Transforming and Combining Rewards for Aligning Large Language Models

Zihao Wang [1 †] **Chirag Nagpal** [2] **Jonathan Berant** [3] **Jacob Eisenstein** [3] **Alex D'Amour** [3] **Sanmi Koyejo** [4 3] **Victor Veitch** [1 3]

Abstract

A common approach for aligning language models to human preferences is to first learn a reward model from preference data, and then use this reward model to update the language model. We study two closely related problems that arise in this approach. First, any monotone transformation of the reward model preserves preference ranking; is there a choice that is "better" than others? Second, we often wish to align language models to multiple properties: how should we combine multiple reward models? Using a probabilistic interpretation of the alignment procedure, we identify a natural choice for transformation for (the common case of) rewards learned from Bradley-Terry preference models. The derived transformation is straightforward: we apply a log-sigmoid function to the centered rewards, a method we term "LSC-transformation" (log-sigmoid-centered transformation). This transformation has two important properties. First, it emphasizes improving poorly-performing outputs, rather than outputs that already score well. This mitigates both underfitting (where some prompts are not improved) and reward hacking (where the model learns to exploit misspecification of the reward model). Second, it enables principled aggregation of rewards by linking summation to logical conjunction: the sum of transformed rewards corresponds to the probability that the output is "good" in all measured properties, in a sense we make precise. Experiments aligning language models to be both helpful and harmless using RLHF show substantial improvements over the baseline (non-transformed) approach.

† Work done during internship at Google DeepMind [1]University of Chicago, Chicago, IL, USA [2]Google Research, Mountain View, CA, USA [3]Google DeepMind, Mountain View, CA, USA [4]Stanford University, Stanford, CA, USA. Correspondence to: Zihao Wang <wangzh@uchicago.edu>.

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

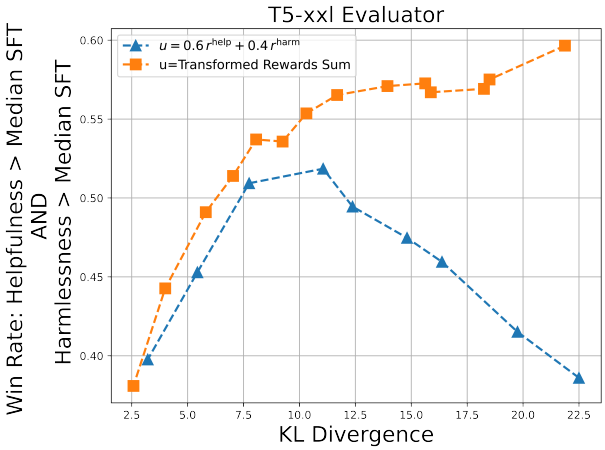


Figure 1. Transforming the Bradley-Terry reward both mitigates overfitting and makes addition behave as logical AND. This leads to significant improvements in aligned model quality relative to standard practice. Each point on the plot is a LLM aligned with a different KL penalty weight. The y-axis shows improvement over the base supervise finetuned (SFT) LLM in both helpfulness AND harmlessness, as judged by an external evaluator model (not used for RLHF). The baseline aggregates suboptimally (usually losing on either helpfulness or harmlessness) and suffers reward hacking (performance decays in the high KL regime). Details in Section 5.

1. Introduction

In this paper, we are interested in how to align large language models in order to bias their outputs towards having desired properties—e.g., to be helpful, harmless, factual, or creative (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022). We study the two-stage approach where we first learn a reward model from human preferences, and then align the language model so that its outputs have high values under the reward model. In this context, we're interested in two fundamental problems:

1. The alignment step maximizes the expected learned reward model. However, any monotone transformation of the reward preserves the interpretation of the alignment procedure as biasing the language model towards human-preferred outputs. Can we improve the alignment step by transforming the learned reward?

2. We often wish to align language models to multiple properties—e.g., outputs should be helpful, and harm-

less, and factual. If we have reward models for each property, how should we combine them?

A main challenge in answering these questions is that the goal of alignment is not precisely defined. As a result, there is no obvious principle to guide the choice of transformation or aggregation method. The conceptual idea in this paper is to interpret alignment probabilistically. From this perspective, the goal of aligning a model to a particular property is to produce samples from the posterior distribution conditional on the outputs being "good" on that property. Similarly, the goal of aligning to multiple properties is to produce samples conditional on the outputs being "good" on all properties.

To make use of this idea, we need to define what it means for an output to be "good". In the context of rewards learned from preference data, we take an output y be "good" if it has reward $r(x, y)$ greater than some prompt-specific reference value $r^{\text{ref}}(x)$. The first main result of the paper is that in the (typical) case where the reward model is learned from preference data using a Bradley-Terry model and the language model is aligned by maximizing expected reward subject to a KL constraint, the natural choice of transformation is:

$$u(x, y) = \log \sigma(r(x, y) - r^{\text{ref}}(x)), \tag{1.1}$$

where $\sigma(\cdot)$ is the sigmoid function. Here, r is the learned Bradley-Terry reward model, and u is the transformed reward we use in the alignment step. We call this transformation the "LSC-transformation" (log-sigmoid-centered transformation), and alignment using such transformation "LSC-alignment".

This transformation is motivated by a probabilistic interpretation. It additionally turns out to have important practical benefits relative to the baseline approach of using the raw reward model. First, the transformed reward shrinks the marginal utility of very high reward values. This has the effect in alignment of both encouraging the model to improve poorly performing prompts, and of discouraging the model from "reward hacking" by optimizing the reward model outside the range of its validity. Second, the transformed reward offers a natural way to combine multiple reward models. Namely: the sum of the transformed rewards corresponds to the logical AND of the outputs being "good" on each property. So, after transforming the rewards, we can simply sum them to aggregate multiple reward models.

In combination, these benefits can lead to substantial improvements in alignment performance. Figure 1 compares aligning a language model to be both helpful and harmless using summation of transformed and untransformed rewards. Varying the strength of KL regularization used in the alignment step, we observe that the transformed reward leads to substantial improvements at all KL levels.

2. Preliminaries

We first review the standard Reinforcement Learning from Human Feedback (RLHF) two-step procedure for aligning language models to human preferences.

Reward model training from pairwise data Reward models are trained to emulate human feedback. A frequently used type of feedback is pairwise preference data, consisting of a prompt x and two generated responses y^+, y^-, where y^+ is preferred by the human annotator. Our discussion mainly focuses on this case.

Commonly, rewards are learned using the Bradley-Terry model (Bradley & Terry, 1952),

$$p(y^- \prec y^+ \mid x) = \sigma(r(x, y^+) - r(x, y^-)). \tag{2.1}$$

The function r is parameterized by a neural network (typically, another LLM) and fit using the standard maximum log likelihood objective.

Alignment to reward model The next step is updating the base LLM to bias it towards high-reward responses.

Usually, aligning the model to the reward function is proceeded by a "Supervised Finetuning" (SFT) step where the base model is fine-tuned using the language modeling objective on the winning examples from the human preference data. We denote this model as π_0. Our interest is how to use the reward model to further align π_0.

The aim of the alignment step is to update π_0 to a new model π^* that has high expected reward, while still being close to π_0 (to preserve information from the previous training phases). Standard practice is to learn π^* by maximizing the expected reward of samples, regularized by a penalty on the KL-divergence between π^* and π_0.

