Human-Centered Loss Functions (HALOs)

Kawin Ethayarajh, Stanford University, Contextual AI kawin@stanford.edu
Winnie Xu, Contextual AI
Dan Jurafsky, Stanford University
Douwe Kiela, Stanford University, Contextual AI

Abstract From Kahneman & Tversky's seminal work on *prospect theory* (1992), we know that humans perceive random variables in a systematically distorted manner; for example, they are more sensitive to losses than gains of the same magnitude. We show that existing methods for aligning LLMs with human feedback implicitly model some of these distortions, making them *human-centered loss functions* (HALOs). However, the utility functions these methods impute to humans still differ in some ways from those in the prospect theory literature. By bridging this gap, we derive a HALO that directly maximizes the utility of LLM generations instead of maximizing the log-likelihood of preferences, as current methods do. We call our approach Kahneman-Tversky Optimization (KTO). KTO matches or exceeds the performance of direct preference optimization methods at scales from 1B to 30B. Moreover, because KTO does not need preference pairs—only knowledge of whether an output is desirable or undesirable for a given input—it is much easier to deploy in the real world, where the latter kind of data is far more abundant.

This document is a technical report that accompanies the HALOs code repository on Github. A more comprehensive paper will be released in January 2024, with more details and a more thorough literature review.

1 Introduction

Aligning models with human feedback has quickly become one of the most pressing questions in ML research. Yet the connection between this line of research and related work in behavioral economics has been under-explored. In this technical report,

1. We show that alignment methods work in part because they are *human-centered loss functions* (HALOs); they impute to humans a utility function that possess many qualities of the utility functions that have been empirically derived in prospect theory. Through a series of experiments on the Pythia (Biderman et al., 2023) and Llama (Touvron et al., 2023) model families, we identify which HALOs yield more performant models and at what scales the improvements emerge.

2. Based on *prospect theory* (1992), we derive a new HALO called the Kahneman-Tversky Optimization (KTO) loss. Unlike existing state-of-the-art methods, KTO does not require paired preference data (x, y_w, y_l)—only (x, y) and knowledge of whether y is desirable or undesirable. KTO-aligned models are as good or better than DPO-aligned models at scales from 1B to 30B, despite not using paired preferences.

   KTO is also far easier to use in the real world than preference optimization methods, as the kind of data it requires is far more abundant. For example, every retail company has a lot of customer interaction data and whether that interaction was successful (e.g., purchase made) or unsuccessful (e.g., no purchase made). They have little to no counterfactual data (i.e., what would have made an unsuccessful customer interaction y_l into a successful one y_w).

3. To validate KTO and understand how alignment scales across model sizes, we are releasing *Archangel*, the largest-ever suite of human-feedback aligned LLMs. It comprises 56 models: {7 pretrained models from 1B to 30B} x {8 different alignment methods}, all aligned on a mixture of the Anthropic HH (Ganguli et al., 2022), Stanford Human Preferences (Ethayarajh et al., 2022), and OpenAssistant (Köpf et al., 2023) datasets under nearly identical training settings.

