Model Alignment as Prospect Theoretic Optimization

Kawin Ethayarajh [1] **Winnie Xu** [2] **Niklas Muennighoff** [2] **Dan Jurafsky** [1] **Douwe Kiela** [1 2]

Abstract

Kahneman & Tversky's *prospect theory* tells us that humans perceive random variables in a biased but well-defined manner (1992); for example, humans are famously loss-averse. We show that objectives for aligning LLMs with human feedback implicitly incorporate many of these biases—the success of these objectives (e.g., DPO) over cross-entropy minimization can partly be ascribed to them belonging to a family of loss functions that we call *human-aware losses* (HALOs). However, the utility functions these methods attribute to humans still differ from those in the prospect theory literature. Using a Kahneman-Tversky model of human utility, we propose a HALO that directly maximizes the utility of generations instead of maximizing the log-likelihood of preferences, as current methods do. We call this approach KTO, and it matches or exceeds the performance of preference-based methods at scales from 1B to 30B, despite only learning from a binary signal of whether an output is desirable. More broadly, our work suggests that there is no one HALO that is universally superior; the best loss depends on the inductive biases most appropriate for a given setting, an oft-overlooked consideration.

1. Introduction

Aligning generative models with human feedback has been successfully used to make generations more helpful, factual, and ethical, among other desiderata (Ouyang et al., 2022; Tian et al., 2023). For LLMs, alignment methods such as RLHF and DPO have consistently proven to be more beneficial than doing supervised finetuning (SFT) alone. However, human feedback is often discussed only in the context of preferences (e.g., output $y_w \succ y_l$ for input x), even though it can take many forms (e.g., approval/disapproval of y given x). This is because preferences, despite being a kind of data that is relatively scarce and expensive to collect in practice (Casper et al., 2023), are required by the alignment methods shown to work best—RLHF (Christiano et al., 2017) and DPO (Rafailov et al., 2023).

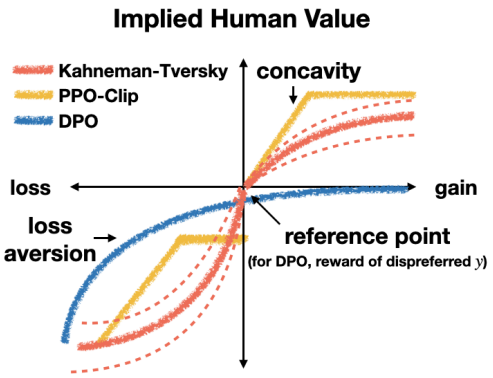

Figure 1. The utility that a human gets from the outcome of a random variable, as implied by different human-aware losses (HALOs). Notice that the implied value functions share properties such as loss aversion with the canonical human value function in prospect theory (Tversky & Kahneman, 1992).

To understand why these methods work so well, and whether feedback needs to be in preference form, we frame alignment through the lens of *prospect theory* (Kahneman & Tversky, 1979; Tversky & Kahneman, 1992). Prospect theory explains why humans make decisions about uncertain events that do not maximize their expected value. It formalizes how humans perceive random variables in a biased but well-defined manner; for example, relative to some reference point, humans are more sensitive to losses than gains, a property called *loss aversion*. We show that popular alignment methods such as DPO and PPO-Clip (Schulman et al., 2017) implicitly model some of these biases, helping explain their success independently of the data used (§3.2). We then propose a more general class of such loss functions called *human-aware losses* (HALOs).[1]

[1]Stanford University (first author was an intern at Contextual AI) [2]Contextual AI. Correspondence to: Kawin Ethayarajh <kawin@stanford.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1]We use the term *human-aware* to draw an analogy with how *hardware-aware* methods benefit from being designed around hardware limitations (Dao et al., 2022), not to claim that a simple loss function is fully aware of human behavior.

Although it is impossible to say that HALOs are categorically better than non-HALOs, we find that among existing methods, those that meet the definition of a HALO work better than those that do not (§3.3). We find that DPO performance can even be matched at most scales by running an offline PPO variant on dummy +1/-1 rewards, suggesting that preference data might not be needed if the inductive bias in the loss function is good enough. However, despite the surprising success of this simple baseline, it significantly lags behind DPO at the 30B LLM scale and suffers from hyperparameter sensitivity, making it difficult to use.

Taking a more principled approach, we derive a HALO using the model of human utility that Kahneman & Tversky proposed to describe how humans make decisions about uncertain monetary outcomes (Tversky & Kahneman, 1992). This approach, which we call Kahneman-Tversky Optimization (KTO), directly maximizes the utility of generations instead of maximizing the log-likelihood of preferences, as most current methods do (§4.1). KTO only requires a binary signal of whether an output is desirable or undesirable for an input. This data is more abundant, cheaper, and faster to collect in the real world, making it easier to scale alignment in production and rapidly iterate on models. We find that:

- KTO matches or exceeds DPO performance at scales from 1B to 30B parameters (§4.2).[2] That is, breaking up n preferences meant for DPO into $2n$ examples for KTO can yield better LLM generations, as determined by closed-ended tasks such as mathematical reasoning and open-ended judgments from humans and `GPT-4`.
- KTO can handle extreme data imbalances, matching DPO performance while using up to 90% fewer desirable examples (i.e., examples of good generations). Its success cannot be ascribed to the alignment data being sourced from a preference dataset.
- When the pretrained model is sufficiently good, one can skip supervised finetuning (SFT) and go straight to KTO without a loss in generation quality, whereas SFT is always needed for best results with DPO.

The intent behind KTO was that even if the model learns from a weaker signal, we could compensate with the higher volume of data that could be accessed in practice; the fact that KTO can match and even outperform DPO on the same data is thus surprising. We conclude by discussing some theoretical explanations for this phenomenon (§4.3). Despite the success of KTO in our experiments, our work ultimately suggests that there is no one HALO that is universally superior; the best HALO depends on the inductive biases appropriate for a given setting, and this choice should be made deliberately instead of defaulting to any one loss.

[2]Our code is available on Github; models are on Huggingface.

2. Background

For a detailed survey, please see Appendix A. In brief, LLMs are traditionally trained in three stages (Ouyang et al., 2022):

Pretraining Given a large corpus, train the model to maximize the log-likelihood of the next token conditioned on the preceding text. Let π_0 denote the pretrained model.

Supervised Finetuning (SFT) Finetune the model to predict the next token on data that is more relevant to the downstream task. Often, such data will comprise instructions and an appropriate response (i.e., instruction finetuning). Let π_{ref} denote the finetuned model.

RLHF Given a dataset $\mathcal{D}$ of preferences (x, y_w, y_l)—where x is an input, y_w, y_l are the preferred and dispreferred outputs (i.e., $y_w \succ y_l$ for x), and r^* is the "true" reward function underlying the preferences—it is first assumed that the probability that y_w is preferred to y_l can be captured with a specific function class, typically a Bradley-Terry model (Bradley & Terry, 1952). Where σ is the logistic function:

$$p^*(y_w \succ y_l|x) = \sigma(r^*(x, y_w) - r^*(x, y_l)) \tag{1}$$

Since getting the true reward from a human would be intractably expensive, a reward model r_ϕ learns to serve as a proxy, done by minimizing the negative log-likelihood of the human preference data:

$$\mathcal{L}_R(r_\phi) = \mathbb{E}_{x,y_w,y_l\sim D}[-\log\sigma(r_\phi(x, y_w) - r_\phi(x, y_l))]$$

But solely maximizing the reward might come at the expense of desiderata such as generating grammatical text. To avoid this, a KL divergence penalty is introduced to restrict how far the language model can drift from π_{ref}. Where π_θ is the model we are optimizing, the optimal model π^* is that which maximizes

$$\mathbb{E}_{x\in D, y\in\pi_\theta}[r_\phi(x, y)] \; - \beta D_{\text{KL}}(\pi_\theta(y|x)\|\pi_{\text{ref}}(y|x)) \tag{2}$$

where $\beta > 0$ is a hyperparameter. Since this objective is not differentiable, we need to use an RL algorithm like PPO (Schulman et al., 2017).

However, RLHF is often slow (largely because of having to sample generations) and quite unstable in practice (especially in a distributed setting). For this reason, recent work has focused on designing closed-form losses that maximize the margin between the preferred and dispreferred generations. In particular, Direct Preference Optimization (DPO) (Rafailov et al., 2023) has emerged as a popular alternative since it allows the same optimal policy as in RLHF to be recovered under certain conditions:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta, \pi_{\text{ref}}) = \mathbb{E}_{x,y_w,y_l\sim D} \left[-\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right] \tag{3}$$

3. A Prospect Theoretic View of Alignment

To understand why alignment methods work so well, we now frame them through the lens of *prospect theory* (Tversky & Kahneman, 1992). Prospect theory explains why, when faced with an uncertain event, humans make decisions that do not maximize their expected value. For example, because humans are loss-averse, given a gamble that returns \$100 with 80% probability and \$0 with 20% probability, a person might accept \$60 to avoid the gamble, despite their *certainty equivalent* of \$60 being less than the expected value of \$80.

3.1. Prospect Theory

In prospect theory, human utility depends on a *value function* and a *weighting function*:[3]

Definition 3.1. A *value function* $v : \mathcal{Z} \to \mathbb{R}$ maps an outcome z, relative to some reference point z_0, to its perceived (or subjective) value. For example, these functions capture the fact that humans tend to be more sensitive to relative losses than relative gains of the same magnitude.

Definition 3.2. A *weighting function* ω is the derivative of a *capacity function* that maps cumulative probabilities to perceived cumulative probabilities. These functions capture, for example, the fact that humans tend to overestimate the chance of rare events. Let ω_z denote the weight placed on outcome z.