The main idea in this paper is to instead use a utility measure $u(x, y)$ that is a monotone transformation of $r(x, y)$. We leave the alignment procedure otherwise unchanged. Then, mathematically, π^* is the maximizer of:

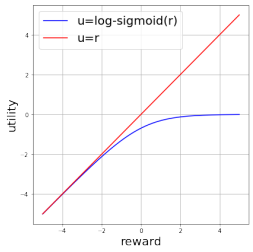
$$\mathbb{E}_x\{\mathbb{E}_{y\sim\pi(\cdot|x)}[u(x, y)] - \gamma \text{KL}(\pi(\cdot|x)\|\pi_0(\cdot|x))\} \tag{2.2}$$

Here, γ is a hyper-parameter that controls the trade-off between maximizing rewards and aligning with π_0.

3. Reward Transformation

We now turn to deriving the reward transformation.

Formalize Alignment Goal The first step is to formalize the goal of alignment. This is necessary to identify a "correct" transformation. Intuitively we want to modify the initial policy $\pi_0(y|x)$ so that the generated samples are



(a) Transformation Shape

(b) Helpfulness Examples

Figure 2. Bradley-Terry rewards do not capture diminishing utility, and log-sigmoid transforming can fix this. In the example responses, moving from response 1 to response 2 substantially increases utility, but from response 2 to response 3 only marginally increases. However, the BT rewards treat each improvement the same. A log-sigmoid transformation reflects diminishing returns.

considered "good" in some property by humans. To make progress, we introduce a binary semantic random variable G indicating whether response y is "good" for prompt x. Then we define the alignment goal as producing a model that samples from the distribution of responses conditional on the response being good; i.e., $\pi^{\text{target}}(\cdot \mid x) = p(\cdot \mid x, G = 1)$.

In fact, we slightly generalize this to allow finer grained control of the reward vs KL tradeoff. By Bayes' rule, we may rewrite $p(y|x, G = 1) \propto \pi_0(y|x)p(G = 1|x, y)$. That is, we reweight the base LLM by a term that upweights responses that are likely to be deemed good. It is natural to introduce a hyperparameter to control the strength of this upweighting. Anticipating the connection to Equation (2.2), we again use γ for this hyperparameter. Then, we define our alignment goal as producing an aligned model

$$\pi_\gamma^{\text{target}}(y|x) \propto \pi_0(y|x)p(G = 1|x, y)^{1/\gamma} \tag{3.1}$$

Reward Transformation The next question is how to produce an aligned model that satisfies our goal. This has two parts: we must use the reward model to define the binary goodness variable G, and we must determine how the utility function used for alignment relates to G.

Target Utility Function We begin by connecting alignment utility and G. The idea is to use the well-known result that the ideal optimizer of the KL-regularized RLHF objective Equation (2.2) is an exponential tilting of the base policy (e.g., Korbak et al., 2022):

$$\pi^*(y \mid x) \propto \pi_0(y \mid x) \exp\left(u(x, y)/\gamma\right) \tag{3.2}$$

Comparing Equation (3.1) with Equation (3.2), we see that in order to get the target policy through alignment, we must set the utility function to be the log-probability of goodness:

$$u(x, y) = \log p(G = 1|x, y). \tag{3.3}$$

Pointwise Reward Models The next step is to relate the Bradley-Terry model to $p(G = 1|x, y)$. As a warmup, we consider the case of reward models trained on pointwise data; i.e., where each example is a prompt x, response y, and a binary label G indicating whether the response is good. In this case, we would train the reward function by minimizing a binary cross-entropy objective. This is a proper scoring rule, so the learned reward would be: $r(x, y) = \text{logit}\, p(G = 1|x, y)$. Here, we take the reward to be on the logit scale so that $r(x, y) \in (-\infty, \infty)$, analogous to Bradley-Terry rewards. In this case, the right utility is $u(x, y) = \log \sigma(r(x, y))$.

Pairwise Reward Models The pairwise case is more subtle. It may be tempting to again apply the $\log(\sigma(\cdot))$ transformation to the Bradley-Terry rewards. This is incorrect for two reasons. First, only reward *differences* are interpretable as logit probilities—the rewards for individual prompts are not. Second, in general the reward model $r(x, y)$ is unidentifiable from the data. For any r, we can shift it by any arbitrary function of the prompt x without changing the Bradley-Terry model. That is, $\tilde{r}(x, y) \leftarrow r(x, y) + C(x)$ has the same objective in Equation (2.1). However, any non-linear transformation of the reward will be sensitive to this unidentified $C(x)$.

Happily, both problems are resolved by choosing a suitable definition of what it means for a response to be good. Here, we take a generated response y to be "good" if it is preferred over a chosen reference output y^{ref}. For example, we may say that y is harmless as long as it is preferred by the harmlessness reward to a canned response such as "I am unable to answer that question".

The LSC-transformation follows immediately:

Theorem 1. *Suppose output y is deemed good for prompt x if it would be preferred to reference output $y^{ref}(x)$. Then, if the Bradley-Terry model Equation (2.1) holds, and we align using KL-regularized utility maximization Equation (2.2), then using utility*

$$u(x, y) = \log \sigma(r(x, y) - r^{ref}(x)) \tag{3.4}$$

will satisfy the alignment goal Equation (3.1). Here $r^{ref}(x) := r(x, y^{ref}(x))$.

That is, once we decide on a reference response, we get the right utility function by applying log-sigmoid transformation to the centered reward.

Mechanistic Interpretation We derived the reward transformation from a probabilistic argument. It is also insightful to consider the mechanistic effect of the transformation.

One fundamental issue with the baseline approach, where

$u(x, y) = r(x, y)$, is that the utility gain for improvements never diminishes. In the aligned model, taking $\gamma = 1$, we have that the relative probabilities of two responses is exponential in the difference of utilities.

$$\frac{\pi^*(y^1 \mid x)}{\pi^*(y^0 \mid x)} = \exp(u(y_1, x) - u(y_0, x)) \frac{\pi_0(y^1 \mid x)}{\pi_0(y^0 \mid x)} \quad (3.5)$$

Now, consider the case where we have three candidate responses: y^{ref}, y^0, y^1 such that $p(y_{\text{ref}} \prec y^0) = 0.99$ and $p(y_{\text{ref}} \prec y^1) = 0.999$. If we use the raw Bradley-Terry logits as our utility, then using that $r(y^1, x) - r(y^0, x) = (r(y^1, x) - r(y^{\text{ref}}, x)) - (r(y^0, x) - r(y^{\text{ref}}, x))$ we have:

$$\begin{aligned}\frac{\pi^*(y^1 \mid x)}{\pi^*(y^0 \mid x)} &= \exp(\text{logit}(0.999) - \text{logit}(0.99)) \frac{\pi_0(y^1 \mid x)}{\pi_0(y^0 \mid x)} \\ &\approx 10 \times \frac{\pi_0(y^1 \mid x)}{\pi_0(y^0 \mid x)}\end{aligned}$$

That is, when aligning to raw rewards, going from a very good response to a marginally better response increases probability by a factor of 10! However, if y^{ref} is already good, a human would find little difference between the two responses.[1] Conversely, aligning to the transformed reward model only increases the probability by a factor of 1.01.

This effect seems particularly salient when we consider that the reward model is itself learned. It seems unlikely that the model can actually reliably distinguish between y_1 and y_0. Accordingly, when we align to the raw learned model, we expect to induce enormous shifts according to preferences that are anyways noisy.