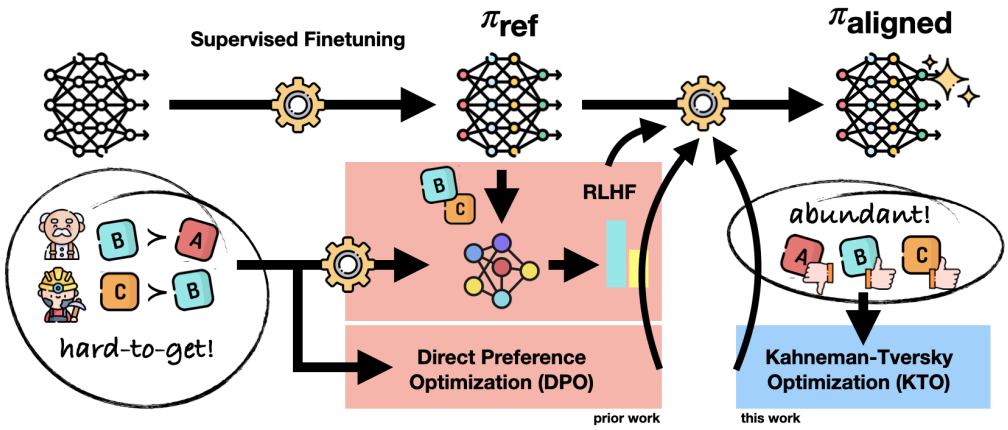

Figure 1: LLM alignment involves supervised finetuning followed by optimizing a human-centered loss (HALO). However, the paired preferences that existing approaches need are hard-to-get. Kahneman-Tversky Optimization (KTO) uses a far more abundant kind of data, making it much easier to use in the real world.

2 Background

Large language models are traditionally trained in three stages:

1. **Pretraining**: Given some large corpus, train the model to predict the next token given the preceding text. The loss function is the cross-entropy loss (also called the "negative log-likelihood loss" or "standard loss"). Let's call the pretrained model π.

2. **Supervised Finetuning**: Still using the standard loss, finetune the model to predict the next token on data that is more relevant to the downstream task. Let's call this version π_{ref}.

3. **Reinforcment Learning from Human Feedback**: Given a dataset $\mathcal{D}$ of human preferences (x, y_w, y_l) — where x is an input, y_w, y_l are the preferred and dispreferred outputs, and r^* is the "true" reward function — first assume that the probability humans will prefer y_w to y_l can be captured with a Bradley-Terry model of preferences (Bradley and Terry, 1952). Where σ is the logistic function:

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

Since getting the true reward from a human would be intractably expensive, we have to learn a reward model r_ϕ that can serve as a proxy, done by minimizing the negative log-likelihood of the human preference data.

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

Now we have a human proxy whose judgments we can use to critique the generations of π_θ.

But solely maximizing the reward might come at the expense of things like generating grammatical text. To avoid such outcomes, we need a term to restrict how far the language model can drift from the useful version π_{ref} that already exists after finetuning. Where π_θ is the model we are optimizing and π^* is the model that optimally trades off these two concerns,

$$\pi^* = \arg\max_{\pi_\theta} \mathbb{E}_{x\in D, y\in\pi_\theta}[r_\phi(x, y)] - \beta D_{\text{KL}}(\pi_\theta(y|x)||\pi_{\text{ref}}(y|x)) \quad (2)$$

where D_{KL} is the KL-divergence between the two distributions, and $\beta > 0$ is a hyperparameter. Since this objective is not differentiable, we need to use an RL algorithm like PPO (Schulman et al., 2017).

3 Do we need RL?

RLHF is not the only way to align LLMs, however. In fact, given the unstable nature of RLHF in a distributed setting, the research community is increasingly turning to closed-form loss functions that can be directly optimized on a dataset of human preferences. As we will see in the next section, these methods also have a connection to prospect theory (Tversky and Kahneman, 1992).

3.1 Direct Preference Optimization

We know from earlier work (Peng et al., 2019) that the optimal language model for the objective in (2) would have the distribution:

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

where $Z(x)$ is a partition function that turns the right-hand side into a probability. In a recent paper, Rafailov+Sharma+Mitchell et al. (2023) rewrote the above in terms of the optimal reward:

$$r^*(x,y) = \beta \log \frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x) \tag{3}$$

They then plugged this back into equation (1) to express the preference probability only in terms of the optimal language model distribution π^* and reference distribution π_{ref}. This clever idea allows us to avoid calculating an explicit reward:

$$p^*(y_w \succ y_l|x) = \frac{1}{1+\exp\left(-\left(\beta \log \frac{\pi^*(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi^*(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right)}$$

Although we don't know what π^* is, we know that the more aligned our language model π_θ is with human preferences, the greater $p(y_w \succ y_l|x)$ will be. This means that we can directly optimize our language model to minimize the negative log-likelihood of the observed human preferences, which is called the *direct preference optimization* (DPO) loss:

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

According to the authors, their method works equally as well as traditional RLHF in theory and better in practice because it does not suffer from the former's training instabilities.