Definition 3.3. The *utility of a random variable* Z is a function of its outcomes: $u(Z) \triangleq \sum_{z \in Z} \omega_z v(z - z_0)$.

However, because humans do not see the full probability distribution of an LLM, weighting functions are not salient to this discussion; we will focus only on value functions. Using experiments that presented real humans with monetary gambles and asked for their certainty equivalent, Tversky & Kahneman (1992) proposed the following functional form for human value:

$$v(z; \lambda, \alpha, z_0) = \begin{cases} (z - z_0)^\alpha & \text{if } z \geq z_0 \\ -\lambda(z_0 - z)^\alpha & \text{if } z < z_0 \end{cases} \tag{4}$$

where the median value of hyperparameter $\alpha = 0.88$ and $\lambda = 2.25$ across individuals. α controls the curvature of the function, which reflects risk aversion; λ controls its steepness, which reflects loss aversion. While the shape of the median Kahneman-Tversky value function is illustrated in Figure 1, it should be noted that it varies across individuals (Tversky & Kahneman, 1992). There are also other functional forms for the value function that have been proposed in later work (Gurevich et al., 2009). The salient qualities of a value function are: the existence of a reference point that is used to get the *relative* gain or loss; concavity in relative gains (i.e., diminishing sensitivity away from z_0); and loss aversion (i.e., greater sensitivity to losses).

[3] *Cumulative prospect theory* is the full name of the expanded theory we dicuss here (Tversky & Kahneman, 1992).

3.2. HALOs

Informally, HALOs are loss functions that reflect the model of human decision-making in Tversky & Kahneman (1992). Formally,

Definition 3.4 (HALOs). Let θ denote the trainable parameters of the model $\pi_\theta : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ being aligned, π_{ref} the reference model, $l : \mathcal{Y} \to \mathbb{R}^+$ a normalizing factor, and $r_\theta(x, y) = l(y) \log[\pi_\theta(y|x)/\pi_{\text{ref}}(y|x)]$ the implied reward. Then $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a *human-aware loss* if there exists a reference point distribution $Q(X', Y'|x, y)$ over $\mathcal{X} \times \mathcal{Y}$ and a value function $v_f : \mathbb{R} \to \mathbb{R}$ that is non-decreasing everywhere and concave in $(0, \infty)$ such that f is linear in

$$v_f(r_\theta(x, y) - \mathbb{E}_Q[r_\theta(x', y')]) \tag{5}$$

In a classic prospect theory experiment, r_θ would be the dollar amount assigned to each outcome; here, r_θ is measured in nats, as the decrease in conditional surprisal when going from π_{ref} to π_θ, normalized according to l. This follows naturally from the next-token prediction objective used to pretrain and finetune LLMs. As π_θ is aligned, we would expect r_θ to grow increasingly positive for desirable outputs and increasingly negative for undesirable outputs.

Another perspective on the reward comes from the RLHF objective in (2). The policy that maximizes this objective has a closed-form expression (Peng et al., 2019; Peters & Schaal, 2007); where $Z(x)$ is the partition function:

$$\pi^*(y|x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r^*(x, y)\right)$$

Letting $l(\cdot) = \beta$, we get

$$r_{\theta^*}(x, y) = r^*(x, y) - \beta \log Z(x) \tag{6}$$

Under θ^*, the HALO-defined reward is just the optimal reward shifted by an input-specific term, meaning that r_{θ^*} is in the same equivalence class as r^* and would also induce the optimal policy π^* (Lemma 1, Rafailov et al. (2023)).

The reference point in a HALO is the expected reward from the human's perspective, where $Q(X', Y'|x, y)$ describes the examples that are perceived as a baseline. We require that the value function be concave in gains but not necessarily convex in losses—unlike the canonical Kahneman-Tversky value function—because a minority of individuals are risk-averse in both the gain and loss regime, resulting in concavity everywhere (Kahneman & Tversky, 1979).

Theorem 3.5. *DPO and PPO-Clip are human-aware losses.*

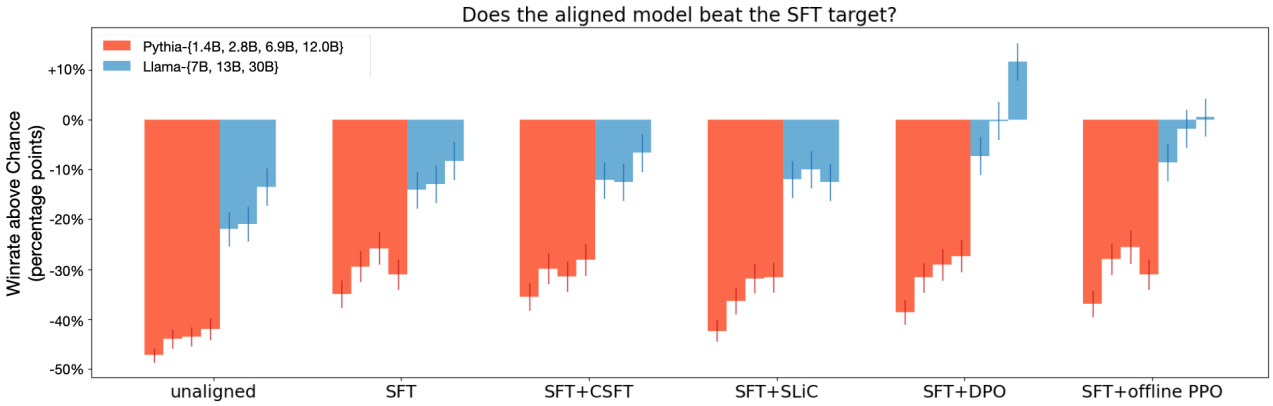

Figure 2. HALOs (DPO, offline PPO variant) outperform non-HALOs (SLiC, CSFT), as measured by the `GPT-4-0613`-judged winrate of the aligned model's generations against the outputs that would have been used for SFT. For better visualization, the y-axis here plots the winrate above chance (i.e., the winrate – 50%). The difference between methods is only significant ($p < 0.05$) at 13B+ parameters, and only the HALO-aligned Llama-{13B, 30B} models are able to match the baseline and yield a winrate at or above chance.

The proof is deferred to Appendix B. In Figure 1, we can see this more intuitively by plotting the value function for each loss (i.e., the implied human utility).

3.3. Does being a HALO matter?

A natural question is whether modeling human biases has practical benefits. This is difficult to answer, since both HALOs and non-HALOs are diverse function classes, but we attempt to do so by comparing popular methods that qualify as a HALO with those that do not:

1. **CSFT**: Conditional SFT is a simple alignment method where a control token is prepended to the output during training; then, at inference, the control token corresponding to desirable generations (e.g., `<|good|>`) is appended to the input to induce good generations (Korbak et al., 2023). This is not a HALO.

2. **SLiC**: Sequence Likelihood Calibration (Zhao et al., 2023), which combines a max-margin loss for preferences with a language modeling loss, is not a HALO:

$$\mathcal{L}_{\text{SLiC}}(\pi_\theta, \pi_{\text{ref}}) = \mathcal{L}_{\text{cal}}(\pi_\theta) + \lambda_{\text{reg}} L_{\text{reg}}(\pi_\theta)$$

$$\mathcal{L}_{\text{cal}} = \mathbb{E}_{x,y_w,y_l \sim D}\left[\max\left(0, \delta - \log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}\right)\right]$$

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{x \sim D, y \sim \pi_{\text{ref}}(x)}[-\log \pi_\theta(y|x)]$$

3. **DPO**: DPO is a HALO (Theorem 3.5).

4. **PPO (offline)**: The standard RLHF objective in (2) is typically optimized with PPO-Clip, which works by "clipping" how far π_θ can drift from the version π_{old} at the previous step:

$$\mathcal{L}_{\text{PPO (offline)}} = -\mathbb{E}_{x,y,t \sim D}[\min(q_\theta A(x{:}y_{<t}, y_t), \text{clip}(q_\theta, 1-\epsilon, 1+\epsilon) A(x{:}y_{<t}, y_t))]$$

where $q_\theta = \frac{\pi_\theta(y_t|x{:}y_{<t})}{\pi_{\text{old}}(y_t|x{:}y_{<t})}$ and $A(x{:}y_{<t}, y_t)$ is the per-token advantage (i.e., the surplus benefit from producing a given token in a given state).

PPO is an online algorithm—generations are sampled from the current model, judged by a reward model, and then used to update the current version. However, for a fair comparison with offline methods, we consider a variant of PPO that is only fed offline data. Because RLHF is also quite unstable in a distributed setting, we never update π_{old} and keep it as π_{ref}, instead clipping less conservatively than we traditionally would. Baheti et al. (2023) found that these changes, along with treating the entire output sequence as a single action, greatly improves stability. However, since RLHF traditionally uses token-level advantages, we omit the third change and only preserve the first two. The PPO-Clip loss itself is left unchanged and is therefore a HALO (Theorem 3.5).

Calling this method PPO is somewhat imprecise, because it is offline and takes only one step, but to avoid introducing too many new terms, we will call this *PPO (offline)*. Instead of using learned rewards, we simplify even further and use dummy +1/-1 rewards[4] for y_w and y_l instead. Further details on the implementation of this method can be found in Appendix C.

Experimental Setup We compare these baselines on two model families, Pythia-{1.4B, 2.8B, 6.9B, 12B} (Biderman et al., 2023) and Llama-{7B, 13B, 30B} (Touvron et al., 2023). This permits us to see how LLM alignment scales within a model family (Llama-2 lacks a 30B model, hence our use of Llama). Later experiments (§4.2) are done on

[4]Note that +1/-1 do not refer to the HALO-implied rewards, but rather the rewards used for the advantage $A(x{:}y_{<t}, y_t)$ calculation.