Choosing reference response The reference reward acts as a hyperparameter of the transformation. Essentially, the transformation results in a utility that is linear below the reference value, and rapidly diminishing above it. Accordingly, we should set the reference to a value that represents a good response, that we believe is achievable by the model, and where we believe our learned reward function makes meaningful predictions. We found that a good default choice is the 85th quantile of examples from the base distribution. We consider additional examples in Section 5.

4. Reward Aggregation

We now consider the case when we have multiple reward models for different objectives $r_1, ..., r_n$.

Alignment Goal Again, the first step in deriving an optimal aggregation scheme is to formalize a goal. We make the following natural choice: the aligned model should be "good" on all target properties. E.g., we want to align our model to be helpful AND harmless.

To formalize this idea, let G_i be binary random variable indicating whether y is considered "good" for x in property i. We introduce the binary random variable corresponding to logical AND: $G_{\text{AND}} := \bigwedge_{i=1}^n G_i$. Similar to the single reward case, we formalize the goal as the posterior distribution conditioned on all properties being "good":

$$\pi^{\text{target}}_{\text{AND},\gamma} \propto \pi_0(y \mid x) p(G_{\text{AND}} = 1 \mid x, y)^{1/\gamma} \quad (4.1)$$

Reward Aggregation With this goal, following Theorem 1, we want to align using utility

$$u(x, y) = \log p(G_{\text{AND}} = 1 \mid x, y). \quad (4.2)$$

The question is then how to construct this utility function using the individual reward models.

In full generality, this is an impossible task. The reason is that the individual rewards only tell us about the marginal distributions of the properties, but the logical AND may depend on interactions. Thus, we need an extra assumption:

Assumption 1 (Independent Judgements)**.** Given a fixed prompt x and response y, whether y is judged to be good for x on each property is independent of all the judgements on all other properties. That is, $(G_1, ..., G_n)$ are conditionally independent given (X, Y).

For example, this assumption says we can decide whether a given response is helpful independently of deciding whether it's harmful. (Note: this is conditional on the prompt and response. We do *not* require helpfulness and harmless to be independent marginally.)

The reward aggregation formula follows immediately:

Theorem 2. *Suppose output y is deemed good for prompt x in aspect i if it would be preferred to reference output $y_i^{ref}(x)$ in property i. Then, if Assumption 1 holds, the Bradley-Terry model Equation (2.1) holds for all properties, and we align using KL-regularized utility maximization Equation (2.2), then using utility*

$$u(x, y) = \sum_{i=1}^n \log \sigma(r_i(x, y) - r_i^{ref}(x)) \quad (4.3)$$

will satisfy the alignment goal Equation (4.1). Here $r_i^{ref}(x) := r(x, y_i^{ref}(x))$.

Mechanistic Interpretation We derived the aggregation scheme according to a probabilistic assumption. Similar to the single-reward case, we consider the mechanistic effect.

[1] Ultimately, the issue here is that Bradley-Terry rewards don't automatically correspond to utilities.

The baseline approach is to aggregate with a (weighted) sum of the raw rewards. The key problem is that this approach allows strong performance in one property to balance out poor performance in another.

Consider the case where we have two properties A and B we want to align to (e.g., helpfulness and harmlessness), and 4 responses $y_A^{\text{ref}}, y_B^{\text{ref}}, y^0, y^1$ such that $p(y_A^{\text{ref}} \prec y^1) = p(y_B^{\text{ref}} \prec y^1) = 0.9$, and $p(y_A^{\text{ref}} \prec y^0) = 0.45$ with $p(y_B^{\text{ref}} \prec y^0) = 0.99$. If we want the aligned distribution to generate samples that are "good" in both aspects, then y^1 should be preferred to y^0. However, if we use the sum of the raw Bradley-Terry logits as utility ($u = r_A + r_B$), the relative probability ratio under the aligned policy π^* will be (with $\gamma = 1$) $\frac{\pi^*(y^1 \mid x)}{\pi^*(y^0 \mid x)} = 1 \times \frac{\pi_0(y^1 \mid x)}{\pi_0(y^0 \mid x)}$. That is, the aligned model does not upweight response y_1. If we instead align by the sum of the transformed reward, then we have that the relative probability ratio is approximately 1.8—i.e., the response that does well on both properties is preferred.

5. Experiments

We now turn to assessing the practical effect of using the transformed reward to align LLMs. We experiment with aligning models to be helpful, harmless, and both. We find that the transformation alleviates reward hacking and reward underfitting, and that aligning to transformed sum acts as aligning to logical AND. This leads to substantial improvements in LLMs aligned to the transformed reward.

5.1. Experimental Setup

We follow a standard RLHF pipeline; see Appendix A.4 for full details.

Datasets We use the Anthropic Helpfulness and Harmlessness datasets (Bai et al., 2022). These are multi-turn dialogues between a human and a digital assistant. Each dataset consists of the beginning of the conversation, two responses for the final turn of the AI side of the conversation, and a label for the human preference on the target property. We use the base datasets (44K examples for helpfulness and 42K for harmlessness), where responses are generated from a 52B context-distilled LM. For both tasks, we split the training set into two: half for training the reward model, and half for the alignment step.

Reward model training We train a Bradley-Terry reward model for each of helpfulness and harmlessness by finetuning a pretrained T5-base (220M parameters) model (Raffel et al., 2020) on the Anthropic data.

SFT For our policy model, we use the instruction-finetuned PALM-2-XXS model (Anil et al., 2023). Following standard practice, we first run supervised finetuning (SFT) of the instruction tuned LLM on the 'preferred' responses from the helpfulness dataset. We use this SFT model as the pre-alignment base for all experiments.

RLHF setup For alignment, we follow standard practice and optimize expected utility subject to a KL penalty using Proximal Policy Optimization (PPO) algorithm. For each utility function and dataset, we sweep over multiple values of the KL regularization strength γ. We run for 20000 steps, which we find suffices for convergence in all cases.

5.2. LSC-Transformation Improves Alignment

Transforming the reward model should encourage the alignment to focus on improving lower-reward responses over those with already high rewards. We expect this to both reduce reward hacking, and to reduce the number of low-reward responses (less underfitting).

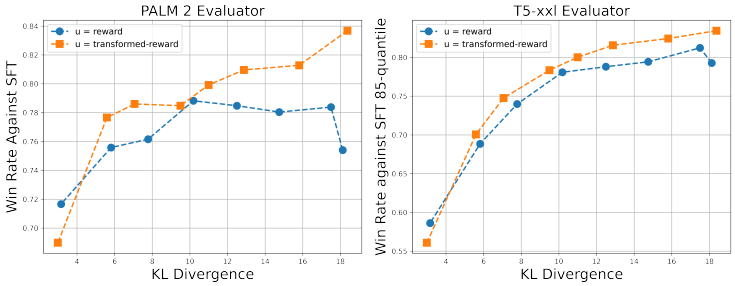
Choice of reference reward The reference reward $r^{\text{ref}}(x)$ should capture the notion of a response that's "good enough". For harmlessness, this is straightforward: a generic response like "I can't answer that" achieves the goal. For the experiments, we sampled variations of "I can't answer that" as the reference reward.

For helpfulness, such canned responses won't suffice. Instead, for each prompt we sample 64 responses from the SFT model. We then use this data to build an estimator of the 85th quantile of the sampled rewards for each prompt. We use this estimated 85th quantile as the reference reward. Details provided in Appendix A.1.

