CORRELATED PROXIES: A NEW DEFINITION AND IMPROVED MITIGATION FOR REWARD HACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Because it is difficult to precisely specify complex objectives, reinforcement learning policies are often optimized using flawed proxy rewards that seem to capture the true objective. However, optimizing proxy rewards frequently leads to reward hacking: the optimized reward function ceases to be a good proxy and the resulting policy performs poorly with respect to the unspecified true reward. Principled solutions to reward hacking have been impeded by the lack of a good definition for the problem. We introduce a definition of reward hacking based on correlation between proxy and true rewards for states and actions seen by a "base policy" that breaks down under optimization. We show that this definition captures reward hacking behavior across several realistic settings, including in reinforcement learning from human feedback (RLHF). We then show theoretically that regularization to the base policy can effectively prevent reward hacking. Our theory suggests regularizing χ^2 divergence between the policies' occupancy measures, rather than the current practice in RLHF of using a KL penalty between action distributions. We intuitively show why this type of regularization is superior, and demonstrate that it better mitigates reward hacking in practice across four realistic settings, including RLHF.

1 INTRODUCTION

A major challenge for the designers of goal-oriented AI systems is specifying a reward function that robustly captures their goals and values. Manually designing reward functions is difficult due to the ambiguities and complexity underlying real-world scenarios (Ibarz et al., 2018). An alternative is to learn reward functions from human data (Sadigh et al., 2017; Jeon et al., 2020), but these often fail to generalize outside the distribution of behavior seen during training (McKinney et al., 2023; Tien et al., 2023). Thus, a learned or hand-specified reward function is often just a *proxy* for the true reward underlying the system designer's intent. Misalignment between the two objectives can lead to *reward hacking*: a learned policy performs well according to the proxy reward function, but not according to the true reward function (Russell et al., 2010; Amodei et al., 2016; Pan et al., 2022; Skalse et al., 2022). A reward hacking policy's behavior is often undesirable and can be especially catastrophic when deployed in safety-critical scenarios, such as autonomous driving (Krakovna et al., 2019; Turner et al., 2019; Knox et al., 2022). Unfortunately, reward hacking is a common phenomenon (Krakovna, 2018) and has harmful effects in the real world (Lum & Isaac, 2016; Corbett-Davies et al., 2017; Obermeyer et al., 2019; Milli et al., 2021; Franchi et al., 2023; Kleinberg et al., 2023).

The ideal solution to prevent reward hacking would be to perfectly align the specified proxy and unknown true reward; however, in many domains, this is impossible to achieve. For example, imagine trying to design a reward function for a self-driving car. It would have to capture the speed at which the car reached the destination, comfort of the passenger, all applicable laws, and other factors, many of which are difficult to robustly measure; furthermore, these factors would have to be carefully weighted against each other (Knox et al., 2022). In practice, reward hacking can occur even with significant reward engineering efforts. For example, hand-designed reward functions for recommender systems have led to adverse outcomes in terms of user experience (Stray et al., 2022).

Since proxy reward functions for complex tasks are nearly always misspecified in practice, what can be done to avoid reward hacking? There is a lack of principled solutions for preventing reward hacking, stemming from a more fundamental problem: *defining* reward hacking in a formal sense

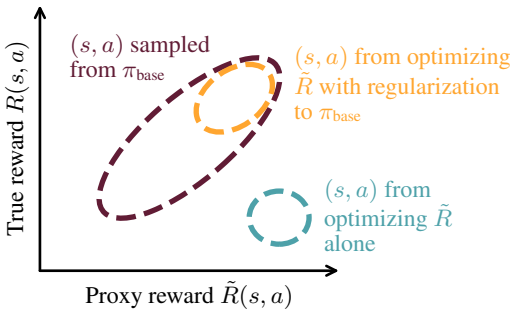


Figure 1: We present a new characterization of reward hacking and a method for preventing it. We define a proxy reward function as one that correlates with an unknown true reward function for state-action pairs sampled from some base policy. However, optimizing the proxy alone can lead to a breakdown in the correlation and worse true reward than the base policy. We show theoretically and empirically that optimizing the proxy with χ^2 occupancy measure regularization to the base policy can allow outperforming the base policy under the unknown true reward.

that captures realistic cases. In particular, it is difficult to characterize what makes a proxy reward function "good". For example, Skalse et al. (2022) define a notion of "unhackability" but find that it can only hold if the proxy is a multiple of the true reward, which is obviously unrealistic.

We argue that proxies are chosen because they seem to capture the true objective under some base distribution of behavior. To formalize this, we define a proxy reward as one that *correlates* with the true reward function under the distribution of states and actions visited by a *base policy*. Then, we define a hackable proxy as one that induces a decrease in true reward compared to the base policy when optimized (Figure 1). We show that these definitions capture a number of intuitive cases of reward hacking in realistic environments, including traffic control, blood glucose regulation, and reinforcement learning from human feedback (RLHF) (Figure 2).

Furthermore, our definition leads to a method for avoiding reward hacking during policy optimization by regularizing the optimized policy to be similar to the base policy. Specifically, we find that optimizing the proxy reward minus a regularization term provides a *provable* lower bound on improvement in true reward. The amount of regularization needed depends on the correlation between the proxy and true rewards; as the correlation becomes stronger, it is possible to use less regularization.

Regularizing policy optimization to a base policy is already used in practice during RLHF via a KL divergence penalty (Stiennon et al., 2020; Bai et al., 2022), and Theorem 5.1 provides some theoretical justification for why this works. However, while the regularization in RLHF penalizes the KL divergence between the optimized and base policies' *action distributions*, our result suggests that it is better to penalize the chi-squared (χ^2) divergence between the policies' *occupancy measures*. A policy's occupancy measure (OM) is the distribution of states or state-action pairs seen by the policy when it interacts with its environment. Unlike action distribution (AD)-based metrics, occupancy measures take into account the states that the agent reaches, not just the actions it takes. We compare OM vs. AD regularization and χ^2 vs. KL divergence, presenting intuitive reasons why χ^2 OM divergence may be a better regularization target in Figures 3 and 4.

Based on our theoretical results, we empirically investigate using the benefits of using χ^2 occupancy measure regularization to prevent reward hacking. We implement OM-based regularization in practice using a discriminator network that approximates the OM divergence between policies. We then optimize policies with misaligned proxy reward functions in multiple reward hacking benchmark environments (Pan et al., 2022) using OM and AD regularization with χ^2 and KL divergence. The results of our experiments demonstrate that training with occupancy measure regularization leads to better performance under the unseen true reward function in all of the environments, validating our theoretical results. In contrast, we find that it is difficult to tune AD regularization in some environments to both prevent reward hacking and allow meaningful improvement over the base policy. Furthermore, regularization with χ^2 divergence leads to more stable results across regularization coefficients and often achieves higher true reward.