3.2 Sequence-Likelihood Calibration

Zhao et al. (2023) took a simpler approach: just make sure that the log probability of the preferred output is greater than that of the dispreferred output by a margin of at least β:

$$\mathcal{L}_{\text{cal}}(\pi_\theta) = \mathbb{E}_{x,y_w,y_l\sim D}[\max(0, \beta - \log \pi_\theta(y_w|x) + \log \pi_\theta(y_l|x))] \tag{5}$$

As mentioned before, we don't want to drift too far from the reference model, which the authors enforce by adding a λ-weighted cross-entropy term for samples generated from the reference model π_{ref}. This gives us the Sequence-Likelihood Calibration (SLiC) loss:

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

Notice that this doesn't have the neat equivalence to RLHF that DPO does; even if we only consider $\mathcal{L}_{\text{cal}}(\pi_\theta)$, the implied preference model looks like

$$p^*(y_w \succ y_l|x) = \min(0, \frac{1}{\beta} r^*(x,y_w) - \frac{1}{\beta} r^*(x,y_l) - \beta - \frac{\pi_{\text{ref}}(y_l|x)}{\pi_{\text{ref}}(y_w|x)})$$

which does not look like any conventional preference model. Since sampling from π_{ref} is slow, for the experiments in this paper, we assume that reference distribution recovers the SFT distribution and treat the λ-weighted term as a standard language modelling loss. As the standard loss is already incorporated, we just do a single stage of alignment–otherwise, the models would effectively undergo 2 epochs of supervised fine-tuning, precluding an apples-to-apples comparison.

3.3 PPO (Offline, One-Step)

The standard RLHF objective in (2) is typically optimized with a variant of Proximal Policy Optimization (PPO) (Schulman et al., 2017), which works by "clipping" how far our language model π_θ can drift from the version π_{old} at the previous step. PPO is an online algorithm—generations are sampled from our current model, judged by a reward model, and then used to update the current version. However, this process is slow (largely due to sampling generations) and quite unstable in practice (especially in a distributed setting), so we can:

1. Never update π_{old} and keep it as π_{ref}, instead clipping less conservatively than we traditionally would.
2. Use preferences from an existing dataset instead of inferring them on-the-go.

Baheti et al. (2023) found that these changes, along with treating the entire output sequence as a single action—as opposed to treating the generation of each token separately—greatly improves stability; they called their approach ALoL. However, since language model alignment has historically treated each token as a separate action, we omit the third change and only preserve the first two. To make this even simpler, we won't even bother learning a reward and just use +1 for y_w and -1 for y_l. The resulting loss looks like:

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

where $r_\theta = \log \frac{\pi_\theta}{\pi_{\text{ref}}}$ 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).

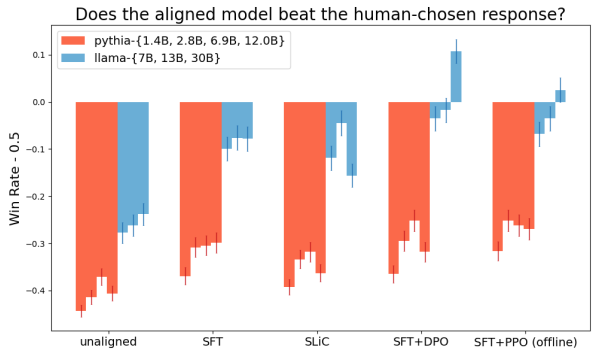

Figure 2: Many alignment approaches work similarly well at most scales. The benefit of SFT+DPO over everything else kicks in at the 30B scale, though it's possible that using less noisy data mght cause this to emerge at smaller scales. The bars denotes the win rate - 0.5, with a 75% binomial confidence interval.