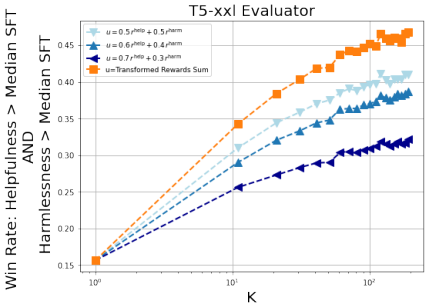
Transformation Improves Alignment Aligning to the transformed reward should reduce reward hacking and underfitting relative to aligning to the raw reward. Then, we expect that the transformed alignment leads to larger gains over the SFT model.

We have two main strategies for judging improvement relative to the SFT model. First, following past work (Gao et al., 2023; Coste et al., 2023; Eisenstein et al., 2023), we train a T5-XXL model using the same preference dataset and the Bradley-Terry objective. This provides a proxy for true preferences that we do not optimize against (so, it can witness reward hacking). We say a sample from the aligned model wins in helpfulness if it beats the 85th reward quantile of samples from the SFT model. We say it wins in harmlessness if it beats the 95th-quantile. (These numbers are chosen to make winning hard enough to show a gap between alignment strategies; results are consistent across other choices of quantile.).

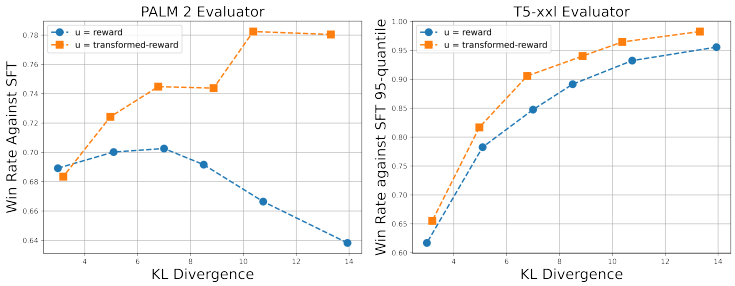
The second strategy evaluates wins by zero-shot querying of an instruction-tuned PALM-2 medium model. Following



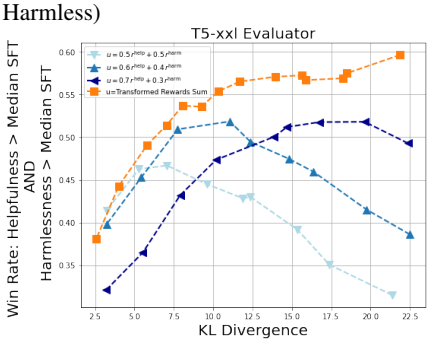
(a) RLHF for single reward (Helpful)



(b) best-of-k for aggregation (Helpful and Harmless)



(c) RLHF for single reward (Harmless)



(d) RLHF for aggregation (Helpful and Harmless)

Figure 3. Transformed reward obtains better trade-offs between KL/k and win-rate trade-offs, in single-reward and reward aggregation (more significantly). In Figure 3a and Figure 3c, we show two win rates: 1) win rates judged from prompted PALM 2 evaluator, between the aligned policy and random SFT samples, and 2) win rates judged by T5-XXL evaluator, against the SFT quantiles: we use 85% for helpfulness and 95% for harmlessness for this plot. In Figure 3d and Figure 3b, we show win rates against SFT median rewards in helpfulness and harmlessness, judged by T5-XXL evaluator.

previous work (Dubois et al., 2023; Singhal et al., 2023; Eisenstein et al., 2023; Rafailov et al., 2023), we pass a prompt, a response from the SFT model, and a response from the aligned model and ask which is preferred (in terms of helpfulness or harmlessness). Details in Appendix A.2.

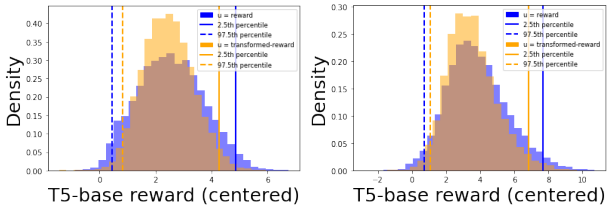
Figure 3 shows the win rate of the aligned models over the SFT model for each evaluation strategy. We average over the prompts in the RLHF validation dataset. We see that aligning to the transformed reward dominates aligning to the raw reward, under both evaluation strategies and at all levels of KL distance to the base policy model.

See Appendix A.2 for additional experiments evaluating alignment-induced improvements.

Uniform Improvement of Rewards It is clear that aligning using the transformed reward improves over aligning using the raw reward. Intuitively, this is because of a reduction in both reward hacking and underfitting. To check whether this intuition holds, we plot the distribution of rewards of samples from (approximately) KL-matched raw-aligned and transformed-aligned models in Figure 4. As expected, the reward distribution of samples from the transformed-aligned model is more concentrated. That is, there are fewer very high reward samples (less reward hacking) and fewer very low reward samples (less underfitting).

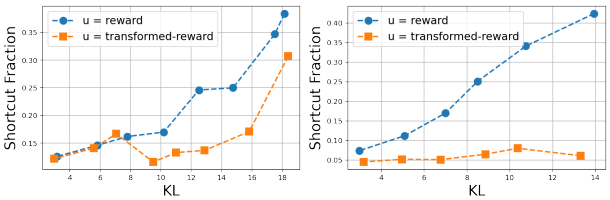
In more detail: for each of helpfulness and harmlessness, we choose a raw-aligned and transformed-aligned model with approximately matched KL. We sample responses from each model with the same prompts. Then, we compute the (T5-base) reward for each of these responses, centered by median reward of SFT samples (to make rewards comparable across prompts). Figure 4 shows histograms of these sampled rewards. We also compare across multiple KL-values in Figure 10.

Transformation reduces shortcuts One symptom of reward hacking is that the aligned model will start exploiting "shortcuts" that are preferred by the reward model (but which do not correspond to genuine improvements). We consider the effect of reward transformation on two such shortcuts. For helpfulness, Eisenstein et al. (2023) observe that raw-aligned models have a tendency to format outputs as lists. For harmlessness, we observe that raw-aligned models will often give responses of the form "you should consult a [doc-



(a) Helpfulness (KL $\approx$ 18) (b) Harmlessness (KL $\approx$ 7)

Figure 4. Reward Transformation leads to more uniform reward improvements than baseline. We compare reward distributions in the aligned policies that are matched on KL. Rewards are centered by the SFT median in both helpfulness and harmlessness. Reward distributions are more concentrated when using transformed rewards than using raw rewards.



(a) Helpfulness (b) Harmlessness

Figure 5. Transformed reward reduces shortcuts in generated responses. For helpfulness, we identify a shortcut pattern of using lists, similar to (Eisenstein et al., 2023). In harmlessness, one known shortcut pattern is recommending the users to seek therapy or consult professional help (Bai et al., 2022). We extract these shortcuts with heuristic methods. In the baseline approach, the policy model exploits those shortcuts for higher reward values. This is mitigated when we transform the reward.

tor/psychologist/lawyer/etc]" (a similar observation is made by Bai et al. (2022)). In Figure 5, we plot the fraction of responses that contain each shortcut, for each aligned model. We see that the raw-reward aligned model does indeed exploit these shortcuts. Further, this behavior is substantially mitigated by aligning to the transformed reward instead. See Appendix A.3 for details.