Our main contributions can be summarized as follows:

1. We provide a new formal definition of reward hacking and show that it captures a number of realistic case studies.
2. Using our definition, we establish that optimizing a proxy reward with χ^2 occupancy measure regularization leads to a provable improvement in the unknown true reward function.
3. We implement χ^2 OM regularization in practice and demonstrate that it is superior to the current standard for preventing reward hacking via regularization.

2 Related Work

While there have been separate lines of work investigating reward hacking and exploring the use of occupancy measure divergences for other applications, to the best of our knowledge, we are the first to specifically study applying occupancy measure regularization to the problem of reward hacking.

Reward hacking. Some prior works establish theoretical models of reward hacking as a special case of Goodhart's Law (Goodhart, 1984; Leike et al., 2018; Manheim & Garrabrant, 2019; Krakovna, 2019; Skalse et al., 2022; Ngo et al., 2023). Krakovna (2018) provide a list of many examples of reward hacking. Pan et al. (2022) categorize types of reward misspecification and relate optimization power to reward hacking.

Safe reinforcement learning. In the context of reward hacking, regularizing policies to be similar to an offline policy based on their action distribution KL divergence was proposed by Stiennon et al. (2020) and has since been widely employed in the context of optimizing LLMs using RLHF (Ouyang et al., 2022; Bai et al., 2022; Glaese et al., 2022). KL regularization for RLHF has been further studied by Vieillard et al. (2021), Gao et al. (2022), and Korbak et al. (2022). Nika et al. (2024) propose a type of occupancy measure regularization in RLHF but it is limited to deterministic MDPs, while we study general stochastic MDPs. Some alternative approaches to avoid reward hacking include quantilizers (Taylor, 2016), "mild" optimization (Taylor et al., 2020), and impact regularization (Turner et al., 2020). While constrained RL can prevent the misbehavior of agents that optimize flawed reward functions (Dalal et al., 2018; Chow et al., 2019; Zhang et al., 2020; Roy et al., 2022), it simply shifts the difficulty of designing a reward function to specifying a set of constraints and weights. Robust RL usually considers a misspecified transition model, but some work has explored misspecified reward functions (Derman et al., 2021; Gadot et al., 2024). Other proposals to address the reward specification problem attempt to infer the true reward function based on the given proxy reward function, environment context, and/or feedback from humans (Hadfield-Menell et al., 2017; Reddy et al., 2020; Lee et al., 2021). Gleave et al. (2021) have previously studied quantifying the similarity of reward functions.

Applications of occupancy measure regularization. Occupancy measure regularization and optimization have been used for a variety of purposes. GAIL (Ho & Ermon, 2016) is an algorithm for robust imitation learning that aims to match the imitator's occupancy measure to that of the demonstrator. Kang et al. (2018) combines GAIL with a reward function to efficiently explore using human data. Another line of work aims to find a policy with the highest-entropy occupancy measure for the purpose of exploring the state space (Hazan et al., 2019; Lee et al., 2020; Nedergaard & Cook, 2023). Various types of distributional regularization are used in model-based RL since learned models may not generalize out-of-distribution (Yang et al., 2022).

Offline reinforcement learning. Prior work in offline RL may appear to be particularly related to our work. Many offline RL algorithms use occupancy measure or action distribution-based regularization to ensure that the learned policy remains within the training data distribution (Fujimoto et al., 2019; Lee et al., 2022; Mandal et al., 2023; He, 2023; Cheng et al., 2022; Rashidinejad et al., 2023; Xie et al., 2023). However, the settings are fundamentally different: while offline RL is limited by a lack of coverage in the training data, the difficulty in our setting is that the reward function is misspecified. While it might be possible to avoid reward hacking by using offline RL algorithms, we leave this to future work and focus on regularization-based approaches in the online RL setting.

3 Preliminaries

To study reward hacking, we consider the setting of an infinite-horizon Markov decision process (MDP). An agent takes actions $a \in \mathcal{A}$ to transition between states $s \in \mathcal{S}$ over a series of timesteps $t = 0, 1, 2, \ldots$. We assume that $\mathcal{S}$ and $\mathcal{A}$ are finite for simplicity but our results can easily generalize to infinite state or action spaces. The first state s_0 is sampled from an initial distribution $\mu_0(s)$, and when an agent takes action a_t in s_t at time t, the next state s_{t+1} is reached at timestep $t+1$ with transition probability $p(s_{t+1} \mid s_t, a_t)$. The agent aims to optimize a reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and rewards are accumulated over time with discount factor $\gamma \in [0, 1)$. A policy π maps each state s to a distribution over actions to take at that state $\pi(a \mid s)$. We define the (normalized) *return* of a policy π under a reward function R as

$$J(\pi, R) = (1-\gamma)\, \mathbb{E}_\pi \left[\textstyle\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right] \tag{1}$$

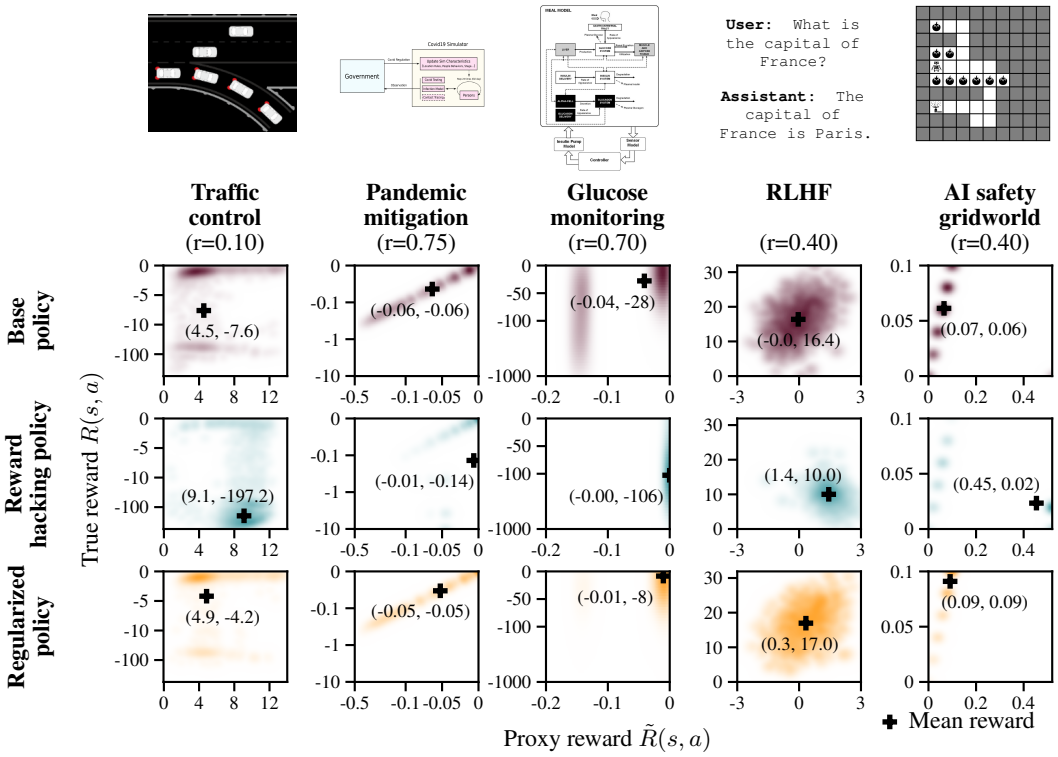


Figure 2: We show that reward hacking in four realistic environments and an illustrative gridworld is captured by our definition. The top row shows the distribution of proxy and true reward values for state-action pairs sampled from a natural base policy for each environment; in all environments, the proxy and true rewards are correlated. However, if the proxy is optimized via reinforcement learning, the correlation breaks down and the true reward is lower than the base policy, which we define as reward hacking (middle row). Our theoretical results show that RL with occupancy measure regularization to the base policy can prevent reward hacking and enable an increase in true reward (bottom row).

where $\mathbb{E}_\pi$ refers to the expectation under the distribution of states and actions induced by running the policy π in the environment.