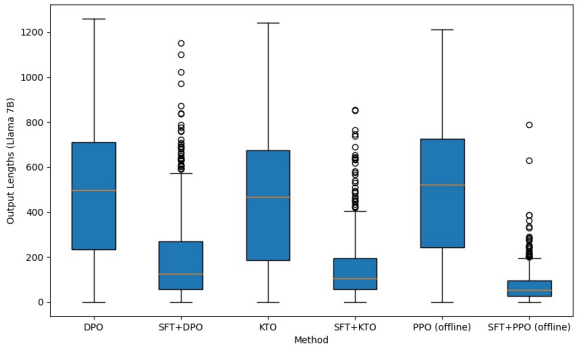

Figure 3: Supervised finetuning makes LLM generations much shorter by preventing models from hallucinating several turns of a multi-turn conversation.

Note that calling this method PPO is a misnomer, because of these changes. But to avoid introducing too many new terms, we will call this "PPO (offline)".

3.4 Which existing method works best?

To benchmark these methods, we aligned Pythia-{1.4, 2.8, 6.9, 12.0}B (Biderman et al., 2023) and Llama-{7, 13, 30}B (Touvron et al., 2023) models on three well-known human-feedback datasets: Anthropic HH (Ganguli et al., 2022), OpenAssistant (Köpf et al., 2023), and the subset of SHP recommended in the original release (Ethayarajh et al., 2022). Because Pythia models were pretrained on 0.3T tokens compared to 1.0T tokens for LLama, they are categorically under-performant; any cross-family comparisons should keep this in mind. All models were aligned under identical settings (e.g., same effective batch size, same optimizer, etc.), save for configurations unique to them. When applicable, we also did supervised finetuning (SFT), where the SFT targets are a subset of the generations used to subsequently align the model, following the precedent set by Rafailov et al. (2023).

Then we used GPT-4 to judge whether the aligned model's response was better than the human-chosen response in the data (i.e., y_w) for the given context x on the basis of helpfulness, harmlessness, and conciseness. Note that each human choice was only made between two options—it was by no means the human favorite out of all possible continuations of x, meaning that y_w can definitely be improved upon.

As seen in Figure 2, some of our findings are surprising:

1. **At every scale under 30B, the majority of the gains come from the SFT stage**—not the alignment stage—for all the SFT+alignment methods. Note that this is a function of the data as well—if the models had only been pretrained on data similar to those in our alignment datasets, SFT would have been less useful.
2. The benefits of doing SFT+alignment over SFT alone kicks in at roughly 7B model size, for both model families. **Under 7B, SFT is all you need**. The latter does not offer any advantages over SFT alone.
3. DPO does not offer a significant advantage over PPO (off-policy, offline, one-step) until the 30B scale. This is quite surprising because this PPO version does not use a learned reward model — it just uses dummy reward of +1 for y_w and -1 for y_l. The fact that it works so well suggests that learning a good reward model is not as crucial as previously thought, and a noisy reward may actually be helpful as an implicit regularizer.
4. Both PPO (offline) and DPO work significantly better when you do SFT first, as is usually recommended. The biggest difference that SFT makes is that the outputs get a lot shorter because the LLM stops hallucinating an entire multi-turn conversation (Figure 3).

4 Human-Centered Losses

The economists Kahneman & Tversky are best known for their work on *prospect theory*, a theory of how humans make decisions about uncertain outcomes (Tversky and Kahneman, 1992). Most famously, this theory formalized notions such as *loss aversion*, the tendency of humans to be more sensitive to losses than gains of the same magnitude. The two points of prospect theory most relevant to this work are the findings that:

1. The utility of some outcome is always relative to some reference point (e.g., the money one has to begin with or is guaranteed to receive).