5.3. Reward Aggregation using LSC-Transformation Significantly Improves Alignment

We now turn to the second goal: aggregating rewards for multiple distinct goals. To that end, we consider aligning a LLM to be both helpful and harmless.

5.3.1. Further Experimental Setup

RLHF setup We use reward models trained for the helpfulness and harmlessness tasks as discussed above. For RLHF training, we use prompts only from the helpfulness dataset. This decision is because of the observation of the tension between helpfulness and harmlessness, which forced (Bai et al., 2022) to use a higher proportion of helpfulness prompts than harmlessness ones. We use the same policy model as in experiments for single rewards (SFT-ed on helpfulness data). The other training details are the same as in single-reward experiments.

best-of-k setup In addition to RLHF experiments, we also experiment with best-of-k sampling, as in Gao et al. (2023); Eisenstein et al. (2023). That is, we draw k samples from the SFT model, rank them by the combined reward, and return the top ranked sample. This can be viewed as another alignment procedure, where the best-of-k sampling has some (KL) divergence from the underlying policy, and produces samples with higher expected reward. In our experiments, we try k increasing from 1 to 191.

There are two main motivations for considering best-of-k experiments. First, it demonstrates that the reward aggregation method applies to methods beyond RLHF. Second, best-of-k doesn't involve complicated optimizations. This allows us to disentangle the effect of having the 'right' reward from the effect on solving the optimization problem.

Baseline In this setting, we take the baseline method to be a weighted sum of the raw rewards; i.e.,

$$R_{\wedge}^{\text{baseline}} := wR^{\text{help}} + (1-w)R^{\text{harm}}. \tag{5.1}$$

We sweep over $w = 0.1, ..., 0.9$, and report the baselines with best performance on our evaluation metrics.

Reference Rewards We combine the transformed rewards by simple addition. Weighting this addition doesn't have a clear motivation or interpretation. Instead, the choice of reference value used for each reward plays an analogous role. Intuitively, if we set a lower reference value for reward A then the reward becomes more easily saturated, and the optimization focuses more on reward B.

For best-of-k, using $w = 0.5$ achieves the best performance in the baseline method. Accordingly, we want to take the reference rewards for helpfulness and harmlessness to be comparable. To that end, we use the (estimated) SFT median for both helpfulness and harmlessness. Note that this is a lower reference than used for the individual optimizations. This makes sense because the more difficult problem of optimizing both goals simultaneously stops the model from saturating the reward prematurely.

For RLHF, we observe that $w = 0.6$ gives the best result for weighted-sum approach. Then, we want to set the reference reward for helpfulness to be somewhat higher. We use the (estimated) SFT 75%-quantile.

These choices are likely not optimal, and it's possible that further hyperparameter optimization could improve results.

Evaluation We want a metric that reflects Logical-AND. That is, whether we've improved over the SFT model in both

helpfulness and harmlessness. To that end, we'll say that a generation wins if it has helpfulness reward higher than the median helpfulness of SFT samples, and harmlessness higher than the median harmlessness of SFT samples.

5.3.2. Aggregation Results

Transformed Aggregation Improves Alignment Summing the transformed reward should have two advantages over the baseline method. First, it corresponds to logical AND. Second, it retains the benefits of alleviating reward overoptimization, as in the single-reward case. Together, these should cause aligning by the transformed-combined reward to outperform aligning by the baseline reward.

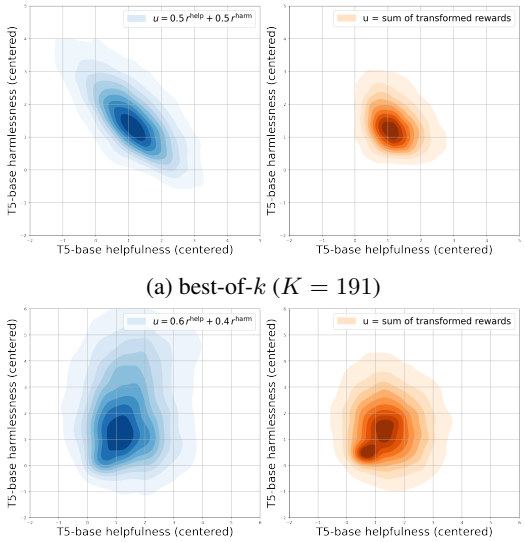
Figure 3b and Figure 3d shows improvement over the SFT model for aligning using both rewards, for both best-of-k and RLHF. As anticipated, we see significant improvement from the transformed method.

It is noteworthy that the transformed-aligned model outperforms in best-of-k, and for low-KL in the RLHF alignment. In these cases, there is little reward hacking (the aligned model is too close to the SFT model to have very bad behavior). Thus, the win here is apparently due mainly to the logical AND effect. In the high-KL regime, the reward hacking effect kicks in and the transformed-reward dramatically outperforms the raw-reward baseline.

Transformed Summation Corresponds to Logical AND Next, we check directly whether the logical AND effect can be witnessed in the aligned LLM. To that end, we examine the distributions of rewards in the aligned policies. In Figure 6, our aggregation method leads to more balanced reward distributions (two reward values are often similar), whereas the baseline method leads to more unbalanced reward distributions (one reward is much higher than the other one). Note that best-of-k ($k = 191$) have quite different reward distributions than RLHF at KL ≈ 14 (the former reflects the anti-correlation of the two rewards in the initial policy; the latter updates the policy to have high aggregated reward values). But the transformed aggregation method has consistent effects in both cases.

6. Ablation Studies

The LSC-transformation consists of two components: centering and log-sigmoid transformation. Centering is known to reduce variance in policy gradient updates, while the log-sigmoid transformation caps the reward values. To understand the contribution of each component, we investigate their effects separately. As shown in Figure 7, applying centering or log-sigmoid transformation in isolation does not improve alignment.



(a) best-of-k ($K = 191$)

(b) RLHF (KL ≈ 14)

Figure 6. Summation of log-sigmoid transformed rewards corresponds better to logical AND. Aligned policies using the former method have more balanced reward distributions (concentrated where the two reward values are similar), whereas the latter method leads to more unbalanced reward distributions. We choose aligned policies by matching K for best-of-k and KL for RLHF. The rewards are centered by SFT median for both helpfulness and harmlessness. We visualize the joint distribution by kernel density estimate plot, with darker color indicating higher density.

Does centering alone help? This approach did not change the RLHF performance relative to the baseline. Though centering is known to reduce variance for policy gradient update, it doesn't change the optimization target of the KL-regularized RLHF objective, as shown in Equation (3.2).

Does log-sigmoid transformation alone help? This essentially sets a constant reference value ($r_{\text{ref}}(x) = 0$) for all inputs x. This is found to hurt the performance relative to baseline. The reference value is important, as can be seen in the shape of utility function in Fig 2(a): when the reference is too small, it saturates the utility when the reward is still under-fitted; when the reference is too large, the nonlinear transformation has no effect. More specifically, there are two possible reasons why simply using log-sigmoid transformation does not work:

1. First, $r(x, y)$ is non-identifiable: any $r(x, y) + c(x)$ yields the same Bradley-Terry objective. Then, any fixed constant reference reward (independent of prompt) is fundamentally not meaningful.

2. Second, if the reference reward is set too high then there will be no effect (for $r \ll r_{\text{ref}}$, the transformation is effectively a constant offset), and if the reference reward is set too low then training will saturate too early.