We define the *state-action occupancy measure* μ_π of a policy π as the expected discounted number of times the agent will be in a particular state and take a specific action:

$$\mu_\pi(s,a) = (1-\gamma)\,\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t \mathbb{1}\{s_t = s \wedge a_t = a\}\right]. \tag{2}$$

Intuitively, the occupancy measure represents the distribution of states and actions visited by the policy over time. Furthermore, combining (1) and (2), it is easy to show that the return of a policy can be re-written as the expected reward for states and actions sampled from the occupancy measure:

$$J(\pi, R) = \sum_{s,a\in\mathcal{S}\times\mathcal{A}} \mu_\pi(s,a)R(s,a) = \mathbb{E}_{\mu_\pi}[R(s,a)]. \tag{3}$$

The standard approach to solving an MDP is to find a policy π that maximizes its return $J(\pi, R)$. However, as we discussed in the introduction, an AI system designer might optimize π using a learned or hand-specified *proxy* reward function $\tilde{R}$ which is misaligned with the *true* reward function R. In order to better understand and mitigate reward hacking, it would be helpful to have a good definition of the problem. Intuitively, reward hacking is when optimizing the proxy reward $J(\pi, \tilde{R})$ of a policy π ultimately leads to low true reward $J(\pi, R)$. However, this intuition is more difficult to formalize than it might seem.

Desiderata for a definition of reward hacking. To understand why defining reward hacking is difficult, consider the case study of RLHF. In this case, optimizing a learned reward function over LLM outputs eventually leads to the LLM producing nonsensical responses. This clearly satisfies our informal definition of "optimizing the proxy makes the true reward go down." However, what if we were to optimize $\tilde{R}(s,a) = -R(s,a)$? In this case, optimizing $\tilde{R}$ also makes the true reward

go down, but arguably this is not reward hacking because $\tilde{R}$ was not a good "proxy" in the first place. Thus, a good definition of reward hacking should distinguish between "reasonable" proxies and reward functions that are clearly unrelated (or opposite) to the true reward function.

In the RLHF example, optimizing until the LLM produces nonsensical outputs seems like bad enough behavior that we would say the proxy has been "hacked." However, what if optimizing the proxy produced mediocre but not obviously bad outputs? Even if optimizing the proxy does not lead to near-optimal true reward, we would not always say that reward hacking has occurred. Thus, a good definition of reward hacking should also choose a threshold for true reward: below the threshold, reward hacking is occurring, and above the threshold it is not.

Prior definitions of reward hacking. Because of these difficulties, prior work has struggled to define reward hacking. With regard to our first desideratum, defining reasonable proxies, Skalse et al. (2022) introduce a notion of an "unhackable" proxy reward but find that a proxy is only unhackable if it is equivalent to the true reward up to scaling; this is clearly too restrictive of a definition. The robust MDP literature has considered proxy rewards that differ from the true reward by at most some constant, e.g., $|R(s,a) - \tilde{R}(s,a)| \leq c$ for all s, a (Derman et al., 2021). However, we find that in many realistic cases of reward hacking, there may be some states where a proxy differs from the true reward by an arbitrarily large amount, and so these definitions are generally not applicable. With regard to the second desideratum, defining a threshold of true reward, there is little existing literature.

4 Defining Reward Hacking

Despite the difficulties in formalizing reward hacking, we argue that both of our desiderata for a definition can be met by defining reward hacking with respect to a *base policy*. We show how this allows for both a precise definition of proxy rewards and a natural threshold for when reward hacking is occurring.

Characterizing proxy rewards. To find a realistic definition of a good proxy reward, consider the process by which system designers create proxies. If they are hand-specified, we argue that designers usually imagine a distribution of behavior and design a reward that captures the objective under that distribution. For example, we study a traffic control simulator (Lopez et al., 2018; Vinitsky et al., 2018; Wu et al., 2022; Pan et al., 2022) where a small number of autonomous cars help to regulate traffic among a larger population of human drivers. In this case, a designer might choose the average speed of all vehicles as a proxy for improving traffic flow; this is a reasonable proxy because under the distribution of typical human driving behavior, higher speeds are associated with better traffic.

We can formalize this intuition by requiring that the proxy and true rewards be *correlated* over states and actions sampled from a *base policy*:

Definition 4.1 (Correlated proxy reward)**.** An r-correlated proxy reward $\tilde{R}$ with respect to a base policy π_{base} is one that has a correlation of $r > 0$ with the true reward for state-action pairs sampled from the base policy:

$$\mathbb{E}_{\mu_{\pi_{\text{base}}}}\left[\left(\frac{\tilde{R}(s,a)-J(\pi_{\text{base}},\tilde{R})}{\sigma_{\tilde{R}}}\right)\left(\frac{R(s,a)-J(\pi_{\text{base}},R)}{\sigma_R}\right)\right] = r,$$

where $\sigma^2_{\tilde{R}} = \mathbb{E}_{\mu_{\pi_{\text{base}}}}\left[\left(\tilde{R}(s,a) - J(\pi_{\text{base}}, \tilde{R})\right)^2\right]$ and $\sigma^2_{R} = \mathbb{E}_{\mu_{\pi_{\text{base}}}}\left[\left(R(s,a) - J(\pi_{\text{base}}, R)\right)^2\right]$ are the variances of proxy and true rewards under the base policy.

This definition intuitively captures cases like the traffic environment, where we define the true reward function as the negative total commute time for all vehicles. Letting π_{base} be an autonomous vehicle policy based on a common model of human driving behavior, we plot the distribution of true and proxy rewards in the top-left corner of Figure 2 and find that they are correlated: higher average speed tends to occur with lower commute times, and vice versa. This validates that average speed is a correlated proxy reward according to Definition 4.1.

Definition 4.1 also avoids too strongly constraining proxy rewards, unlike past formalisms. The proxy and true reward functions can diverge arbitrarily at some state-action pairs, as long as those state-action pairs have low or zero occupancy measure under the base policy.