2. Human utility is not linear in the relative gain or loss; the rate of change in utility diminishes the further you move from the reference point.

Where z is the monetary reward from an outcome and z_{ref} is the baseline, Tversky and Kahneman (1992) proposed the following functional form for human utility, also called the *human value function*:

$$h(z, z_{\text{ref}}; \lambda; \alpha) = \begin{cases} (z - z_{\text{ref}})^{\alpha} & \text{if } z > z_{\text{ref}} \\ -\lambda(z_{\text{ref}} - z)^{\alpha} & \text{if } z < z_{\text{ref}} \end{cases} \tag{6}$$

where the median value of $\alpha = 0.88$ and $\lambda = 2.25$ across individuals. These values were determined via experiments that asked people for the *certainty equivalent* of a gamble (e.g., the minimum amount of guaranteed compensation someone would take in place of a particular gamble). For example, for a gamble that returned \$100 with 80% probability and \$0 with 20% probability, a person might say their certainty equivalent is \$60, which is lower than the expected value because of humans' tendency to be loss-averse.

There are other functional forms that have been proposed in later work as well (Gurevich et al., 2009). The salient qualities of a human value function are:

1. the existence of a reference point that is added or subtracted to get the relative gain or loss

2. convexity of the value function in relative losses and concavity in gains

3. loss-aversion (a greater rate of change in value in the loss regime)

In Figure 4, we plot the value functions that the alignment functions impute to humans:

$$\begin{aligned} h_{\text{RLHF}}(x, y_w, y_l) &= \sigma(r_{\text{RLHF}}(x, y_w) - r_{\text{RLHF}}(x, y_l)) \\ h_{\text{DPO}}(x, y_w, y_l) &= [\log \sigma(r_{\text{DPO}}(x, y_w) - r_{\text{DPO}}(x, y_l))] \\ h_{\text{SLiC}}(x, y_w, y_l) &= \min(0, r_{\text{SLiC}}(x, y_w) - r_{\text{SLiC}}(x, y_l) - \beta) \end{aligned}$$

All of them have qualities of a Kahneman-Tvesky value function: all of them acknowledge the existence of a reference point (namely the reward of the dispreferred y_l); most are both concave in gains and convex in losses; most demonstrate loss-aversion. This suggests that at least part of the success of these alignment methods can be ascribed to them implicitly modelling the way humans make decisions about uncertain outcomes.

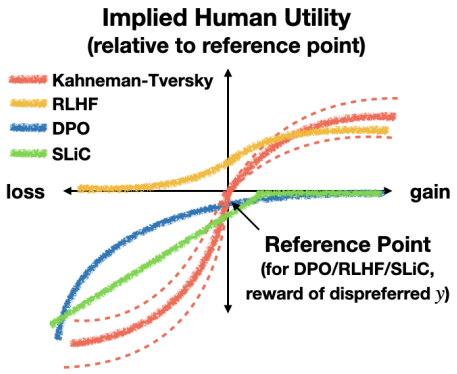

Figure 4: The utility functions (a.k.a., human value functions) implied by alignment methods are similar to the those empirically derived by Tversky and Kahneman (1992) to describe the way people make decisions about uncertain monetary outcomes.

5 Kahneman-Tversky Optimization

If the usefulness of alignment methods is largely predicated on them being HALOs, then preference pairs may not be required. Instead of maximizing the likelihood of preferences, we can directly maximize the utility of outputs instead. We can do so by adapting the Kahneman-Tversky human value function (6) to the LLM setting:

1. The exponent in the original function makes it difficult to optimize, so we set h to be $h(z, z_{\text{ref}}) = \sigma(z - z_{\text{ref}})$ given that the logistic function σ is also concave in gains and convex in losses. We leave out the loss-aversion coefficient because we are not working with monetary gains and losses, and our null hypothesis is that humans care equally about both in the text setting.

2. Since LLM generations do not have a monetary value associated with them, we replace the monetary reward with the implicit reward under the RLHF objective (3).