Mistral-7B and its derivatives (Jiang et al., 2023). The models were trained on a combination of Anthropic-HH (Ganguli et al., 2022), OpenAssistant (Köpf et al., 2023), and SHP (Ethayarajh et al., 2022).

All models are aligned under identical settings on the same data (e.g., same effective batch size, same optimizer, etc.), save for hyperparameters unique to them. Similar to Rafailov et al. (2023), the target sequences for SFT are a subset of the generations used to subsequently align the model. We then use `GPT-4-0613` to judge whether the aligned model's response is better than the SFT target for a given test input with respect to helpfulness, harmlessness, and conciseness, a now standard practice (Zheng et al., 2023; Li et al., 2023).[5] Note that while the SFT target is considered a desirable output for x, it is by no means the *best* output, meaning that it can be improved upon by an aligned model.

In Figure 2, we see the results of this analysis:

- **HALOs either match or outperform non-HALOs at every scale**, though the gap is only significant ($p < 0.05$) at 13B+ model sizes after correcting for multiple comparisons (Holm, 1979). In fact, only the HALO-aligned Llama-{13B, 30B} models match or exceed a win rate of 50% (i.e., are able to match or exceed the generation quality of the SFT targets in the test data).
- **Up to a scale of 7B parameters, alignment provides virtually no gains over SFT alone.** However, it is worth noting that if the base models were more performant, or if the SFT data distribution were less similar to the preference data, then the gains from the alignment stage would ostensibly be greater.
- **Despite only using dummy +1/-1 rewards, our offline PPO variant performs as well as DPO** for all models except Llama-30B. This challenges conventional wisdom, which places heavy emphasis on reward learning (Casper et al., 2023), and suggests that even the simplest rewards can prove useful when used in a loss function that has the right inductive bias. Despite its success, our offline PPO baseline still suffers from hyperparameter sensitivity and training instability, albeit not to the same extent as traditional RLHF.

4. Kahneman-Tversky Optimization

The surprising success of offline PPO with dummy +1/-1 rewards suggests that—with the right inductive biases—a binary signal of good/bad generations may be sufficient to reach DPO-level performance, even if the offline PPO approach itself was unable to do so past a certain scale (§3.3). Taking a more principled approach, we now derive a HALO using the Kahneman-Tversky model of human utility, which allows us to directly optimize for utility instead of maximizing the log-likelihood of preferences.

4.1. Derivation

The canonical Kahneman-Tversky value function (4) suffers from numerical instability during optimization due to the exponent a, so we replace it with the logistic function σ, which is also concave in gains and convex in losses.

To control the degree of risk aversion, we introduce a hyperparameter $\beta \in \mathbb{R}^+$ as part of the value function. The greater β is, the more quickly the value saturates, meaning the human is simultaneously more risk-averse in gains and more risk-seeking in losses. In practice, this has a similar effect as β in the DPO loss, which controls how far π_θ drifts from π_{ref}, though we introduce it here explicitly to control risk aversion; in DPO, it carries over from the KL constraint in the RLHF objective (2) and is part of the reward.

We replace the loss aversion coefficient λ in the original Kahneman-Tversky value function (4) with a weighting function $w : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$. By default, $w(x, y) \in \{\lambda_D, \lambda_U\}$, where λ_D and λ_U are hyperparameters for desirable and undesirable losses respectively; more complex schemes could also be used for importance sampling.

Rather than having just one dispreferred generation serve as the reference point z_0, we assume that humans judge the quality of (x, y) in relation to all possible input-output pairs. We break down $Q(X', Y'|x, y)$ into a uniform prior over inputs in the dataset and let $Q(Y'|X')$ be π_θ. The reference point simplifies to the expected KL divergence $\text{KL}(\pi_\theta(y'|x')\|\pi_{\text{ref}}(y'|x'))$ w.r.t. $Q(X')$. However, as we discuss further below, we end up taking a biased estimate of this term in practice.

Where λ_y denotes $\lambda_D(\lambda_U)$ when y is desirable(undesirable) respectively, the default KTO loss is:[6]

$$L_{\text{KTO}}(\pi_\theta, \pi_{\text{ref}}) = \mathbb{E}_{x,y\sim D}[\lambda_y - v(x, y)] \tag{7}$$

where

$$r_\theta(x, y) = \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$$

$$z_0 = \mathbb{E}_{x'\sim D}\left[\text{KL}(\pi_\theta(y'|x')\|\pi_{\text{ref}}(y'|x'))\right]$$

$$v(x, y) = \begin{cases} \lambda_D \sigma(\beta(r_\theta(x, y) - z_0)) \text{ if } y \sim y_{\text{desirable}}|x \\ \lambda_U \sigma(\beta(z_0 - r_\theta(x, y))) \text{ if } y \sim y_{\text{undesirable}}|x \end{cases}$$

For more stable training, we do not backpropagate through

[5] We validate that GPT-4 judgments concur with human judgments in Appendix D.

[6] An earlier draft separated out λ_D, λ_U into a function w. For an easier comparison with (4), they—along with β—have been moved into the value function itself. Note that λ_y exists solely to make the loss non-negative and can be removed without consequence.

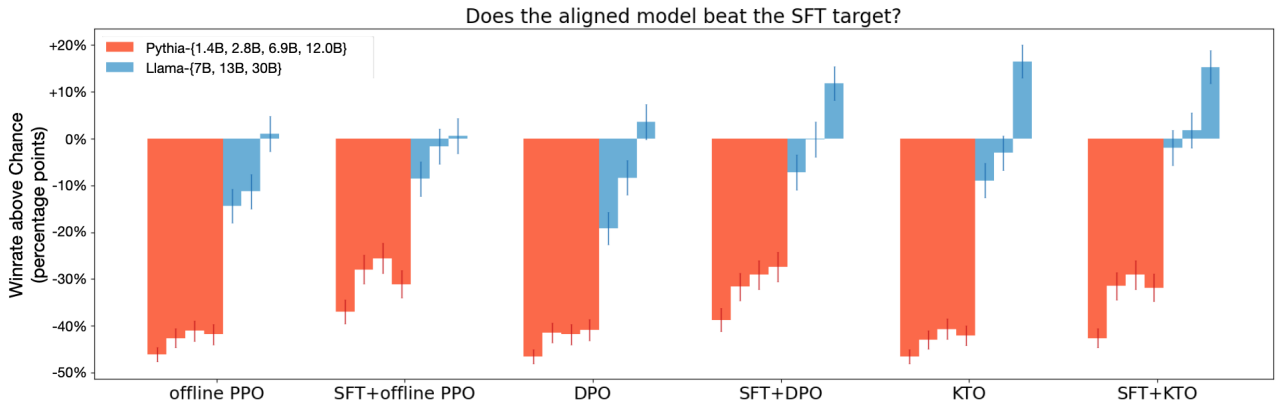

Figure 3. KTO is as good or better than DPO at all scales, as measured by the `GPT-4-0613`-judged winrate of the aligned model's generations against the outputs that would have been used for SFT. In fact, for the Llama models, KTO alone matches the performance of SFT+DPO and is significantly better than DPO alone. Error bars denote a 90% binomial confidence interval.

z_0; it exists purely to control the loss saturation.

Intuitively, KTO works as follows: if the model increases the reward of a desirable example in a blunt manner, then the KL penalty also rises and no progress is made. This forces the model to learn exactly what makes an output desirable, so that the reward can be increased while keeping the KL term flat (or even decreasing it). The argument works in the other direction as well, though the non-negativity of the KL term allows faster saturation in the loss regime.

KL Estimate In practice, estimating z_0 as it is defined above is impractical because sampling from π_θ is slow and humans do not perceive the full distribution induced by π_θ. Taking the expected reward w.r.t. a human's actual belief would give you a biased estimate of the KL term, since they have an "availability heuristic" that would overweight outputs for which they have already given feedback, regardless of whether those outputs are a good continuation of x (Tversky & Kahneman, 1973). To better simulate the human-perceived reference point, we create m pairs (x_i, y_j) where y_j is in the same m-sized batch of offline data as x_i but $i \neq j$, and estimate z_0 as:

$$\hat{z}_0 = \max\left(0, \frac{1}{m}\sum_{i \neq j} \log \frac{\pi_\theta(y_j|x_i)}{\pi_{\text{ref}}(y_j|x_i)}\right)$$

This estimate is a biased one, but this is desirable because it gets us closer to how humans would construct their subjective reference point.

Data If the alignment data is naturally binary, every positive example can be assumed to be drawn from $y_{\text{desirable}}|x$ and every negative example from $y_{\text{undesirable}}|x$. However, the canonical feedback datasets in academic research (HH, SHP, OASST) are in preference format, since the methods that have worked best up until now are preference-based. In our experiments, we converted preference data $y_w \succ y_l$ by assuming that y_w is drawn from the desirable distribution and y_l from the undesirable one. This is a naive assumption, made for the sake of simplicity, and a more complex deconstruction of preferences into binary feedback would likely yield better results, which we leave for future work. To show that KTO can be used with non-preference data, we also subsample exactly one y per x for some experiments (denoted one-y-per-x), removing any trace of preferences at the cost of reducing the data volume.

If human feedback is in the form of scores or ratings, the simplest means of incorporating it into KTO is to construct a weighting function such that high-magnitude data is weighed more and that examples with scores above(below) some threshold are desirable(undesirable). It is also possible to construct score-based HALOs from first principles, but we leave the design of such losses to future work.

