates (in steps).	75, 150, 250
	PPO, PCL, REINFORCE	T_{out}	Decoder output temperature.	1.0, 1.5, 2.0
Direct Preference	EXO	ω_{anchor}	Weight of the anchor distribution to prevent policy drift.	0.01, 0.05, 0.1
	DPO	β_{DPO}	Regularization weight for alignment to the base SFT model.	0, 0.1, 0.5
	IPO	β_{IPO}	Regularization weight (analogous to DPO’s β).	0, 0.5, 1.0
	DRO	β_{DRO}	Regularization weight (analogous to DPO’s β).	0, 0.5, 1.0
	GPO	β_{GPO}	Regularization weight (analogous to DPO’s β).	Not Specified
	KTO	β_{KTO}	Starting value for KL divergence during the warmup period.	0, 0.5, 1.0
	SLiC	β_{SLiC}	Cross-entropy loss weight to stay close to the fine-tuned baseline.	0.5, 1.0, 2.0
Sampling	BON	n	Number of candidate samples generated for selection.	2, 4, 8
	RSO	k	Number of samples rejected in each round.	2, 4, 8

Prompt 1:

Summarize the following Text:

I’m a seventeen year old going into grade twelve, and I don’t know anything about my future... I really don’t know where I’m going career-wise. That’s one thing; I also have a girlfriend of 2+ years and we have yet to kiss, I haven’t gotten even my learner’s license yet, and throughout my life I have refused to use any drugs/alcohol to this point.

Now, I’ve been really satisfied with all this untill today, when an old friend and I got together and met up with our third friend (who turns out to now smoke weed/is now on a two month light-every-day streak). This lead to a messed up day of following them to dealers and near fights, small town gangs and free ki-ji kittens, all leading to getting some friends together to light up.

I did nothing but follow and listen to them to-day, and what an average day for any one of them amazed me. I left and thought about my current life, and now I’m here on askreddit while laying in bed.

Basically, I want to know how normal I am.

Model output: *I’m a seventeen year old going into grade twelve, and idk where future going career related. Been with my amazing girlfriend 2 yrs, never used drugs/alcohol. Encountered some friends to-day who smoked weed (everyday for 2 months straight). Left and thought about current life while now laid in bed. How normal am I? REF I don’t do anything in my life, and I spent a day following friends function through their weed related day to day activities. This ‘normal’ teenager life contradicts mine a lot. Am I still a normal person?*

Prompt 2:

Summarize the following Text:

I am the legal guardian of my cousin [10/M]. We were both abused by my uncle/his father, who is currently in prison far away from us, and his mother signed over her rights to me. I've been Cousin's guardian for two years now, and we're doing well.

Recently, our grandmother (uncle's mother) has tried to get into contact with us again. I cut off a huge swathe of my family during the trial – essentially everyone who was not explicitly on our side – and my grandmother was part of that. Grandmother wants to make amends for her "ambivalence," and that she is not talking to my uncle.

Cousin loves Grandmother and was upset when I cut contact with her, even when I explained the situation, and I want to do right by him. But I do not trust her at all. If my uncle even finds out a smidge of information about Cousin, I will not be responsible for my actions. We do not live in a state with grandparent visitation rights, so I am not worried about legal repercussions, but I don't know how to approach anyone on that side of the family.

Model output: *I am the legal guardian of my cousin [10/M]. We were BOTH abused by my uncle/his father, and I cut off HUGE swathe of the family during trial. Recently grandmother wants to contact us again to make amends for her "ambivalence." Do NOT trust her at all. Do not know how to proceed with anyone on that side. No legal repercussions but how to proceed? REF Gained legal guardianship of my cousin because my uncle was abusive. My grandmother, who cousin loves, wants to reconnect but I do not trust her.*

Prompt 3:

Summarize the following Text:

I was seeing this guy for a month and it started off really strong. We would try to see each other all the time, we would talk on the phone every night for hours, we would text all day.

When we first met, there was a lot of chemistry and we were very upfront about what we wanted - both of us wanted relationships as an endgoal.

I was hesitant at first about being so lovey-dovey so quick, but he seemed so genuine when he told me how he felt, so I kind of dove in right behind him.

After our first couple of dates, he went on vacation and things were still amazing. We would talk all the time and feelings only grew. He came back and we had another date (and sex) and things fizzled.

After that, he flaked on our next planned date, but made sure to plan another one in its place. We did go out, had sex again, but things didn't change. It still felt different. I told him if he had lost interest, it was fine - it happens. But he still said he wanted to talk to and see me, but it was just work and distance (he lives an hour away) that was making it hard.

We tried make plans one more time for him to come down. He said "he will let me know," and didn't end up texting me all that day. I broke things off because I've played this game before - someone who wants to be with me, would be. This guy did not. So I pretty much ended things, saying it was just disappointing but for the best, and he agreed.

Days later, he texted me and apologized. Said the more he saw me, it didn't feel "natural for us to be together" and work just made it harder. Also told me it wasn't anyone else in the picture either. We kind of got into a back and forth and he admit, he just lost interest. He pretty much said he strung me along because "he just wanted to like me back" which is confusing to me, because in the beginning all he could talk about was how much he liked me.

Model output: *Was seeing this guy for 1 mo. Things were heating up fast. He simply STOPS talking/seeing after a planned date initially. Says he just lost interest due to distance and previous experiences. He still texts though. REF A guy I saw a lot of potential with fizzled out on me because apparently he didn't like me back, and I can't stop thinking about it and what went wrong.*

An example of bad model behavior:

Prompt 1 Summarize the following Text:

So this is a bit of a odd question, and a bit of a moral one as well.

A local company named ARC is in the Nashville area. They are similar to goodwill in the respect that they will pick up your clothes for a tax break.

The branch of the group here prints off yellow bags with their number here on the bag to call and schedule pick ups of the bags. (They employ mentally handicap members of society to do most of the labor of the company.)