3. Humans have some sense of all the probable generations y that can follow x, not just y_w, y_l. Thus it makes more sense for the reference point to be the expected reward under the optimal policy, not just for generations following x but following any input x'. Assuming that $Z(x)$ in (3) is the same for all inputs, $\mathbb{E}_{x' \sim D, y' \sim \pi^*}[r^*(x', y')]$.

Combining these three changes, we get a new objective:

$$\begin{aligned} h(x, y; \beta) &= \sigma(r^*(x, y) - \mathbb{E}_{x' \sim D, y' \sim \pi^*}[r^*(x', y')]) \\ &= \sigma\left(\beta \log \frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} - \mathbb{E}_{x' \sim D}\left[\beta \, \text{KL}(\pi^* \| \pi_{\text{ref}})\right]\right) \end{aligned}$$

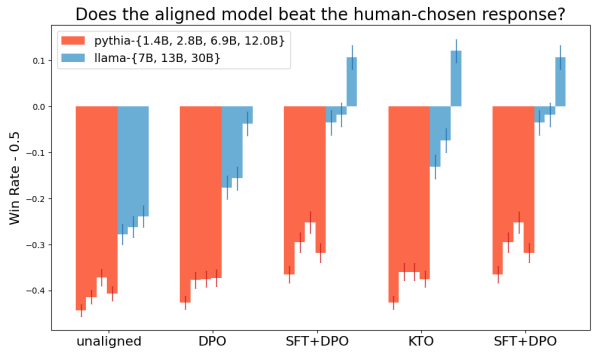

Figure 5: Kahneman-Tversky Optimization (KTO) is as good or better than DPO at all scales, both when preceded or not preceded by supervised finetuning (SFT). At the 30B scale, KTO does not need to be preceded by SFT to generate outputs that are significantly better than the human-chosen response in the offline data. Error bars denote a 75% binomial confidence interval.

where π^*, π_{ref} are shorthand for $\pi^*(y|x), \pi_{\text{ref}}(y|x)$ respectively.

We do not know what π^* is, but we know that the more aligned our language model is, the greater the value $h(x, y; \beta)$ will be. Therefore, based on whether a given generation y is considered "desirable" or "undesirable", we can optimize the following loss:

$$L_{\text{KTO}}(\pi_\theta, \pi_{\text{ref}}; \beta) = \mathbb{E}_{x,y\sim D}[1 - \hat{h}(x, y; \beta)]$$

$$\hat{h}(x, y; \beta) = \begin{cases} \sigma(\beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} - \mathbb{E}_{x'\sim D}\left[\beta\, \text{KL}(\pi^*\|\pi_{\text{ref}})\right]) \\ \qquad \text{if } y \sim y_{\text{desirable}}|x \\ \sigma(\mathbb{E}_{x'\sim D}\left[\beta\, \text{KL}(\pi^*\|\pi_{\text{ref}})\right] - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}) \\ \qquad \text{if } y \sim y_{\text{undesirable}}|x \end{cases} \tag{7}$$

In practice, we found it optimal to have minibatches containing an equal number of desirable and undesirable examples, since the KL term was estimated within each minibatch; even minibatches as small as 4 examples per GPU caused no stability issues. The KL term for $y \sim y_{\text{desirable}}|x$ was estimated using the undesirable half, and vice-versa. This is because if a desirable y is used in the estimate of the KL term for other desirable examples, it is possible that the $\frac{dL_{\text{KTO}}}{dy}$ gradient is positive instead of negative (as intended). A similar argument holds for the undesirable examples in the batch. Thus for a minibatch of m examples with m different inputs x and a corresponding y that is (un)desirable, we get m losses.

We align the same suite of models as in section 3 on the same data with the KTO loss (see Figure 5). We find that:

1. At the 13B+ scale, we see LLM responses being as good or better than the human-chosen response y_w given x. Recall that it's possible to beat y_w because the human choice was made between two options and is by no means the perfect continuation of x.