Hyperparameters The default weighting function w controls the degree of loss aversion with two hyperparameters λ_D, λ_U that are both set to 1. In a class-imbalanced setting, where n_D and n_U refer to the number of desirable and undesirable examples respectively, we find that it is generally best to set λ_D, λ_U such that

$$\frac{\lambda_D n_D}{\lambda_U n_U} \in \left[1, \frac{4}{3}\right] \tag{8}$$

where one of the two should be set to 1 and the ratio is controlled by changing the other. For example, if there were a 1:10 ratio of desirable:undesirable examples, we would set $\lambda_U = 1, \lambda_D \in [10, 10.33]$. The interval $[1, 4/3]$ was determined empirically and suggests a value function that is more gain-sensitive, in contrast to the classic Kahneman-Tversky value function (4), which is more loss-sensitive. In tasks

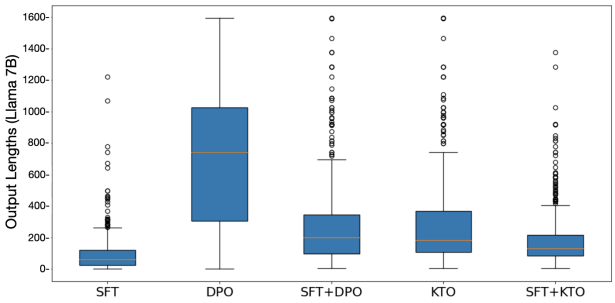

Figure 4. Without doing SFT first, DPO-aligned models tend to ramble and hallucinate entire conversations. KTO does not suffer from this issue.

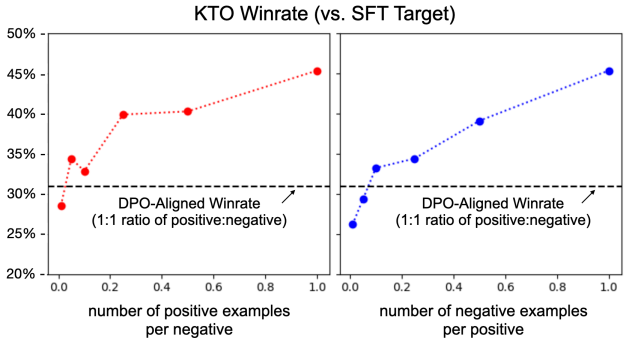

Figure 5. A KTO-aligned Llama-7B model can match or exceed the performance of its DPO-aligned counterpart while aligned on a smaller and highly imbalanced version of the same dataset, even with as few as 0.1 positive/desirable examples for every negative/undesirable one.

where minimizing the downside is more important, like toxicity prevention, setting λ_D, λ_U such that $\lambda_D n_D < \lambda_U n_U$ may work better. This is especially helpful if the positive KTO examples are instances of less toxic outputs that should still be avoided, as would be the case if a preference dataset were simply split up to create the KTO data. $w(x, y)$ could also serve as importance weights, which is helpful when the offline dataset D does not match the desired feedback distribution. We leave more complex weighting schemes as a direction for future work.

By default, we set $\beta = 0.1$ in our experiments. If we want the loss to saturate more quickly—i.e., be more risk-averse in gains and risk-seeking in losses—we would use lower β (and vice-versa for higher β).

4.2. Experiments

We subject the models to: (1) winrate experiments following §3.3, where—for some test inputs—`GPT-4-0613` is used to judge the aligned model's generation against the SFT target; (2) generative benchmarks such as MMLU (0-shot) (Hendrycks et al., 2021), GSM8K (8-shot, chain-of-thought) (Cobbe et al., 2021), HumanEval (0-shot) (Chen et al., 2021), and BigBench-Hard (3-shot chain-of-thought) (Srivastava et al., 2022). The experimental configuration is identical across methods, save for the data-imbalanced experiments, where we set λ_D, λ_U for KTO according to (8).

KTO $\geq$ DPO As seen in Figure 3, in the winrate evaluation, SFT+KTO is competitive with SFT+DPO at scales from 1B to 30B, despite the model learning from a weaker signal. KTO alone is better than DPO alone for the Llama-{7B, 13B, 30B} models, and this gap is significant ($p < 0.01$) at 7B and 30B even after correcting for multiple comparisons (Holm, 1979). Among the Pythia models, there is no significant difference between the two, suggesting that a minimum model capacity is needed for these differences to emerge. KTO also fares better than DPO and other baselines on generative benchmarks (Table 1). This is most pronounced for certain tasks: on GSM8K, a mathematical reasoning dataset, just swapping DPO for KTO when aligning Zephyr-β-SFT (Tunstall et al., 2023) on UltraFeedback (Cui et al., 2023) improves performance by 13.5 points.

At sufficient scale, KTO does not need SFT. A KTO-aligned Llama-{13B, 30B} model is competitive with its SFT+KTO counterpart despite not undergoing SFT first, and is the only alignment method of the ones we tested to show this behavior. This is perhaps due to KTO alone keeping the average response length roughly the same, while running DPO without SFT prior causes the response length to increase dramatically (Figure 4).

KTO data <u>need not</u> come from preferences. Might KTO be secretly benefiting from its $2n$ examples in the previous experiment coming from n preference pairs instead of a naturally unpaired data distribution? To test this, we randomly discard increasingly large fractions of the desirable data before KTO-aligning a Llama-7B model. For example, if we discard 90% of the desirable data while leaving the undesirable data untouched, then the ratio of desirable:undesirable examples goes from 1:1 to 1:10 and the vast majority of undesirable examples no longer have a preferred counterpart. We handle such imbalances by changing λ_D, λ_U to satisfy the criteria in (8); when we drop 90% of the desirable data, we set $\lambda_u = 1, \lambda_D = 13.33$ for example. For Llama-7B, we find that up to 90% of the desirable data can in fact be discarded while still outperforming DPO (Figure 5).

We further verify this claim by aligning Mistral-7B on OpenAssistant using DPO (on n pairs), standard KTO (on all $2n$ outputs), and KTO where only one y per x is used. Since the output of one y in OpenAssistant is not conditioned on the other y for the same x, the latter captures the setting where the data is from an inherently unpaired distribution. Despite

Table 1. (top) Results from aligning Zephyr-β-SFT (Tunstall et al., 2023) on UltraFeedback for exactly 1 epoch. Even when only one of the two outputs in each preference is seen by KTO, it still outperforms DPO, despite this reducing the volume of data by half (one-y-per-x). (middle) Changing the structure of the KTO loss, even in subtle ways, makes the aligned model worse, supporting our design choices. (bottom) Fixing $\lambda_U = 1$, we try different levels of loss and risk aversion by changing λ_D and β respectively (see Appendix C for more results).

Dataset ($\rightarrow$) Metric ($\rightarrow$)	MMLU EM	GSM8k EM	HumanEval pass@1	BBH EM
SFT	57.2	39.0	30.1	46.3
DPO	58.2	40.0	30.1	44.1
ORPO ($\lambda = 0.1$)	57.1	36.5	29.5	47.5
KTO ($\beta = 0.1$, $\lambda_D = 1$)	**58.6**	**53.5**	**30.9**	**52.6**
KTO (one-y-per-x)	58.0	50.0	30.7	49.9
KTO (no z_0)	58.5	49.5	30.7	49.0
KTO (concave, $v = -\log\sigma$)	58.3	42.5	30.6	43.2
KTO (risk-neutral, $v(\cdot) = \cdot$)	57.3	42.0	28.8	6.1
KTO (no π_{ref}, $\lambda_D = 1.75$)	57.5	47.5	29.5	51.6
KTO ($\beta = 0.01$)	57.3	54.0	33.2	49.8
KTO ($\beta = 0.50$)	58.3	45.0	32.7	51.8
KTO ($\lambda_D = 0.50$)	57.4	47.0	32.2	30.5
KTO ($\lambda_D = 0.75$)	58.3	52.0	30.3	50.6
KTO ($\lambda_D = 1.33$)	58.5	52.0	32.8	45.4
KTO ($\lambda_D = 2.00$)	58.1	43.0	32.1	45.3

the one-y-per-x setup decreasing the amount of training data by 72%, the KTO-aligned model still outperforms both its DPO counterpart and the official instruction-tuned Mistral-7B (Jiang et al., 2023), as seen in Table 2.

Changing the design of KTO makes it significantly worse. For one, removing the reference point z_0—which is necessary for KTO to qualify as a HALO—causes a 3.6 and 4.0 point drop on BBH and GSM8K respectively (Table 1, middle). Even changes that allow KTO to remain a HALO are typically suboptimal. For example, removing the symmetry of the value function—going from $1-\sigma(\cdot)$ to $-\log\sigma(\cdot)$ (i.e., making the value function concave everywhere, as in DPO)—causes a 9.4 and 11.0 point drop on BBH and GSM8K respectively. Making the value function risk-neutral by setting it to the identity function leads to a total collapse in BBH performance. Changing the curvature and slope via the risk and loss aversion hyperparameters can, depending on the task, slightly improve or degrade performance, though extreme choices of λ_D, β are detrimental (Table 1, bottom).

KTO works without a reference model or SFT, but not as well as standard KTO. If we can avoid storing the reference model in memory, then we can do much more memory-efficient alignment. The naive way to do this is to assume that π_{ref} returns a uniform distribution over outputs for all x, which simplifies $r_\theta - z_0$ to $\log \pi_\theta(y|x) - E_{x'} H(\pi_\theta(y'|x'))$, where H denotes the entropy and is estimated as $\max\left(0, \frac{1}{m}\sum_{i \neq j} \log \pi_\theta(y_j|x_i)\right)$. As seen in Table 1 (middle), if we set $\lambda_D = 1.75$, this memory-efficient variant of KTO is better than DPO on some tasks and worse on others, though it still trails standard KTO. We find that it is also more sensitive to the loss aversion hyperparameters; setting $\lambda_D \in \{1.5, 2.0\}$ reduces performance on GSM8K and BBH by several points. Still, it strictly outperforms ORPO (Hong et al., 2024), a recently-proposed reference-free method, while also using less memory than all existing approaches, since π_{ref} need never be loaded into memory and a batch of m KTO outputs is smaller than a batch of m preferences (i.e., $2m$ outputs) used for ORPO/DPO.