Finally, Definition 4.1 captures *learned* proxy rewards in addition to hand-specified ones. Reward

learning begins with collecting annotations—for example, preference comparisons (as in RLHF) or scalar feedback—for states and actions collected from a rollout policy—for example, the SFT model in RLHF. Then, a reward function is estimated via supervised learning over this dataset with an appropriate loss function. If the estimated reward function generalizes well in-distribution, then we should expect the true and learned rewards to be well correlated under states and actions sampled from the rollout policy, satisfying Definition 4.1 (see Lemma A.7).

Choosing a threshold for reward hacking. As discussed in Section 3, the other difficulty in defining reward hacking is specifying a threshold of true reward below which performance is poor enough to be considered hacking. If we are already characterizing proxy rewards with respect to a base policy, then it makes sense to also use this base policy as a baseline for evaluating a policy that optimizes the proxy. If optimizing the proxy leads to worse true reward than the base policy achieves, then the system designer may as well simply use the base policy.

Definition 4.2 (Hackable proxy reward)**.** Suppose a $\tilde{R}$ is an r-correlated proxy with respect to π_{base} (Definition 4.1). Then we say *reward hacking* occurs when a policy π optimized for $\tilde{R}$ has lower true reward than the base policy π_{base}, i.e., when $J(\pi, R) < J(\pi_{\text{base}}, R)$. If an optimal policy for $\tilde{R}$ exhibits reward hacking, then we say that the proxy reward $\tilde{R}$ is *hackable*:

$$J(\pi, R) < J(\pi_{\text{base}}, R) \qquad \text{for some} \qquad \pi \in \arg\max_{\pi} J(\pi, \tilde{R}).$$

For example, in the traffic control environment, optimizing for the proxy of average speed leads to the autonomous vehicles blocking a highway on-ramp. This decreases the speed of the cars on the ramp to zero, but allows cars on the highway to move at high speeds, overall increasing the proxy reward. However, the true reward becomes extremely low, since commute times for the cars on the on-ramp are aribitrarily long. This constitutes reward hacking under Definition 4.2 since the true reward of the proxy-optimized policy is lower than that achieved by typical human driving.

Verifying our definition experimentally. To test whether Definition 4.2 accurately captures intuitive cases of reward hacking, we consider four realistic enviroments and an illustrative gridworld. We consider the aforementioned traffic simulator and two other environments originally studied by Pan et al. (2022) as examples of reward hacking. SimGlucose (Man et al., 2014) is based on an FDA-approved simulator of Type 1 Diabetes patients in which a policy monitors glucose levels and administers insulin; the true reward captures patient health while the proxy reward prioritizes reducing the monetary cost of insulin. PandemicSimulator (Kompella et al., 2020) uses a specialized SEIR model (Mwalili et al., 2020) to simulate the COVID-19 pandemic among a population; the policy controls the level of lockdown restrictions placed on the population by observing the results of testing. The proxy reward omits the political cost of certain decisions.

We also study RLHF, in which LLMs are optimized based on a reward function learned from human preferences. Following Gao et al. (2022) and Coste et al. (2024), we use a large (more robust) reward model as the true reward function and a smaller (less robust) one as the proxy. Finally, as an interpretable and illustrative example, we include the tomato-watering gridworld from Leike et al. (2017). In this environment, a robot that waters tomatoes receives true reward depending on how many plants are watered; however, the proxy additionally rewards the robot for standing in a sprinkler where it appears all the tomatoes are watered. Besides the gridworld, all of our environments reflect complex, realistic tasks with large or infinite state state spaces. To test our definition of reward hacking, for each of the five environments we construct a natural base policy. See Appendix C for more details about the environments and base policies.

The top row of Figure 2 shows the distribution of true and proxy reward values for state-action pairs sampled from these base policies in each environment. We find that in all cases, the proxy rewards correlate with the true reward, satisfying our definition of a correlated proxy. The middle row of Figure 2 shows the result of optimizing for the proxies: for each environment, we plot the distribution of true and proxy reward values for state-action pairs reached by a policy optimized on the proxy reward via RL. In all cases, we see that the true reward drops significantly compared to the base policy. Thus, these intuitive cases of reward hacking are captured by Definition 4.2.

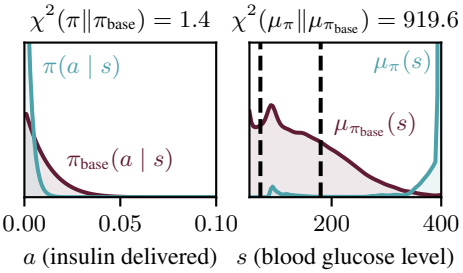


Figure 3: Unlike RLHF, which regularizes *action distribution* (AD) divergence to prevent reward hacking, our results suggest regularizing using *occupancy measure* (OM) divergence. These plots of the glucose monitoring environment show the typical ADs and OMs of two policies. π is close to π_{base} in AD; it is more likely to give slightly less insulin because the proxy reward penalizes monetary cost. However, this leads to a vastly different OM, with typical glucose levels for π far outside the healthy range (dotted lines). Thus, regularizing ADs to be close to π_{base} is not enough to prevent reward hacking; instead, divergence between the OMs better captures the reward hacking behavior.

5 Mitigating Reward Hacking with Occupancy Measure Regularization

We now discuss how our new definition of reward hacking can lead to better methods for preventing it. Ideally, we would like to be able to optimize a proxy reward function and have it translate into an improvement in true reward over the base policy. To understand how this might be possible, consider again the traffic environment. The reward hacking policy exhibits behavior which is very unlikely under the base policy: human drivers hardly ever stop on on-ramps and refuse to move. That is, optimizing the proxy reward leads to *out-of-distribution states and actions* where the correlation that made the proxy good in the first place breaks down. Visually, this can be seen in the second row of Figure 2: the reward hacking policy usually finds state-action pairs with high proxy reward and low true reward that were very unlikely to be reached by the base policy.

Thus, to prevent reward hacking, one solution could be to optimize the proxy reward while avoiding states that are unlikely under the base policy. The following theorem formalizes this idea.

Theorem 5.1. *Suppose that $\tilde{R}$ is an r-correlated-proxy for the true reward function R, and let $\sigma_{\tilde{R}}$ and σ_R be defined as in Definition 4.1. Then for any policy π such that $\mu_\pi \ll \mu_{\pi_{base}}$ (i.e., $\mu_{\pi_{base}}(s,a) = 0 \Rightarrow \mu_\pi(s,a) = 0$), we have*

$$\frac{J(\pi, R) - J(\pi_{base}, R)}{\sigma_R} \geq \frac{1}{r}\left(\frac{J(\pi, \tilde{R}) - J(\pi_{base}, \tilde{R})}{\sigma_{\tilde{R}}} - \sqrt{(1-r^2)\chi^2\left(\mu_\pi \| \mu_{\pi_{base}}\right)}\right), \tag{4}$$

where $\chi^2\left(\mu_\pi \| \mu_{\pi_{base}}\right) = \mathbb{E}_{\mu_\pi}\left[\frac{\mu_\pi(s,a)}{\mu_{\pi_{base}}(s,a)} - 1\right]$ is the χ^2 divergence between μ_π and $\mu_{\pi_{base}}$.