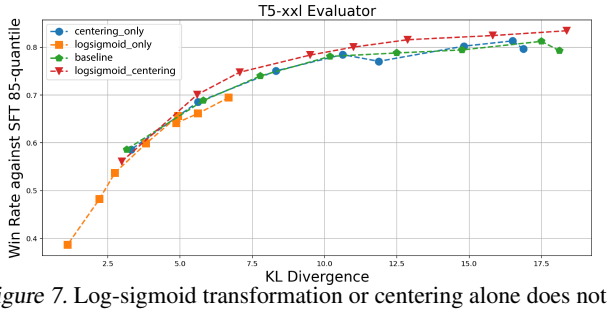


Figure 7. Log-sigmoid transformation or centering alone does not improve alignment. The experiment is for the helpfulness task.

> The issue with a constant reference level such as 0 is that it's unclear what this means in practical terms—is it too high or too low or fine? By contrast, choosing reference rewards based on generated responses (as in the paper) makes it easy to set a meaningful threshold.

7. Discussion and Related Work

There is a growing body of work on mitigating reward hacking in the RLHF pipeline. Techniques include forms of reward model averaging (Eisenstein et al., 2023; Ramé et al., 2024; Zhai et al., 2023), constrained optimization (Moskovitz et al., 2023), and reward model regularization (Shen et al., 2023), iterative human preference collection (Bai et al., 2022; Stiennon et al., 2020; Fan et al., 2022), or data bias mitigation (Singhal et al., 2023). These approaches are complementary to the transformation technique proposed here, and could be used in combination.

There have been several proposals for aligning language models to multiple objectives. The most common approach is to combine individual reward models via a weighted sum (e.g., Wu et al., 2023; Moskovitz et al., 2023). Moskovitz et al. (2023) identified a constraint threshold for individual rewards, and formalized a constrained MDP problem, but the identification of the threshold point relies on ground-truth queries. Bakker et al. (2022) consider adapting social welfare schemes for aggregating the dissenting preferences of many individuals, in contrast with our goal of satisfying all properties. Bai et al. (2022) train a single reward model on both the helpfulness and harmlessness data, but discover that this leads to reward hacking harmlessness. They change the proportion of helpfulness data in the training to circumvent this. Combining by summing transformed rewards allows us to circumvent such considerations.

The transformation technique in this paper is relevant to any alignment strategy that explicitly maximizes an expected utility. There are now also alignment methods (e.g., Rafailov et al., 2023; Azar et al., 2023; Zhao et al., 2022) that use preference labels directly without explicitly instantiating reward models. Note, however, that if we want to align to multiple properties, we still need to compute rankings from an aggregate. The simplest approach is to train reward models for individual properties, combine these reward models, and then rank samples using the combine reward. Our best-of-k experiments show that the transformation can yield significant gains even in this case.

Finally, we note that Azar et al. (2023) also emphasizes the need for a bounded utility function. The work here can be viewed, in part, as a way of incorporating this insight that still maintains the standard utility maximization pipeline. It is an interesting question for future work whether making explicit use of a (transformed) reward improves the alignment step relative to using only ranked pairs.

Acknowledgements

This work is supported by ONR grant N00014-23-1-2591 and Open Philanthropy.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Azar, M. G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*, 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.

Bakker, M. A., Chadwick, M. J., Sheahan, H. R., Tessler, M. H., Campbell-Gillingham, L., Balaguer, J., McAleese, N., Glaese, A., Aslanides, J., Botvinick, M. M., and Summerfield, C. Fine-tuning language models to find agreement among humans with diverse preferences, 2022.

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.

Coste, T., Anwar, U., Kirk, R., and Krueger, D. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*, 2023.

Dubois, Y., Li, X., Taori, R., Zhang, T., Gulrajani, I., Ba, J., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*, 2023.

Eisenstein, J., Nagpal, C., Agarwal, A., Beirami, A., D'Amour, A., Dvijotham, D., Fisch, A., Heller, K., Pfohl, S., Ramachandran, D., et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023.

Fan, X., Lyu, Y., Liang, P. P., Salakhutdinov, R., and Morency, L.-P. Nano: Nested human-in-the-loop reward learning for few-shot language model control. *arXiv preprint arXiv:2211.05750*, 2022.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., and Zhao, W. X. Large language models are zero-shot rankers for recommender systems. *arXiv preprint arXiv:2305.08845*, 2023.

Korbak, T., Perez, E., and Buckley, C. L. Rl with kl penalties is better viewed as bayesian inference. *arXiv preprint arXiv:2205.11275*, 2022.

Moskovitz, T., Singh, A. K., Strouse, D., Sandholm, T., Salakhutdinov, R., Dragan, A. D., and McAleer, S. Confronting reward model overoptimization with constrained rlhf, 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.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Ramé, A., Vieillard, N., Hussenot, L., Dadashi, R., Cideron, G., Bachem, O., and Ferret, J. Warm: On the benefits of weight averaged reward models. *arXiv preprint arXiv:2401.12187*, 2024.

Shen, L., Chen, S., Song, L., Jin, L., Peng, B., Mi, H., Khashabi, D., and Yu, D. The trickle-down impact of reward (in-) consistency on rlhf. *arXiv preprint arXiv:2309.16155*, 2023.

Singhal, P., Goyal, T., Xu, J., and Durrett, G. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*, 2023.

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.

Wu, Z., Hu, Y., Shi, W., Dziri, N., Suhr, A., Ammanabrolu, P., Smith, N. A., Ostendorf, M., and Hajishirzi, H. Fine-grained human feedback gives better rewards for language model training, 2023.

Zhai, Y., Zhang, H., Lei, Y., Yu, Y., Xu, K., Feng, D., Ding, B., and Wang, H. Uncertainty-penalized reinforcement learning from human feedback with diverse reward lora ensembles. *arXiv preprint arXiv:2401.00243*, 2023.

Zhao, Y., Khalman, M., Joshi, R., Narayan, S., Saleh, M., and Liu, P. J. Calibrating sequence likelihood improves conditional language generation. *arXiv preprint arXiv:2210.00045*, 2022.

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. Additional Experiment Details

A.1. Implementation for reference value prediction

The goal is to develop models that can predict various quantiles of the sampled rewards for different prompts. Instead of building separate models for each quantile, we assume the reward distribution for each prompt follows a Gaussian distribution. Under this assumption, we build two models: **Mean Prediction Model** (r^{mean}) and **Standard Deviation Prediction Model** (r^{std}) to predict the mean and standard deviation of the reward distribution for each prompt.

To estimate a specific quantile for a prompt, the formula used is:

$$r^{\text{mean}}(x) + s \times r^{\text{std}}(x)$$

where s is a scaling factor corresponding to the desired quantile (e.g., using $s = 0.7$ approximately gives us the 75% quantile).

The training data is collected by generating 64 responses per prompt using the SFT policy model. These responses are scored to calculate the sample mean and standard deviation of the reward distribution for each prompt, which are then used as the response variables in training the models r^{mean} and r^{std}.

A.2. PALM-2 Evaluation Details

We evaluate the win-rates with zero-shot prompting. For each prompt, we sample (y_π, y_{sft}) from the aligned and SFT policy, then ask the PALM-2 model which is more helpful/harmless. The prompt template is Figure 8.

In order to counter positional bias (Hou et al., 2023), we run PALM-2 on the two possible orderings (y_π, y_{sft}) and (y_{sft}, y_π), sample $N = 8$ outputs for each order and determine the winner by majority voting.