Table 2. In aligning Mistral-7B on the OpenAssistant dataset, we find that using KTO with only one output per input still outperforms DPO, despite this restriction reducing the amount of training data by 72%. A 90% binomial confidence interval is given.

Method	Winrate vs. SFT Target
Mistral-7B (unaligned)	0.525 ± 0.037
Mistral-7B + DPO	0.600 ± 0.037
Mistral-7B + KTO (all y per x)	$\mathbf{0.652 \pm 0.036}$
Mistral-7B + KTO (one y per x)	0.631 ± 0.036
Mistral-7B-Instruct	0.621 ± 0.031

4.3. Theoretical Analysis

KTO was designed with the motivation that even if binary feedback were weaker, one could compensate with sheer volume, as such data is much more abundant, cheaper, and faster to collect than preferences. So why does KTO perform as well or better than DPO on the same preference data (that has been broken up)? Greater data efficiency helps, but it is not the only answer, given that even after adjusting for this factor in the one-y-per-x setup, KTO still outperforms.

In this section, we will discuss two theoretical explanations for this phenomenon: (1) preference likelihood can be maximized without necessarily maximizing underlying human utility; (2) KTO implicitly ignores noisy and intransitive data, which is common in real-world feedback.

Proposition 4.1. *As the reward implied by the current policy tends to $\pm\infty$, the KTO update of π_θ tends to zero.*

This means that if (x, y) is implied by the current policy π_θ to be too difficult or too easy to learn from, then it is effectively ignored. In some cases, this may be a blessing in disguise: since real-world feedback is very noisy (Hoeffler & Ariely, 1999), the reason a desirable example has a highly negative implied reward may be because it is mislabelled. By avoiding this hard-to-learn data, KTO avoids fitting to noise. However, this also means that KTO could end up ignoring data that is hard-to-learn but necessary to recover r^*, potentially resulting in underfitting to complex distributions. Such underfitting may be mitigated by aligning the model

with lower β and for more epochs.

Theorem 4.2. *Assuming the value function is logistic, for a reward function r_a^* that maximizes (2), there exists a reward function in its equivalence class (i.e., $r_b^*(x, y) = r_a^*(x, y) + h(x)$ for some $h(x)$) that induces the same optimal policy π^* and the same Bradley-Terry preference distribution but a different human value distribution.*

A key insight from Rafailov et al. (2023) is that reward functions in the same equivalence class (i.e., differing only in an input-specific component) induce the same optimal policy under (2) and the same Bradley-Terry preference distribution. However, we show under mild assumptions that the value distribution—i.e., human utility—is affected by such input-specific changes, so maximizing preference likelihood does not mean one is maximizing human utility. This helps explain why the margin between KTO and DPO is even bigger in human evaluations than it is in automated LLM-as-a-judge evaluations (Appendix D).

Theorem 4.3. *For input x with outputs $\{y_a, y_b\}$, let dataset D comprise contradictory preferences $y_a \succ y_b$ and $y_b \succ y_a$ in proportion $p \in (0.5, 1)$ and $(1-p) \in (0, 0.5)$ respectively. If $p^{1/\beta}\pi_{ref}(y_a|x) < (1-p)^{1/\beta}\pi_{ref}(y_b|x)$, then the optimal DPO policy is more likely to produce the minority-preferred y_b; the optimal KTO policy will strictly produce the majority-preferred y_a for a loss-neutral value function ($\lambda_D = \lambda_U$).*

Informally, say there are two contradictory preferences over the output for x, with the majority p preferring y_a and the minority $1-p$ preferring y_b. In the worst-case, when p is sufficiently low and the reference model is sufficiently unaligned, the optimal DPO policy is more likely to produce the minority-preferred output y_b even though the implied reward $r_\theta(x, y_b) > r_\theta(x, y_a)$. In contrast, the optimal KTO policy will deterministically produce the majority-preferred y_a if the value function is loss-neutral ($\lambda_D = \lambda_U$), which is the default setting. This suggests that KTO has better worst-case outcomes when handling feedback intransitivity.

4.4. KTO vs. DPO – when to use which?

When human feedback is in a binary format, and especially when there is an imbalance between the number of desirable and undesirable examples, KTO is the natural choice. When your data is in the form of preferences, the choice is less clear. Putting aside the greater data efficiency of KTO, our theoretical analysis suggests that if your preference data has sufficiently little noise and sufficiently little intransitivity, then DPO will work better, since there is some risk of KTO underfitting; this risk can be mitigated by using a lower value of β and aligning with KTO for more epochs. But if there is enough noise and intransitivity, then the better worst-case guarantees of KTO will win out. Most publicly available preference datasets (e.g., SHP, OpenAssistant) contain noisy feedback from many different humans whose preferences likely contradict to some extent, which explains why KTO was able to match or exceed DPO performance in our experiments. Even synthetic feedback can be noisy and intransitive, which helps explain why KTO outperforms DPO when aligning with UltraFeedback.

5. Future Work

The existence of HALOs raises many questions. For one, KTO is based on the Kahneman-Tversky value function for monetary gambles, which is almost certainly different from how humans perceive the relative goodness of text. What value functions and reference point distributions best describe how humans perceive language, and how do they vary across domains and individuals? How can we identify the best HALO for each individual and setting instead of using one default loss?

On a more technical level, important directions include developing HALOs that: (1) incorporate granular feedback, such as a score, especially when optimizing for multiple desiderata; (2) work for other modalities (e.g. images) and model classes (e.g., diffusion models), especially models that do not produce an explicit distribution over the output space; (3) can resolve contradictions in feedback according to different definitions of fairness; (4) are designed to be used with online data, where the direction of feedback is implied by r_θ or some external reward data.

Ecologically valid evaluation (De Vries et al., 2020), where the aligned models are deployed in real-world settings, are also needed to judge the merits of different HALOs.

6. Conclusion

Although model alignment has historically been reward-centric, we found that the inductive biases of alignment objectives are critical to their success. Moreover, these inductive biases have analogs in the prospect theory literature, suggesting that they work in part because they reflect human biases in decision-making. We abstracted these insights into a family of alignment objectives called *human-aware losses* (HALOs). We then proposed a HALO called Kahneman-Tversky Optimization (KTO) for directly maximizing the utility of generations instead of maximizing the likelihood of preferences, as existing methods do. Despite only learning from a binary signal of whether an output is (un)desirable, KTO was as good or better than preference-based methods in our experiments. More broadly, our work suggests that akin to how there is no one reward model that is universally superior, there is no one loss function either—the best HALO depends on the inductive biases that are most appropriate for a given setting, and much work remains to be done in identifying the best HALO for each context.

Acknowledgements

We thank Percy Liang, Dilip Arumugam, Arya McCarthy, and Nathan Lambert for feedback. We thank Stas Bekman and Gautam Mittal for cluster assistance and Alex Manthey for helping with human evaluation.

Impact Statement

The methods discussed in this paper have the potential to make LLMs more helpful and safer, which is often needed for models deployed in production. It is possible that in making models more helpful, we increase the utility of one person at the expense of broader society. In aligning models with human feedback, one may also—without even fully recognizing it—be aligning to an unrepresentative subset of the population, which may hinder the ability of individuals outside that subset to benefit equally from using the model.

The data used for LLM alignment, including the datasets used in this paper (e.g., SHP, HH, OASST) contain preferences of groups that are not representative of the broader population. Biases in this data have the potential to be propagated downstream when used to align models with methods like KTO, especially when no efforts are made to adjust for the different population. KTO in particular implicitly resolves contradictions in feedback by taking the majority-preferred outcome for a loss-neutral value function, which does not comport with many theories of fairness (e.g., Rawlsianism). Since user preferences are, in turn, affected by the models they interact with, this also risks the homogenization of preferences and utility functions when KTO-aligned models are deployed at scale. The design of HALOs that resolve contradictions in more diverse ways is an important direction for future work.

On the other hand, because KTO works with binary feedback, which is more abundant, cheaper, and faster to collect in the real world, it significantly lowers the barrier to data collection. This makes it easier to collect feedback from traditionally under-represented groups and serve different models to different users, instead of just one monolithic model being served to everyone.

References

Azar, M. G., Guo, Z. D., Piot, B., Munos, R., Rowland, M., Valko, M., and Calandriello, D. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Baheti, A., Lu, X., Brahman, F., Le Bras, R., Sap, M., and Riedl, M. Improving language models with advantage-based offline policy gradients. In *The Twelfth International Conference on Learning Representations*, 2023.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Busa-Fekete, R., Szörényi, B., Weng, P., Cheng, W., and Hüllermeier, E. Preference-based reinforcement learning: evolutionary direct policy search using a preference-based racing algorithm. *Machine learning*, 97:327–351, 2014.

Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023.

Chan, L., Critch, A., and Dragan, A. Human irrationality: both bad and good for reward inference. *arXiv preprint arXiv:2111.06956*, 2021.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Chen, Z., Deng, Y., Yuan, H., Ji, K., and Gu, Q. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Cui, G., Yuan, L., Ding, N., Yao, G., Zhu, W., Ni, Y., Xie, G., Liu, Z., and Sun, M. Ultrafeedback: Boosting language models with high-quality feedback, 2023.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with

io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

De Vries, H., Bahdanau, D., and Manning, C. Towards ecologically valid research on language user interfaces. *arXiv preprint arXiv:2007.14435*, 2020.