See Appendix A for the proof. Equation (10) gives a lower bound on how much the policy π improves over the base policy π_{base} on the true reward, normalized by the standard deviation of the true reward under the base policy. This is exactly we would like to maximize, but we can't optimize the unknown true reward directly, so it makes sense to instead optimize the right hand side of (10). The right-hand side (RHS) consists of two terms. The first is the normalized improvement of the policy π's proxy reward over the base policy; optimizing this term alone often leads to reward hacking. However, the second term penalizes the divergence of π's occupancy measure from that of π_{base}. By incentivizing π to achieve high proxy reward but also stay close to the base policy, π can achieve high true reward.

Scaling the RHS of (10) and removing terms that are constant in π suggests using the following regularized policy optimization objective to avoid reward hacking:

$$\text{maximize} \quad J(\pi, \tilde{R}) - \lambda\sqrt{\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right)} \qquad \text{where} \qquad \lambda = \sigma_{\tilde{R}}\sqrt{1-r^2}. \tag{5}$$

The amount of regularization needed to improve on the base policy depends on the strength of correlation r. The higher the correlation, the lower the regularization strength $\sqrt{1-r^2}$. Given the prefactor of $\frac{1}{r}$ on the RHS of (10), it may appear that a lower correlation leads to a larger gain in true reward. However, Lemma A.2 shows that in fact, the lower bound decreases as a function of r.

While Theorem 5.1 does not *guarantee* that the true reward can be increased by optimizing (5), it does at least allow us to *provably avoid* reward hacking. In Theorem A.3 in the appendix, we show that it is difficult to guarantee an improvement in true reward in general. However, if the lower bound on the RHS of (10) can be increased above zero—which can be tested empirically—then we know that the true reward has also increased. In our experiments in Section 6, we show that in many realistic

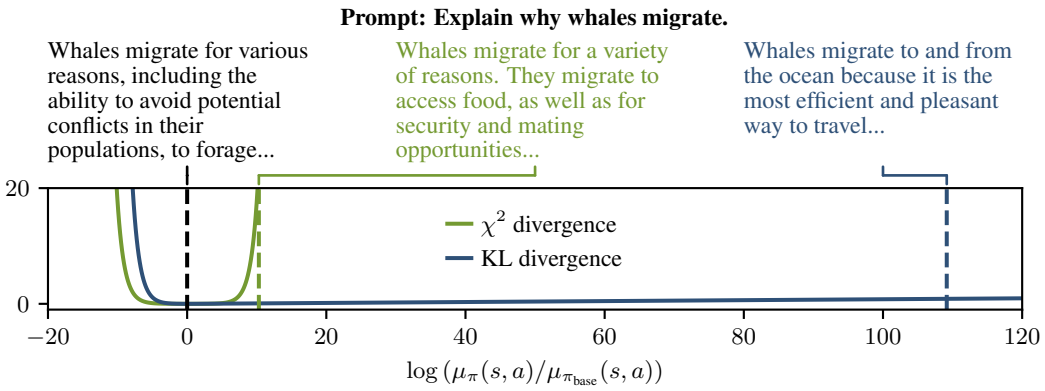


Figure 4: Our theory also suggests preventing reward hacking by regularizing with χ^2 *divergence* instead of *KL divergence* like prior work. This plot illustrates how χ^2 regularization is more effective at preventing reward hacking in RLHF. Both divergences can be written as the expectation of a function $g\big(\log\big(\mu_\pi(s,a)/\mu_{\pi_{\text{base}}}(s,a)\big)\big)$ which penalizes state-action pairs more the further the log-ratio is from zero. The $g(\cdot)$ associated with KL divergence only increases slowly for large log-ratios, so policies trained with KL divergence may produce nonsensical text. In contrast, the $g(\cdot)$ for χ^2 divergence increases exponentially, better constraining the LLM to produce text similar to the SFT policy.

environments it is possible to increase the true reward by optimizing (5).

Comparison to KL regularization in RLHF. Regularization to a base policy is already widely used to prevent reward hacking in RLHF (Stiennon et al., 2020; Bai et al., 2022). Specifically, a KL penalty is applied between the distributions of responses generated by the optimized policy and the SFT policy. In our setting, we can write this as

$$\text{maximize} \quad J(\pi, \tilde{R}) - \lambda(1-\gamma)\mathbb{E}_\pi\left[\textstyle\sum_{t=0}^{\infty} \gamma^t D_{\text{KL}}\big(\pi(\cdot \mid s_t) \parallel \mu_{\pi_{\text{base}}}(\cdot \mid s_t)\big)\right]. \tag{6}$$

That is, in RLHF, the expected KL divergence between the action distributions of π and π_{base} is penalized. Action distribution divergence is easy to calculate and optimize, and training LLMs with (6) seems to work well in practice. However, unlike our objective in (5), (6) lacks theoretical guarantees, and it is unclear if it works in other environments.

Our regularized objective in (5) differs in two main ways from the KL regularization used in RLHF. First, our results suggest optimizing the *occupancy measure* (OM) divergence between policies, whereas RLHF uses the *action distribution* (AD) divergence. Second, Theorem 5.1 applies to χ^2 divergence, while RLHF uses KL divergence. In the remainder of this section, we explore intuitively why OM divergence is preferable to AD divergence, and why χ^2 divergence is superior to KL divergence. Then, in Section 6, we empirically explore applying different types of regularization to prevent reward hacking in five environments.

Occupancy measure vs. action distribution regularization. While AD regularization works well for RLHF, this may be because RLHF is essentially a contextual bandit problem, meaning that $\gamma = 0$; in this case, OM and AD divergence are equivalent (see Appendix A.3). However, in other cases, AD divergence may not suffice to prevent reward hacking behavior. This is because, in longer-horizon environments, a small change in action distribution at a single state can lead to a much higher probability of reaching undesirable states. Figure 3 shows an example of this in the glucose monitoring environment: policies that are close in action distribution regularization produce vastly different patient glucose levels. Occupancy measure regularization avoids this issue by directly preventing the distribution of glucose levels from differing too much from the base policy. In Theorem A.4 in the appendix, we show that in general it is *impossible* to lower bound the improvement in true reward using almost any form of AD regularization, in contrast to our results on OM regularization.