2. SFT+KTO is competitive with SFT+DPO at all scales, despite not using pairs of preferences.

3. KTO alone is significantly better than DPO alone at the {13B,30B} scales. In fact, a KTO-aligned Llama-30B model is competitive with its SFT+KTO counterpart, despite not undergoing supervised finetuning first, and is the only alignment method of the ones we tested to show this behavior.

It is worth noting that these results understate the practical improvement that KTO has over DPO. In real-world settings, KTO will have access to far more data than DPO-like methods because it does not rely on paired preference data. For example, a retail company will have a lot of customer interactions and knowledge of whether they went well or poorly (i.e., $(x, y, \mathbb{1}[y \text{ is desirable}])$); they will have little counterfactual data of the type (x, y_w, y_l).

6 Archangel

We are releasing all 56 models we trained as the *Archangel* suite: {4 Pythia models + 3 Llama models} x {SFT, SLiC, DPO, SFT+DPO, PPO (offline), SFT+PPO (offline), KTO, SFT+KTO (offline)}.[1] The models were all trained and sampled under nearly identical settings (e.g., same random seed, same optimizer, same learning rate scheduler, effective batch size of 32, etc.). Hyperparameters unique to a model were set according to a sweep. Unsurprisingly, values of hyperparameters that had the same meaning across different loss functions (e.g., β in KTO and DPO) ended up having the same value. Because some methods relied on pairs of preferences and others did not, the order in which the training data was seen was different across the two kinds of losses (e.g., preference-based vs. preference-free) but identical within the same type of loss. The prompts used to sample generations for GPT-4 judgments were identical across all models. By aligning these 56 models in close-to-identical settings, we hope that the research community can better understand how the effectiveness of alignment evolves across different methods and at different scales.

[1] Models are available on Huggingface and our code is available on Github under ContextualAI/HALOs.

7 Future Work

The existence of HALOs as a distinct class of functions raises many interesting questions:

- Is there a human value function — and corresponding HALO — that better describes how humans see language? The KTO loss is based on the median human value function for monetary gains and losses, which is almost certainly different from how humans perceive the relative goodness/badness of text. So what does a human value function for language specifically look like? What is its median form and how does it vary across individuals?
- What differences in helpfulness/harmfuless emerge at different scales? All else constant, are feedback-aligned LLMs more likely to be sycophantic when they are larger (Perez et al., 2022), as some others have pointed out? Or is harmfulness more of an issue with smaller models, simply because they have a worse sense of what is good and bad?
- Given that the data that KTO needs is much more accessible, how far can we push synthetic data?For example, if we wanted to create a toxicity dataset to align our models to be less toxic, creating a tuple (x, y_w, y_l) where y_l is more toxic than y_w is tricky. However, with KTO, we can easily create a dataset $(x, y, \mathbb{1}[y \text{ is desirable}])$ where desirability is determined by some black-box toxicity detection API. The ability to align models with score-based data is a huge appeal of PPO, and KTO permits a binary version of this.

Acknowledgements

We thank Dilip Arumugam and Nathan Lambert for their feedback on this report.

References

Baheti, Ashutosh, Ximing Lu, Faeze Brahman, Ronan Le Bras, Maarten Sap, and Mark Riedl. 2023. Improving language models with advantage-based offline policy gradients. *arXiv preprint arXiv:2305.14718*.

Biderman, Stella, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

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

Ethayarajh, Kawin, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Ganguli, Deep, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

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

Köpf, Andreas, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. 2023. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

Peng, Xue Bin, Aviral Kumar, Grace Zhang, and Sergey Levine. 2019. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*.

Perez, Ethan, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. 2022. Discovering language model behaviors with model-written evaluations. *arXiv preprint arXiv:2212.09251*.

Rafailov, Rafael, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Zhao, Yao, 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*.