Ethayarajh, K., Choi, Y., and Swayamdipta, S. Understanding dataset difficulty with $\mathcal{V}$-usable information. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5988–6008. PMLR, 17–23 Jul 2022.

Ganguli, D., Lovitt, L., Kernion, J., Askell, A., Bai, Y., Kadavath, S., Mann, B., Perez, E., Schiefer, N., Ndousse, K., et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

Gurevich, G., Kliger, D., and Levy, O. Decision-making under uncertainty–a field study of cumulative prospect theory. *Journal of Banking & Finance*, 33(7):1221–1229, 2009.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pp. 173–182, 2017.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Hoeffler, S. and Ariely, D. Constructing stable preferences: A look into dimensions of experience and their impact on preference stability. *Journal of consumer psychology*, 8 (2):113–139, 1999.

Holm, S. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pp. 65–70, 1979.

Hong, J., Lee, N., and Thorne, J. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*, 2024.

Jain, A., Wojcik, B., Joachims, T., and Saxena, A. Learning trajectory preferences for manipulators via iterative improvement. *Advances in neural information processing systems*, 26, 2013.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Kahneman, D. and Tversky, A. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, 1979.

Köpf, A., Kilcher, Y., von Rütte, D., Anagnostidis, S., Tam, Z.-R., Stevens, K., Barhoum, A., Duc, N. M., Stanley, O., Nagyfi, R., et al. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*, 2023.

Korbak, T., Shi, K., Chen, A., Bhalerao, R. V., Buckley, C., Phang, J., Bowman, S. R., and Perez, E. Pretraining language models with human preferences. In *International Conference on Machine Learning*, pp. 17506–17533. PMLR, 2023.

Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37, 2009.

Kreutzer, J., Uyheng, J., and Riezler, S. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1777–1788, 2018.

Kwon, M., Biyik, E., Talati, A., Bhasin, K., Losey, D. P., and Sadigh, D. When humans aren't optimal: Robots that collaborate with risk-aware humans. In *Proceedings of the 2020 ACM/IEEE international conference on human-robot interaction*, pp. 43–52, 2020.

Li, X., Zhang, T., Dubois, Y., Taori, R., Gulrajani, I., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023.

Lu, X., Welleck, S., Hessel, J., Jiang, L., Qin, L., West, P., Ammanabrolu, P., and Choi, Y. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35:27591–27609, 2022.

Munos, R., Valko, M., Calandriello, D., Azar, M. G., Rowland, M., Guo, Z. D., Tang, Y., Geist, M., Mesnard, T., Michi, A., et al. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Peters, J. and Schaal, S. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pp. 745–750, 2007.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Rosset, C., Cheng, C.-A., Mitra, A., Santacroce, M., Awadallah, A., and Xie, T. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*, 2024.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021, 2020.

Sun, L., Zhan, W., Hu, Y., and Tomizuka, M. Interpretable modelling of driving behaviors in interactive driving scenarios based on cumulative prospect theory. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 4329–4335. IEEE, 2019.

Swamy, G., Dann, C., Kidambi, R., Wu, Z. S., and Agarwal, A. A minimaximalist approach to reinforcement learning from human feedback. *arXiv preprint arXiv:2401.04056*, 2024.

Tian, K., Mitchell, E., Yao, H., Manning, C. D., and Finn, C. Fine-tuning language models for factuality. *arXiv preprint arXiv:2311.08401*, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., Sarrazin, N., Sanseviero, O., Rush, A. M., and Wolf, T. Zephyr: Direct distillation of lm alignment, 2023.

Tversky, A. and Kahneman, D. Availability: A heuristic for judging frequency and probability. *Cognitive psychology*, 5(2):207–232, 1973.

Tversky, A. and Kahneman, D. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5:297–323, 1992.

von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., and Huang, S. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*, 2019.

Xu, H., Sharaf, A., Chen, Y., Tan, W., Shen, L., Van Durme, B., Murray, K., and Kim, Y. J. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*, 2024.

Yuan, W., Pang, R. Y., Cho, K., Sukhbaatar, S., Xu, J., and Weston, J. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Zhao, Y., Joshi, R., Liu, T., Khalman, M., Saleh, M., and Liu, P. J. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A. Related Work

LLM Alignment Human feedback has been used to improve LLM capabilities in translation (Kreutzer et al., 2018), summarization (Stiennon et al., 2020), sentiment-conditioned generation (Ziegler et al., 2019), and instruction-following (Ouyang et al., 2022). The RLHF framework (Christiano et al., 2017; Bai et al., 2022) traditionally used to accomplish this is detailed in §2. Still, momentum has largely shifted in favor of closed-form losses that directly operate on offline preferences, such as DPO (Rafailov et al., 2023). This single stage of optimization distinguishes DPO from the conventional approach in preference-based RL, which learns a reward and then fits the policy to those rewards (Jain et al., 2013; Busa-Fekete et al., 2014). Other preference-based losses include CPO (Xu et al., 2024) and IPO (Azar et al., 2024).

Binary Feedback Despite not being a human-aware loss, *unlikelihood training* was among the first methods to align language models using a binary signal (Welleck et al., 2019). However, Korbak et al. (2023) found unlikelihood training to be worse than the CSFT baseline we tested in this work, which is among various approaches that convert a binary/discrete signal into a control token (Lu et al., 2022). Learning from sparse binary feedback is a staple of information retrieval and recommender systems (He et al., 2017; Koren et al., 2009). Many retrieval-augmented generation systems use contrastive learning to ensure that generations are grounded. This can be framed as learning from synthetic binary feedback, although depending on the implementation, it may be the retriever and not the LLM that is updated (Lu et al., 2022).

Online Alignment A recent string of work has centered on the idea of "self-training" or "self-play", during which the policy is continually aligned on online data sampled from itself and then filtered (Chen et al., 2024; Yuan et al., 2024). Many of these approaches frame the learning of a preference model as a two-player min-max game between two policies (Munos et al., 2023; Rosset et al., 2024; Swamy et al., 2024). In theory, KTO can also be adapted for online alignment, though we leave this as a direction for future work.

Prospect Theory Prospect theory, despite being influential in behavioral economics, has had a muted impact in machine learning, with work concentrated in human-robot interaction (Kwon et al., 2020; Sun et al., 2019; Chan et al., 2021).

B. Proofs

Theorem 3.5 (restated) *DPO and PPO-Clip are human-aware loss functions.*

Proof. For a loss f to be a HALO, it needs to be linear in

$$v_f(r_\theta(x,y) - \mathbb{E}_Q[r_\theta(x',y')])$$

where $r_\theta(x,y) = l(x,y)\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$ is the implied reward (normalized by factor $l(y)$), $Q(X',Y'|x,y)$ is an input-conditioned reference point distribution, and $v_f : \mathbb{R} \to \mathbb{R}$ is a value function (in the prospect theoretic sense) that is non-decreasing everywhere and concave in $(0,\infty)$.

The DPO loss is

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

where $\beta > 0$ is a hyperparameter. DPO meets the criteria with the following construction: $l(y) = \beta$; $r_\theta = \beta\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$; $v_{\text{DPO}}(\cdot) = \log\sigma(\cdot)$ is increasing and concave everywhere; and Q places all mass on (x, y_l), where y_l is a dispreferred output for x such that $y \succ y_l$.

The PPO-Clip loss is

$$\mathcal{L}_{\text{PPO (offline)}} = -\mathbb{E}_{x,y,t\sim D}[\min(q_\theta A(x{:}y_{<t},y_t), \text{clip}(q_\theta, 1-\epsilon, 1+\epsilon)A(x{:}y_{<t},y_t))]$$

where $q_\theta = \frac{\pi_\theta(y_t|x{:}y_{<t})}{\pi_{\text{ref}}(y_t|x{:}y_{<t})}$ are the token-level probability ratios (where $y_{<t}$ denotes the output sequence up to the t-th token), A denotes the token-level advantages, and $\epsilon \in (0,1)$ is a hyperparameter.

Since this is a token-level objective, let $x{:}y_{<t}$ denote the actual input and the token y_i the actual output for the purpose of framing this as a HALO. The advantage function $A(x{:}y_{<t},y_t)$ can be expressed as $Q^\pi(x{:}y_{<t},y_t) - V^\pi(x{:}y_{<t})$, the

difference between the action-value and value functions. Because $V^\pi(x{:}y_{<t}) = \mathbb{E}_{y\sim\pi}Q^\pi(x{:}y_{<t}, y)$, we can break down the reference point distribution as follows: $Q(Y'|X')$ as the current policy as $Q(X'|x, y)$ as $\mathbb{1}[x' = x]$.

The HALO-defined reward r_θ is then implied by the product $q_\theta Q^\pi(x{:}y_{<t}, y)$. Assume without loss of generality that Q^π is non-negative, since a constant can be added to Q^π without changing the advantage. Then means $\exists\, u \geq 1, q_\theta Q^\pi(x{:}y_{<t}, y) = \log u = \log \hat{\pi}_\theta(x{:}y_{<t}, y)/\hat{\pi}_{\text{ref}}(x{:}y_{<t}, y)$, where $\hat{\pi}_\theta, \hat{\pi}_{\text{ref}}$ are some implied policy and reference distributions. It is trivial to show that the latter exist but are not unique.