χ^2 vs. KL divergence. Compared to KL divergence, χ^2 divergence may be superior for preventing reward hacking because it more strongly penalizes out-of-distribution state-action pairs. To illustrate this, we can write both divergences as expectations over functions of the log-ratio of the occupancy measures, which we denote $d(s,a)$:

$$D_{\text{KL}}(\mu_\pi \parallel \mu_{\pi_{\text{base}}}) = \mathbb{E}_{\mu_\pi}\left[d(s,a) + e^{-d(s,a)}\right] \qquad \chi^2(\mu_\pi \parallel \mu_{\pi_{\text{base}}}) = \mathbb{E}_{\mu_\pi}\left[e^{d(s,a)} + e^{-d(s,a)}\right]$$
$$\text{where} \qquad d(s,a) = \log\big(\mu_\pi(s,a)/\mu_{\pi_{\text{base}}}(s,a)\big). \tag{7}$$

Method	Environment: Traffic control ($\times 10^3$)	Pandemic mitigation	Glucose monitoring ($\times 10^3$)	RLHF	AI safety gridworld
Action dist. χ^2	-1.29 $\pm$ 0.10	-12.29 $\pm$ 0.05	-74.8 $\pm$ 11.8	**16.94** $\pm$ 0.07	6.24 $\pm$ 0.09
State occupancy χ^2	-2.18 $\pm$ 0.38	-10.68 $\pm$ 0.15	-54.7 $\pm$ 1.0	—	9.07 $\pm$ 0.06
State-action occupancy χ^2	**-1.15** $\pm$ 0.05	-11.17 $\pm$ 0.17	**-47.6** $\pm$ 0.6	—	**9.17** $\pm$ 0.11
Action dist. KL	-1.33 $\pm$ 0.05	-12.20 $\pm$ 0.06	-73.4 $\pm$ 8.3	16.81 $\pm$ 0.27	6.33 $\pm$ 0.11
State occupancy KL	-1.34 $\pm$ 22.6	**-10.24** $\pm$ 0.54	-58.4 $\pm$ 3.4	—	7.07 $\pm$ 0.11
State-action occupancy KL	-1.25 $\pm$ 0.06	-11.73 $\pm$ 0.19	-48.9 $\pm$ 0.5	—	6.86 $\pm$ 0.17
π_{base}	-2.28 $\pm$ 0.00	-12.26 $\pm$ 0.00	-72.6 $\pm$ 0.0	16.37 $\pm$ 0.00	5.86 $\pm$ 0.00
No regularization	-57.38 $\pm$ 3.53	-29.57 $\pm$ 6.86	-599.0 $\pm$ 1.6	9.16 $\pm$ 0.80	2.35 $\pm$ 0.14
Training with true reward	-0.93 $\pm$ 0.11	-2.65 $\pm$ 0.83	-43.4 $\pm$ 0.8	—	8.54 $\pm$ 0.12

Table 1: We compare using various types of regularization to prevent reward hacking in the five environments form Figure 2. The median true reward and standard deviation across 5 random seeds is shown for the best regularization coefficient for each type of regularization. The bottom rows show results for the baselines: the base policy π_{base}, a policy trained on the proxy reward without regularization (exhibiting reward hacking), and a policy trained on the true reward function (impossible in practice, but included as an upper bound on performance). We find that occupancy measure regularization consistently improves on action distribution regularization, and that χ^2 divergence is often superior to KL divergence.

As $d(s, a)$ increases, the optimized policy is visiting state-action pairs that are less likely under the base policy. However, KL divergence only penalizes increases in $d(s, a)$ linearly, while χ^2 penalizes them exponentially, resulting in stronger regularization even with a lower coefficient. Figure 4 plots the functions in (7) and shows how in practice χ^2 divergence better prevents reward hacking in RLHF.

6 Experiments

We now show that our theoretical results—which suggest χ^2 occupancy measure regularization can prevent reward hacking—lead to empirical success in realistic environments.

Practical occupancy measure regularization. Occupancy measure regularization is more difficult to implement in practice compared to action distribution regularization. While AD regularization can be added as a loss term to deep RL algorithms like proximal policy optimization (PPO), OM divergences cannot be calculated in closed form. Instead, we follow several previous works (e.g., Ho & Ermon 2016; Kang et al. 2018) and use a discriminator network to approximate OM divergences. Specifically, in Appendix B, we show the objective in (5) can be optimized via policy gradient with an adjusted reward function that depends on a discriminator $\hat{d}_\phi$:

$$R'(s,a) = \tilde{R}(s,a) - \frac{\lambda}{\sqrt{\widehat{\chi^2}}} e^{\hat{d}_\phi(s,a)} \qquad \text{where} \qquad \widehat{\chi^2} = \mathbb{E}_{\mu_\pi}\left[e^{\hat{d}_\phi(s_t,a_t)} - 1 \right]$$

$$\text{and} \qquad \phi = \arg\min_{\phi'} \mathbb{E}_{\mu_\pi}\left[\log(1 + e^{-\hat{d}_{\phi'}(s,a)}) \right] + \mathbb{E}_{\mu_{\pi_{\text{base}}}}\left[\log(1 + e^{\hat{d}_{\phi'}(s,a)}) \right]. \tag{8}$$

That is, optimizing a discriminator network $\hat{d}_\phi$ to minimize the given loss can be used to estimate and optimize χ^2 OM divergence. We alternately train $\hat{d}_\phi$ via gradient descent on (8) and the policy π via PPO based on the adjusted reward. We call this algorithm Occupancy-Regularized Policy Optimization (ORPO). See Appendix B for a full derivation of the approximations used in ORPO and Algorithm 1 for a formal description. We use a similar strategy for regularizing based on OM KL divergence; see the appendix for details.

Experimental setup. In each of the five environments shown in Figure 2, we train policies with four types of regularization towards the base policy: AD KL, AD χ^2, OM KL, and OM χ^2. In Appendix A.5, we show that Theorem 5.1 also holds for state-only occupancy measures if the environment's reward function does not depend on the action; thus, we experiment with regularizing based on both state-action and state-only OM divergence. For each environment and type of regularization, we test a number of regularization coefficients λ. Theorem 5.1 suggests setting $\lambda = \sigma_{\tilde{R}}\sqrt{1-r^2}$ for an r-correlated proxy, so for χ^2 regularization we test a range of values $\lambda = c\,\sigma_{\tilde{R}}$ from $c = 1$ to 10^{-2} (10^{-4} for RLHF). Since it is less clear theoretically how to set the coefficient λ for KL regularization,