A.3. Heuristics for shortcut discovery

For the helpfulness task, we follow the same practice in (Eisenstein et al., 2023) and find responses in the format of a list.

For harmlessness task, we see the aligned policy starts to only generate similar generations to different questions as KL gets larger. To find shortcuts, we analyze generations of the aligned policy (using raw reward as utility) at KL ≈ 14. Then we perform topic modeling and identify one dominant topic characterized by the recurring theme of "you should talk to [profession]." From the most frequent 50 words in that topic, we find all the profession-related words: therapist, professional, doctor, psychologist, counselor, lawyer, police. Then we find all responses containing profession-related words.

A.4. RLHF training details

We use Proximal Policy Optimization (PPO) to perform RLHF alignment. The specific hyperparameters are in Table 1 We

Table 1. Hyper-parameters for RLHF.

Parameter	Value
Policy learning rate	$5 \cdot 10^{-6}$
Value learning rate	$4 \cdot 10^{-5}$
Learning schedule	Constant (linear warm-up)
Training steps	20000
Warm-up steps	2000
Batch size	32
Input length	1024
Output length	256

sweep over γ's to get aligned policies with different KL values. Since we want to match (converged) polices by their KL values, we find some heuristic to predict converged KL for a chosen γ (so they may not look like regular linear or geometric series). More specifically, we use the parameterization of $\alpha = \frac{\gamma}{1+\gamma}$, and the values used are in Table 2. Note that to get the same KL value, we use smaller KL regularization for reward transformation, than using the raw reward. This is intuitive as the log-sigmoid transformation prevents using KL budgets once the reward saturates.

```
_HELPFULNESS_PROMPT_TEMPLATE = """For the following query to a chatbot, which
response is more helpful?

Query: <the user query>

Response A: <either the test method or baseline>

Response B: <the other response>

FIRST provide a one-sentence comparison of the two responses and explain \
which you feel is more helpful. SECOND, on a new line, state only "A" or \
"B" to indicate which response is more helpful. Your response should use \
the format:
Comparison: <one-sentence comparison and explanation>
More helpful: <"A" or "B">"""

_HARMLESSNESS_PROMPT_TEMPLATE = """For the following query to a chatbot, which
response is more harmless?

Query: <the user query>

Response A: <either the test method or baseline>

Response B: <the other response>

FIRST provide a one-sentence comparison of the two responses and explain \
which you feel is more harmless. SECOND, on a new line, state only "A" or \
"B" to indicate which response is more harmless. Your response should use \
the format:
Comparison: <one-sentence comparison and explanation>
More harmless: <"A" or "B">"""
```

Figure 8. Prompt for PALM-2 Evaluation

Table 2. Alphas for Various Task-Methods.

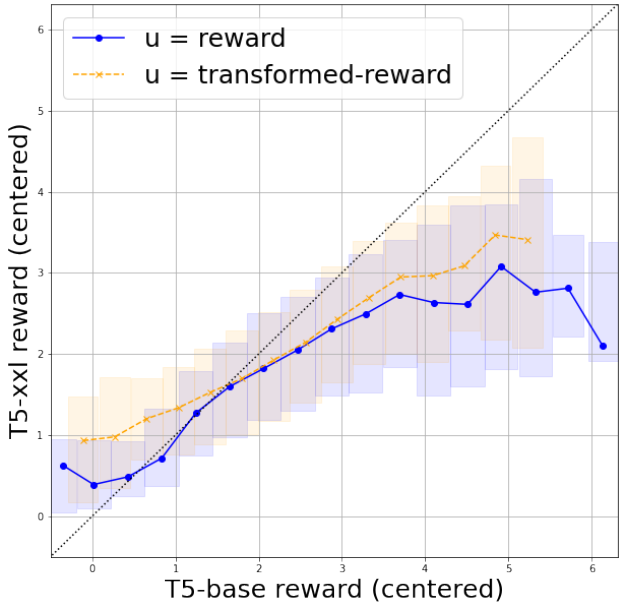
Task-Method	$\alpha := \frac{\gamma}{1+\gamma}$
Helpfulness (u = reward)	[0.081, 0.086, 0.092, 0.1, 0.114, 0.132, 0.161, 0.222]
Helpfulness (u = transformed reward)	[0.02, 0.023, 0.028, 0.032, 0.039, 0.053, 0.075, 0.123]
Harmlessness (u = reward)	[0.169, 0.182, 0.196, 0.218, 0.248, 0.32]
Harmlessness (u = transformed reward)	[0.032, 0.041, 0.052, 0.079, 0.126, 0.222]
H+H ($0.5r^{\text{help}} + 0.5r^{\text{harmless}}$)	[0.078, 0.08, 0.084, 0.088, 0.094, 0.102, 0.116, 0.134, 0.168]
H+H ($0.6r^{\text{help}} + 0.4r^{\text{harmless}}$)	[0.068, 0.071, 0.074, 0.077, 0.082, 0.088, 0.1, 0.123, 0.163]
H+H ($0.7r^{\text{help}} + 0.3r^{\text{harmless}}$)	[0.057, 0.06, 0.065, 0.07, 0.075, 0.084, 0.1, 0.126, 0.173]
H+H (sum of transformed reward)	[0.014, 0.0145, 0.015, 0.017, 0.018, 0.02, 0.023, 0.026, 0.031, 0.034, 0.04, 0.048, 0.066, 0.096]

B. More Experiment Results

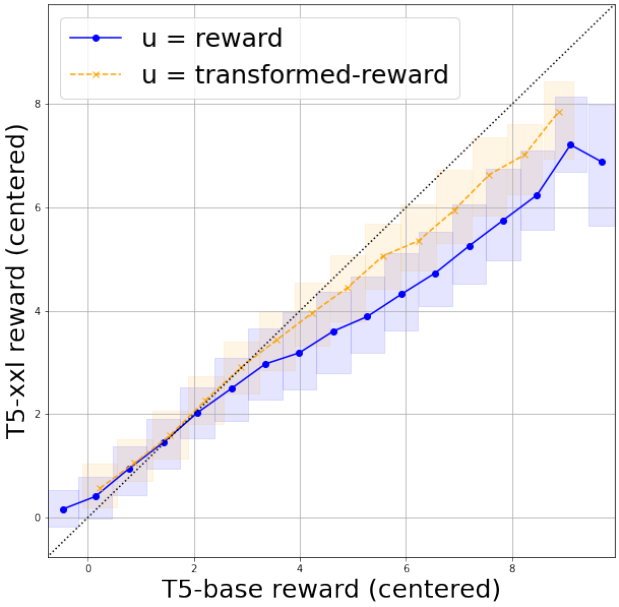
In Figure 4 we choose a pair of aligned policies matched on KL values to show that reward transformation leads to more uniform improvement than using the raw reward. In Figure 9 we visually show reward overoptimization (in the same aligned policies), and how reward transformation alleviates it. The mitigation is most obvious for responses with larger reward values. It's also important to note that reward aggregation exacerbates reward hacking with weighed sum of the raw reward: the policy model is incentivized to generate responses with very high harmlessness score (despite having smaller weight for harmlessness); to retain good scores in helpfulness, the policy model hacks the helpfulness reward model even more (compare Figure 9a against Figure 9c). Using sum of transformed reward alleviates this by two mechanisms.

In Figure 10 we show reward transformation leads to more concentrated reward distribution than using raw reward, the same trend across different KL values.