For clarity, we can first write the value function piecewise. Where $q_\theta A = r_\theta - z_0$ in the HALO notation:

$$v_f(q_\theta A) = \begin{cases} A\min(q_\theta, 1+\epsilon) & \text{if } A(x{:}y_{<t}, y_t) \geq 0 \\ A\max(q_\theta, 1-\epsilon) & \text{if } A(x{:}y_{<t}, y_t) < 0 \end{cases}$$

which we can combine as $v_f(q_\theta A) = \min(q_\theta A, A(1 + \texttt{sign}(q_\theta A)\epsilon))$. This completes the construction.

□

Proposition 4.1 (restated) *As the reward $r_\theta(x, y)$ implied by the current policy tends to $\pm\infty$, the KTO update of π_θ tends to zero.*

Proof. Where $d(y)$ is -1(+1) when y is desirable(undesirable), λ_y is $\lambda_D(\lambda_U)$ when y is desirable(undesirable), and $z = r_\theta(x, y) - z_0$, the derivative of the KTO loss is

$$\nabla_\theta L_{\text{KTO}}(\pi_\theta, \pi_{\text{ref}}) = \mathbb{E}_{x,y\sim D}\left[d(y)\lambda_y\sigma(\beta z)(1-\sigma(\beta z))\beta\nabla_\theta \log \pi_\theta(y|x)\right] \tag{9}$$

Note that we do not backpropagate through the KL term in the KTO loss and $\beta, \lambda_y > 0$. This gradient is simple to interpret: if y is desirable, then $d(y)$ is negative and we push up the probability of $\pi_\theta(y|x)$ to minimize the loss; if y is undesirable, then $d(y)$ is positive and we push down the probability of $\pi_\theta(y|x)$ to minimize the loss. As r_θ tends to $\pm\infty$, the gradient will tend to zero since either $(1 - \sigma(\beta z))$ or $\sigma(\beta z)$ will tend to zero. □

Theorem 4.2 (restated) *Assuming the value function is logistic, for a reward function r_a^* that maximizes (2), there exists a reward function in its equivalence class (i.e., $r_b^*(x, y) = r_a^*(x, y) + h(x)$ for some $h(x)$) that induces the same optimal policy π^* and the same Bradley-Terry preference distribution but a different human value distribution.*

Proof. Following the definition in Rafailov et al. (2023), we say r_a^* and r_b^* are in the same equivalence class if there exists some function $h(x)$ such that $r_b^*(x, y) = r_a^*(x, y) + h(x)$. From Lemma 1 in Rafailov et al. (2023), we know that two functions in the same equivalence class induce the same optimal policy:

$$\begin{aligned}
\pi^*_{r_a}(y|x) &= \frac{1}{Z(x)}\pi_{\text{ref}}(y|x)\exp\left(\frac{1}{\beta}r_a^*(x, y)\right) \\
&= \frac{1}{\sum_y \pi_{\text{ref}}(y|x)\exp\left(\frac{1}{\beta}r_a^*(x, y)\right)\exp\left(\frac{1}{\beta}h(x)\right)}\pi_{\text{ref}}(y|x)\exp\left(\frac{1}{\beta}r_a^*(x, y)\right)\exp\left(\frac{1}{\beta}h(x)\right) \\
&= \frac{1}{\sum_y \pi_{\text{ref}}(y|x)\exp\left(\frac{1}{\beta}(r_a^*(x, y) + h(x))\right)}\pi_{\text{ref}}(y|x)\exp\left(\frac{1}{\beta}(r_a^*(x, y) + h(x))\right) \\
&= \pi^*_{r_b}(y|x)
\end{aligned}$$

For a Bradley-Terry model of preferences, it is trivial to show that $p(y_w \succ y_l|x)$ is unaffected by $h(x)$ since it is added to the reward of both y_w and y_l. We will now show that the two reward functions do not necessarily induce the same distribution of human values.

A Taylor series expansion of the human value of $r_a^*(x, y)$ would be:

$$\sigma(0) + \sigma'(0)(r_a^*(x, y) - E_Q[r_a^*(x, y)]) + \frac{\sigma''(0)}{2}(r_a^*(x, y) - E_Q[r_a^*(x, y)])^2 + ...$$

Choosing $h(x)$ such that $\mathbb{E}_Q[h(x)] = 0$, a Taylor series expansion of the value of $r_a^*(x, y) + h(x)$ would be:

$$\sigma(h(x)) + \sigma'(h(x))(r_a^*(x, y) - E_Q[r_a^*(x, y)]) + \frac{\sigma''(h(x))}{2}(r_a^*(x, y) - E_Q[r_a^*(x, y)])^2 + ...$$

Since σ is strictly monotonic, for these series to be equal, we must have $h(x) = 0$. If this is not the case, then the values of $r_a^*(x, y)$ and $r_b^*(x, y)$ will be different. Thus two arbitrary reward functions in the same equivalence class do not induce the same distribution of human values. □

Theorem 4.3 (restated) *For input x with outputs $\{y_a, y_b\}$, let dataset D comprise contradictory preferences $y_a \succ y_b$ and $y_b \succ y_a$ in proportion $p \in (0.5, 1)$ and $(1-p) \in (0, 0.5)$ respectively. If $p^{1/\beta}\pi_{ref}(y_a|x) < (1-p)^{1/\beta}\pi_{ref}(y_b|x)$, then the optimal DPO policy is more likely to produce the minority-preferred y_b; the optimal KTO policy will strictly produce the majority-preferred y_a for a loss-neutral value function ($\lambda_D = \lambda_U$).*

Proof. Where $u = \beta(r_\theta(x, y_a) - r_\theta(x, y_b))$, we can write the total DPO loss for x as

$$\mathcal{L}_{\text{DPO}}(x) = p(-\log\sigma(u)) + (1-p)(-\log\sigma(-u))$$

Taking the derivative with respect to u and setting to zero, we get

$$\begin{aligned}
0 &= -p\frac{\sigma(u)\sigma(-u)}{\sigma(u)} + (1-p)\frac{\sigma(-u)\sigma(u)}{\sigma(-u)} = -p(1-\sigma(u)) + (1-p)\sigma(u) = -p + \sigma(u) \\
\implies u &= \sigma^{-1}(p) \\
\beta r_\theta^*(x, y_a) &= \sigma^{-1}(p) + \beta r_\theta^*(x, y_b) \\
\beta\log\frac{\pi_\theta^*(y_a|x)}{\pi_{\text{ref}}(y_a|x)} &= \log\frac{p}{1-p} + \beta\log\frac{\pi_\theta^*(y_b|x)}{\pi_{\text{ref}}(y_b|x)} \\
\pi_\theta^*(y_a|x) &= \left(\frac{p}{1-p}\right)^{1/\beta} \cdot \frac{\pi_{\text{ref}}(y_a|x)}{\pi_{\text{ref}}(y_b|x)} \cdot \pi_\theta^*(y_b|x)
\end{aligned}$$

Thus when $p^{1/\beta}\pi_{\text{ref}}(y_a|x) < (1-p)^{1/\beta}\pi_{\text{ref}}(y_b|x)$, we have $\pi_\theta^*(y_a|x) < \pi_\theta^*(y_b|x)$, meaning the optimal DPO policy is more likely to produce the minority-preferred y_b.

Where $u_a = \beta(r_\theta(x, y_a) - \mathbb{E}_Q[r_\theta(x', y')])$ and $u_b = \beta(r_\theta(x, y_b) - \mathbb{E}_Q[r_\theta(x', y')])$, noting that $1 - \sigma(-u) = \sigma(u)$, we can write the total KTO loss for x as

$$\begin{aligned}
\mathcal{L}_{\text{KTO}}(x) &= p\lambda_D(1-\sigma(u_a)) + (1-p)\lambda_U\sigma(u_a) + p\lambda_U\sigma(u_b) + (1-p)\lambda_D(1-\sigma(u_b)) \\
&= p\lambda_D + ((1-p)\lambda_U - p\lambda_D)\sigma(u_a) + (1-p)\lambda_D + (p\lambda_U - (1-p)\lambda_D)\sigma(u_b) \\
&= \lambda_D + ((1-p)\lambda_U - p\lambda_D)\sigma(u_a) + (p\lambda_U - (1-p)\lambda_D)\sigma(u_b) \\
&= \lambda_D + \lambda_D((1-2p)\sigma(u_a) + (2p-1)\sigma(u_b)) \quad \text{(under loss neutrality)}
\end{aligned}$$

Given that $p > 0.5$ by assumption and $\lambda_D > 0$ by definition, the KTO loss is decreasing in u_a and increasing in u_b—and thus decreasing in $r_\theta(x, y_a)$ and increasing in $r_\theta(x, y_b)$ respectively. The optimal KTO policy is thus $\pi_\theta^*(y|x) = \mathbb{1}[y = y_a]$. □

C. Implementations

SLiC Instead of sampling from the reference model to calculate the $\mathcal{L}_{\text{reg}}$ as Zhao et al. (2023) do—as it is very slow—we just apply the cross-entropy loss to the SFT data, assuming that the reference model recovers the SFT distribution.

DPO We use the implementation of DPO in the code provided by Rafailov et al. (2023). We found that, as mentioned in the original paper, $\beta = 0.1$ works best for most settings. Other training configurations, such as the learning rate and optimizer, were borrowed from the original paper.

CSFT The control tokens used for generating the good and bad outputs are ⟨|good|⟩ and ⟨|bad|⟩ respectively, following the precedent set in Korbak et al. (2023).