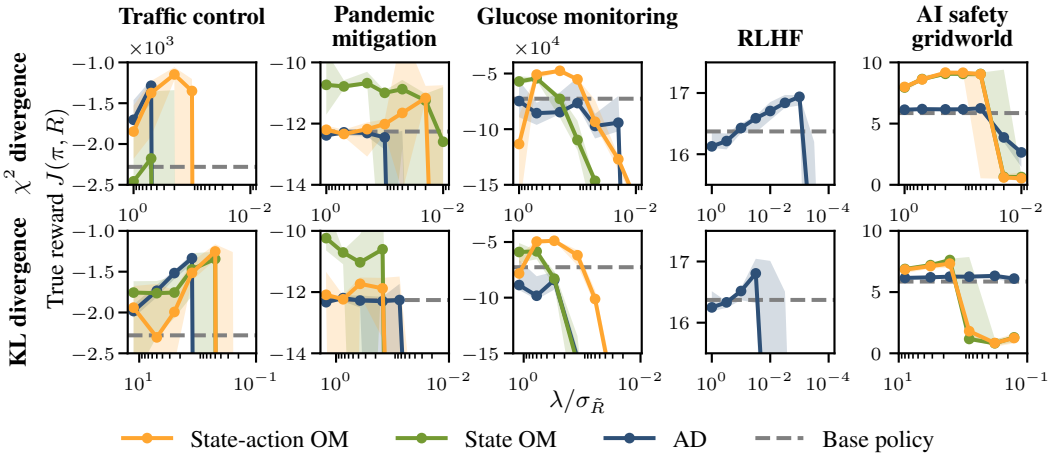


Figure 5: The true reward achieved by policies regularized with varying amounts of action distribution or occupancy measure regularization using χ^2 and KL divergence. The x-axis is the regularization coefficient λ normalized by the standard deviation of proxy rewards under the base policy. Dots indicate the median reward and the shaded area is the range over random seeds. For RLHF, AD and OM regularization are equivalent, which is why OM regularization results are not shown for that column.

we experiment with a wider range of values.

We train each combination of $\{\chi^2$ divergence, KL divergence$\} \times \{$AD, state OM, state-action OM$\} \times \{\lambda_1, \lambda_2, \dots\}$ with five random seeds and measure the resulting policies' expected returns under the true reward. As a baseline, we train a policy without any regularization, which leads to reward hacking in all environments. We also train a policy directly on the true reward function as an upper limit for performance. In RLHF, we only consider AD regularization since it is equivalent to OM regularization for LLM chatbots (see Appendix A.3). We do not train a policy on the true reward for RLHF as we found it could be hacked with enough optimization pressure. See Appendix D for all hyperparameter and experiment details.

Results. The results of our experiments are shown in Table 1 and Figure 5. Table 1 shows the median true reward with the best coefficient for each type of regularization. We find that OM regularization consistently outperforms AD regularization across the four non-RLHF environments. In two environments (glucose monitoring and pandemic mitigation), AD regularization fails to improve on the base policy's true reward at all. Furthermore, χ^2 regularization tends to perform similarly to or better than KL regularization across all environments. In RLHF in particular, χ^2 regularization leads to a larger improvement over the base policy compared to the industry-standard KL penalty, and is more stable across seeds as well.

In Figure 5, we show the true reward achieved when training with each type of regularization across a range of λ values. In addition to performing best with an optimal coefficient, χ^2 and OM regularization seem to also perform well over a larger range of coefficients compared to AD regularization. In Appendix E, we present the full results of our experiments and ablations of ORPO.

7 Conclusion

We have introduced a new definition for reward hacking based on correlation between a proxy reward function and the unknown true reward that breaks down when optimizing the proxy. Furthermore, we leveraged this definition to show theoretically and empirically that χ^2 occupancy measure regularization can effectively prevent reward hacking. Our results have implications for settings, like RLHF, where RL is used to optimize complex, hard-to-specify objectives. We suggest that the heuristic KL penalty used currently should be replaced by a more principled form of regularization. While OM and AD regularization are equivalent for today's formulation of RLHF as a contextual bandit, they will no longer remain so as LLM-based agents are optimized over multi-turn conversations or with tool use (Wang et al., 2023; Abdulhai et al., 2023; Shani et al., 2024). Thus, our results provide a principled path to continuing to ensure the safety of increasingly powerful AI systems.

REFERENCES

Marwa Abdulhai, Isadora White, Charlie Snell, Charles Sun, Joey Hong, Yuexiang Zhai, Kelvin Xu, and Sergey Levine. LMRL Gym: Benchmarks for Multi-Turn Reinforcement Learning with Language Models, November 2023. URL http://arxiv.org/abs/2311.18232. arXiv:2311.18232 [cs].

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete Problems in AI Safety, July 2016. URL http://arxiv.org/abs/1606.06565. arXiv:1606.06565 [cs].

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback, April 2022. URL http://arxiv.org/abs/2204.05862. arXiv:2204.05862 [cs].

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 2397–2430. PMLR, July 2023. URL https://proceedings.mlr.press/v202/biderman23a.html. ISSN: 2640-3498.

Ching-An Cheng, Tengyang Xie, Nan Jiang, and Alekh Agarwal. Adversarially Trained Actor Critic for Offline Reinforcement Learning, July 2022. URL http://arxiv.org/abs/2202.02446. arXiv:2202.02446 [cs].

Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based Safe Policy Optimization for Continuous Control, February 2019. URL http://arxiv.org/abs/1901.10031. arXiv:1901.10031 [cs, stat].

Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness, June 2017. URL http://arxiv.org/abs/1701.08230. arXiv:1701.08230 [cs, stat].

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward Model Ensembles Help Mitigate Overoptimization, March 2024. URL http://arxiv.org/abs/2310.02743. arXiv:2310.02743 [cs].

Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe Exploration in Continuous Action Spaces, January 2018. URL http://arxiv.org/abs/1801.08757. arXiv:1801.08757 [cs].

Esther Derman, Matthieu Geist, and Shie Mannor. Twice regularized MDPs and the equivalence between robustness and regularization, October 2021. URL http://arxiv.org/abs/2110.06267. arXiv:2110.06267 [cs, math].

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S. Liang, and Tatsunori B. Hashimoto. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/5fc47800ee5b30b8777fdd30abcaaf3b-Abstract-Conference.html.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback, January 2024. URL http://arxiv.org/abs/2305.14387. arXiv:2305.14387.

Ian Fox, Joyce Lee, Rodica Pop-Busui, and Jenna Wiens. Deep Reinforcement Learning for Closed-Loop Blood Glucose Control, September 2020. URL http://arxiv.org/abs/2009.09051. arXiv:2009.09051 [cs, stat].

Matt Franchi, J. D. Zamfirescu-Pereira, Wendy Ju, and Emma Pierson. Detecting disparities in police deployments using dashcam data. In *2023 ACM Conference on Fairness, Accountability, and Transparency*, pp. 534–544, June 2023. doi: 10.1145/3593013.3594020. URL http://arxiv.org/abs/2305.15210. arXiv:2305.15210 [cs].

Scott Fujimoto, David Meger, and Doina Precup. Off-Policy Deep Reinforcement Learning without Exploration. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2052–2062. PMLR, May 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html. ISSN: 2640-3498.

Uri Gadot, Esther Derman, Navdeep Kumar, Maxence Mohamed Elfatihi, Kfir Levy, and Shie Mannor. Solving Non-Rectangular Reward-Robust MDPs via Frequency Regularization, February 2024. URL http://arxiv.org/abs/2309.01107. arXiv:2309.01107 [cs].