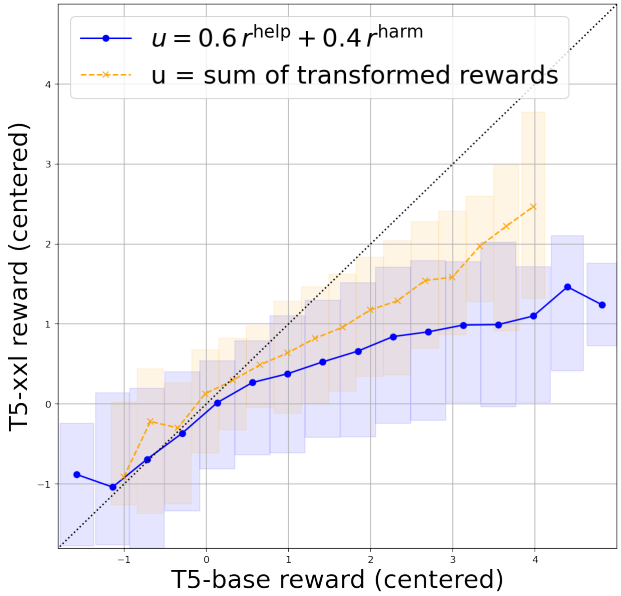
In Figure 3 we report win-rate against random sample under PALM-2, and against SFT 85%-quantile for helpfulness and 95%-quantile for harmlessness. In Figure 11, we report the extra evaluations.



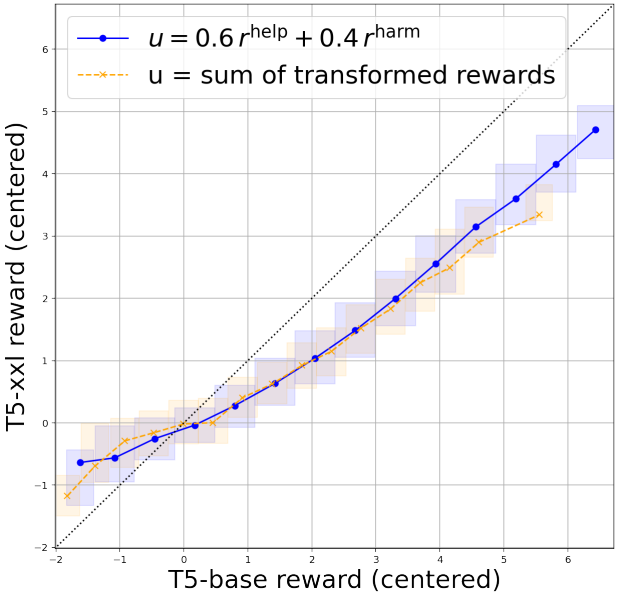
(a) Helpful (single reward): KL $\approx$ 18



(b) Harmless (single reward): KL $\approx$ 7

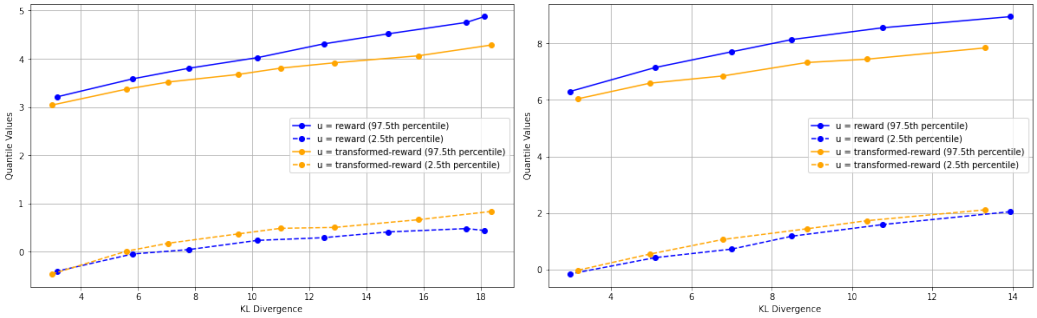


(c) Helpful (aggregated reward): KL $\approx$ 14



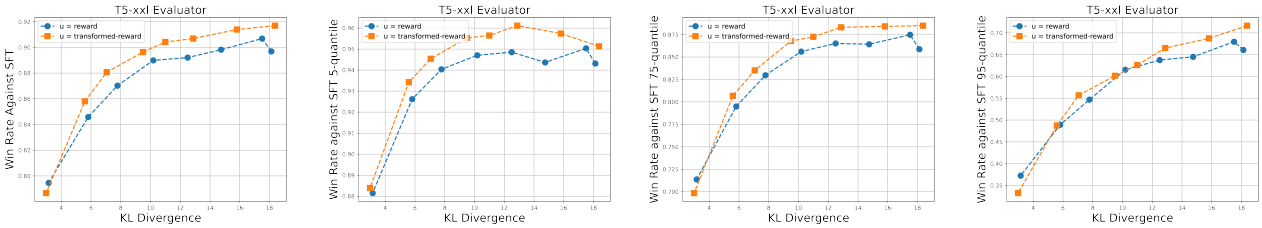
(d) Harmless (aggregated reward): KL $\approx$ 14

Figure 9. Reward transformation mitigates reward overoptimization, particularly for responses with larger reward values, and in reward aggregation where reward hacking is more severe. We compare reward overoptimization patterns in policies aligned with raw reward and transformed reward, in single reward (Figure 9a, Figure 9b) and reward aggregation (Figure 9c, Figure 9d) settings. The choice of aligned policies and score centering are the same as in Figure 4. For each plot, we sort the centered scores from T5-base reward model into 20 equal-width bins. For each bin with more than 10 data points: in the x-axis we show the interval medium; in the y-axis, we visualize the 25%, 50% and 75% quantile of the corresponding centered scores from T5-xxl.



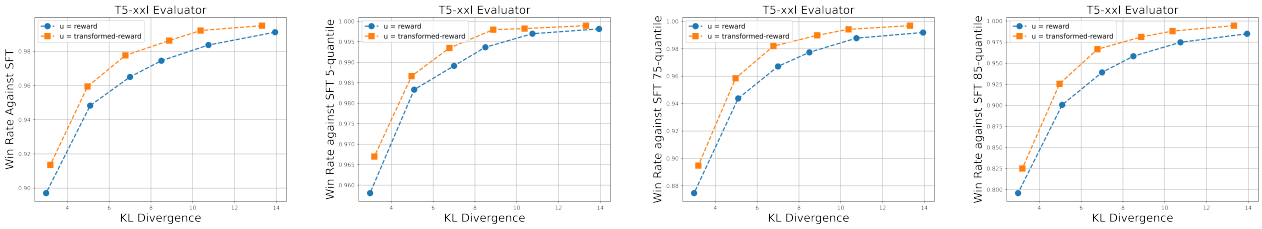
(a) Helpfulness (b) Harmlessness

Figure 10. Reward Transformation leads to more uniform reward improvements than baseline. The reward values are centered by median SFT rewards. This complements the results in Figure 4 for comparisons across KL values.



(a) SFT Random Sample (b) SFT 50%-quantile (c) SFT 75%-quantile (d) SFT 95%-quantile

(e) Helpfulness



(f) SFT Random Sample (g) SFT 50%-quantile (h) SFT 75%-quantile (i) SFT 85%-quantile

(j) Harmlessness

Figure 11. Transformed reward obtains better KL and win-rate trade-offs. These are win-rates compared against SFT random sample, and q-th quantile of the rewards of SFT samples. See details in Figure 3.