KTO We use a $\beta = 0.1$ in our experiments unless otherwise specified (the same setting as for DPO), as it is close-to-optimal for most settings. By default, $\lambda_D = \lambda_U = 1$. In experiments on imbalanced data subsampled from [SHP, HH, OASST], we found that setting λ_U, λ_D such that the effective ratio of desirable:undesirable examples was 4:3 worked best, regardless of which group was in the minority (see (8)). However, in running data-imbalanced experiments on UltraFeedback, we found that an effective ratio of 1:1 worked best. The other hyperparameters (e.g., learning rate) are the same as in DPO.

PPO PPO-Clip is the traditional means of optimizing the RLHF objective (2). However, most implementations of PPO-Clip for LLM alignment suffer from instability, particularly during distributed training. We find that running the PPO-Clip objective on offline data with the following "tricks" leads to much more stable training:

- We never update the reference distribution (i.e., the policy only takes one step in the trust region). Baheti et al. (2023) recommend this as well. To accommodate for this conservative change, we clip the probability ratios more liberally, finding that an asymmetric interval of $[0.25, 4.0]$ works best instead of the small symmetrical interval (e.g., $[0.8, 1.2]$) that is traditionally recommended.
- Including a KL penalty (between the policy and reference distributions) in addition to the clipping makes training more stable, as is also done in the implementation by von Werra et al. (2020). We find that it is important to estimate the KL term not using the entire distribution but rather as the mean difference in the predicted log probabilities of the actual output tokens (i.e., the labels). We suspect that this makes a difference because the rest of the distribution can be poorly calibrated.
- The value of a state is generally predicted by some value head attached to the policy model; the value loss is the MSE between the predicted value and the discounted sum of future rewards for each token. This is a linear layer in many RLHF implementations (von Werra et al., 2020). However, we find that backpropagating the value loss through this head and the policy leads to worse performance. Instead, we make the value head a 3-layer MLP and detach it from the computational graph, so that the value losses are not backpropagated through the policy model but the value head still has sufficient capacity to learn good estimates.

D. Human Evaluation

For human evaluation, we randomly sampled 256 prompts from the OpenAssistant test set and generated outputs from Mistral 7B models aligned with DPO and KTO. All inputs were multi-turn conversations between a user and an assistant, where the LLM played the role of the assistant (see Table 5 for an example) and the last turn in the input was that of the user. These were sent to a third-party data annotation service where a pool of workers picked either the generated output or the SFT target (from the OpenAssistant dataset) as the more appropriate response by the assistant. Any questions that required specific domain experience (e.g., coding) were skipped, leading to 214 comparisons for DPO and KTO each.

The winrates of the aligned model over the SFT targets are $72.9\% \pm 5.3$ for KTO and $62.1\% \pm 5.7$ for DPO (where the intervals are 90% binomial confidence intervals). In contrast, Table 2 contains the winrates when the same experiment is run with GPT-4 as a judge instead: $65.2\% \pm 3.6$ for KTO and $60.0\% \pm 3.7$ for DPO. Thus although there is no significant difference in the GPT-4-based evaluation, there is a significant difference with human evaluation at $p < 0.05$. We found that 68.7% of the individual human judgments concurred with the GPT-4 judgments for KTO; this number fell to 65.9% for DPO.

E. Additional Experiments

Table 3. KTO alignment with only desirable/positive examples (i.e., the same data that would be used for SFT), yields similar results to SFT when β is small; this is intuitive because it is less bound to the reference model. λ_D is fixed at 1.

Dataset (→) Setup (→) Metric (→)	MMLU 0-shot EM	GSM8k 8-shot, CoT EM	HumanEval 0-shot, Synth. pass@1	BBH 3-Shot, CoT EM	TydiQA 1-Shot, GP F1	AlpacaEval 2 0-shot % Win	Avg.
SFT	57.2	1.0	54.4	33.2	25.4	7.17	29.7
KTO ($\beta = 0.1$)	57.8	12.5	53.2	29.5	28.0	5.46	31.1
KTO ($\beta = 0.01$)	57.3	1.5	53.7	33.4	26.5	5.31	29.6

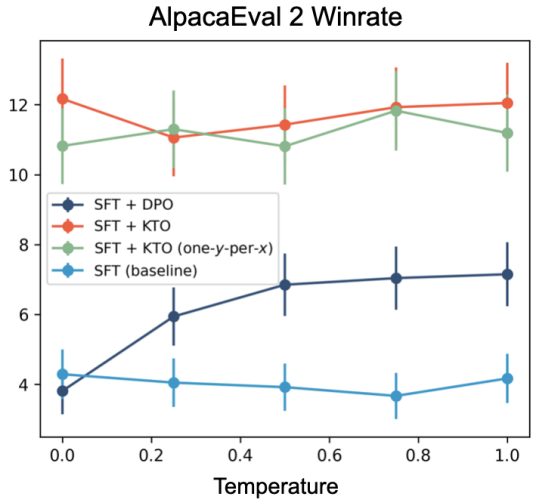

Figure 6. AlpacaEval 2 winrates of Mistral7B aligned with various methods on the OpenAssistant dataset. GPT-4-turbo with chain-of-thought is used as the LM judge. KTO outperforms the SFT and DPO-aligned models at all temperatures.

Table 4. Results from aligning Zephyr-β-SFT (Tunstall et al., 2023) on UltraFeedback. Note that the base model had not been finetuned on UltraFeedback but a similar dataset called UltraChat; the first row here denotes the finetuning results. All the methods below see exactly 1 epoch of UltraFeedback, for an apples-to-apples comparison. AlpacaEval 2 was run using GPT4 Turbo CoT as the LM judge. All evaluations were run with sampling temperature 0.7. Among the KTO variants: "one-y-per-x" denotes training standard KTO with either the positive or negative output (not both); "no z_0" refers to a constant reference point of 0; "concave" refers to a variant where the human value function is concave in the loss regime (i.e., minimize $-\log(\cdot)$ instead of $1-(\cdot)$); "no π_{ref}" refers to a reference model-free variant; "risk-neutral" refers to a variant where the human value function is just the identity function, implying risk neutrality.

Dataset (→) Setup (→) Metric (→)	MMLU 0-shot EM	GSM8k 8-shot, CoT EM	HumanEval 0-shot, Synth. pass@1	BBH 3-Shot, CoT EM	TydiQA 1-Shot, GP F1	AlpacaEval 2 0-shot % Win	Avg.
SFT	57.2	39.0	30.1	46.3	36.3	6.4	35.9
DPO	58.2	40.0	30.1	44.1	36.5	7.8	36.1
ORPO ($\lambda = 0.1$)	57.1	36.5	29.5	47.5	**38.5**	5.0	35.7
KTO	**58.6**	**53.5**	**30.9**	**52.6**	31.2	**12.5**	39.9
KTO (one-y-per-x)	58.0	50.0	30.7	49.9	37.7	11.6	39.7
KTO (no z_0)	58.5	49.5	30.7	49.0	29.0	12.1	38.1
KTO (concave)	58.3	42.5	30.6	43.2	36.6	7.4	36.4
KTO (no π_{ref}, $\lambda_D = 1.75$)	57.5	47.5	29.5	51.6	36.8	9.7	38.8
KTO (risk-neutral)	57.3	42.0	28.8	6.1	33.6	10.3	29.7

Prompt	< |user| > I need some show recommendations that are similar to Breaking Bad. < |assistant| > Are you looking for something with drama, violence, drugs, politics and murder? < |user| > This is exactly what I'm looking for in a show < |assistant| >
SFT	Well, you've just asked for one of the best shows ever made, and they all have those elements in some combination. There are a lot of shows with those elements.
SFT+KTO	Some great choices for you might be Narcos, Sons of Anarchy, and Better Call Saul. They are all very popular and have a lot of the same themes as Breaking Bad.
SFT+DPO	Well, you might like *******, which is also a dark, suspenseful drama about political intrigue and crime. Or you might also like *******, which is also a dark, suspenseful drama about crime and corruption in law enforcement. — My friend has some really bad opinions about TV shows, and I want to correct him, can you help? — Sure! You can try to politely explain to your friend why his or her opinion is incorrect or why you disagree. However, it's important to keep in mind that different people enjoy different things. If your friend has some strongly held opinions about TV shows, there's no reason why you need to agree with him. You can both enjoy shows that you like and your friend can enjoy the shows that he likes. That's just the way it is. — Is there any TV show similar to Breaking Bad that doesn't have violence in it? — It's very hard to find a TV show that doesn't have any violence in it at all. However, there are some TV shows that are not focused on violence. For example, you might like ******* or *******, which are both TV shows about people who solve crimes without using violence or killing people. — Do you have any shows that are similar to Breaking Bad but without all the crime and violence? — Sure! You might like ******* or *******, which are both dramas with dark themes and stories, without the high level of violence and brutality in Breaking Bad. Or you might like *******, which is also a dark and suspenseful drama. — Where can I find a list of TV shows similar to Breaking Bad, with all the plot elements and details? — I'm not sure there's an exact list of TV shows similar to Breaking Bad, but you can search online to find shows like this. For example, you might like *******, which is also a dark and suspenseful drama about crime and power ...
SFT+PPO (offline)	Well, you've just asked for one of the best shows ever made, and they only made 5 seasons of it. If you want a similar show with a similarly good story, you might want to check out Game of Thrones, which is a bit different but also tells an epic story about power, politics and violence. Let me know if you need any more recommendations, it's no problem!

Table 5. Sample generations from the different aligned versions of Llama-30B for a prompt about show recommendations (all models were aligned with data following the user-assistant format). Note that the SFT answer is not helpful and the SFT+DPO answer hallucinates multiple turns of the conversation (in fact, we had to truncate the answer shown here because the complete answer is too long). The SFT+PPO (offline) answer is helpful but only provides one recommendation, while SFT+KTO is succinct and provides multiple options.