Leo Gao, John Schulman, and Jacob Hilton. Scaling Laws for Reward Model Overoptimization, October 2022. URL http://arxiv.org/abs/2210.10760. arXiv:2210.10760 [cs, stat].

Amelia Glaese, Nat McAleese, Maja Trebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Sona Mokra, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. Improving alignment of dialogue agents via targeted human judgements, September 2022. URL http://arxiv.org/abs/2209.14375. arXiv:2209.14375 [cs].

Adam Gleave, Michael Dennis, Shane Legg, Stuart Russell, and Jan Leike. Quantifying Differences in Reward Functions. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LwEQnp6CYev.

C. A. E. Goodhart. Problems of Monetary Management: The UK Experience. In C. A. E. Goodhart (ed.), *Monetary Theory and Practice: The UK Experience*, pp. 91–121. Macmillan Education UK, London, 1984. ISBN 978-1-349-17295-5. doi: 10.1007/978-1-349-17295-5_4. URL https://doi.org/10.1007/978-1-349-17295-5_4.

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart Russell, and Anca Dragan. Inverse Reward Design, 2017. URL http://arxiv.org/abs/1711.02827. arXiv:1711.02827 [cs].

Alexander Havrilla, Maksym Zhuravinskyi, Duy Phung, Aman Tiwari, Jonathan Tow, Stella Biderman, Quentin Anthony, and Louis Castricato. trlX: A Framework for Large Scale Reinforcement Learning from Human Feedback. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8578–8595, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.530. URL https://aclanthology.org/2023.emnlp-main.530.

Elad Hazan, Sham M. Kakade, Karan Singh, and Abby Van Soest. Provably Efficient Maximum Entropy Exploration, January 2019. URL http://arxiv.org/abs/1812.02690. arXiv:1812.02690 [cs, stat].

Haoyang He. A Survey on Offline Model-Based Reinforcement Learning, May 2023. URL http://arxiv.org/abs/2305.03360. arXiv:2305.03360 [cs, eess].

Jonathan Ho and Stefano Ermon. Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://papers.nips.cc/paper_files/paper/2016/hash/cc7e2b878868cbae992d1fb743995d8f-Abstract.html.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in Atari. 2018. doi: 10.48550/ARXIV.1811.06521. URL https://arxiv.org/abs/1811.06521. Publisher: arXiv Version Number: 1.

Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. Unpacking DPO and PPO: Disentangling Best Practices for Learning from Preference Feedback. 2024. doi: 10.48550/ARXIV.2406.09279. URL https://arxiv.org/abs/2406.09279. Publisher: arXiv Version Number: 1.

Hong Jun Jeon, Smitha Milli, and Anca D. Dragan. Reward-rational (implicit) choice: A unifying formalism for reward learning, December 2020. URL http://arxiv.org/abs/2002.04833. arXiv:2002.04833 [cs].

Bingyi Kang, Zequn Jie, and Jiashi Feng. Policy Optimization with Demonstrations. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2469–2478. PMLR, July 2018. URL https://proceedings.mlr.press/v80/kang18a.html. ISSN: 2640-3498.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. The Challenge of Understanding What Users Want: Inconsistent Preferences and Engagement Optimization, October 2023. URL http://arxiv.org/abs/2202.11776. arXiv:2202.11776 [cs].

W. Bradley Knox, Alessandro Allievi, Holger Banzhaf, Felix Schmitt, and Peter Stone. Reward (Mis)design for Autonomous Driving, March 2022. URL http://arxiv.org/abs/2104.13906. arXiv:2104.13906 [cs].

Vladimir Koltchinskii. Local Rademacher Complexities and Oracle Inequalities in Risk Minimization. *The Annals of Statistics*, 34(6):2593–2656, 2006. ISSN 0090-5364. URL https://www.jstor.org/stable/25463523. Publisher: Institute of Mathematical Statistics.

Varun Kompella, Roberto Capobianco, Stacy Jong, Jonathan Browne, Spencer Fox, Lauren Meyers, Peter Wurman, and Peter Stone. Reinforcement Learning for Optimization of COVID-19 Mitigation policies, October 2020. URL http://arxiv.org/abs/2010.10560. arXiv:2010.10560 [cs].

Tomasz Korbak, Ethan Perez, and Christopher L. Buckley. RL with KL penalties is better viewed as Bayesian inference, October 2022. URL http://arxiv.org/abs/2205.11275. arXiv:2205.11275 [cs, stat].

Victoria Krakovna. Specification gaming examples in AI, April 2018. URL https://vkrakovna.wordpress.com/2018/04/02/specification-gaming-examples-in-ai/.

Victoria Krakovna. Classifying specification problems as variants of Goodhart's Law, August 2019. URL https://vkrakovna.wordpress.com/2019/08/19/classifying-specification-problems-as-variants-of-goodharts-law/.

Victoria Krakovna, Laurent Orseau, Ramana Kumar, Miljan Martic, and Shane Legg. Penalizing side effects using stepwise relative reachability, March 2019. URL http://arxiv.org/abs/1806.01186. arXiv:1806.01186 [cs, stat].

Cassidy Laidlaw, Stuart Russell, and Anca Dragan. Bridging RL Theory and Practice with the Effective Horizon, April 2023. URL http://arxiv.org/abs/2304.09853. arXiv:2304.09853 [cs, stat].

Jongmin Lee, Cosmin Paduraru, Daniel J. Mankowitz, Nicolas Heess, Doina Precup, Kee-Eung Kim, and Arthur Guez. COptiDICE: Offline Constrained Reinforcement Learning via Stationary Distribution Correction Estimation, April 2022. URL http://arxiv.org/abs/2204.08957. arXiv:2204.08957 [cs].

Kimin Lee, Laura Smith, and Pieter Abbeel. PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training, June 2021. URL http://arxiv.org/abs/2106.05091. arXiv:2106.05091 [cs].

Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient Exploration via State Marginal Matching, February 2020. URL http://arxiv.org/abs/1906.05274. arXiv:1906.05274 [cs, stat].

Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A. Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. AI Safety Gridworlds, November 2017. URL http://arxiv.org/abs/1711.09883. arXiv:1711.09883 [cs].

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction, November 2018. URL http://arxiv.org/abs/1811.07871. arXiv:1811.07871 [cs, stat].

Eric Liang, Richard Liaw, Philipp Moritz, Robert Nishihara, Roy Fox, Ken Goldberg, Joseph E. Gonzalez, Michael I. Jordan, and Ion Stoica. RLlib: Abstractions for Distributed Reinforcement Learning, June 2018. URL http://arxiv.org/abs/1712.09381. arXiv:1712.09381 [cs].

Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wiessner. Microscopic Traffic Simulation using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582, November 2018. doi: 10.1109/ITSC.2018.8569938. URL https://ieeexplore.ieee.org/document/8569938. ISSN: 2153-0017.

Kristian Lum and William Isaac. To Predict and Serve? *Significance*, 13(5):14–19, October 2016. ISSN 1740-9705. doi: 10.1111/j.1740-9713.2016.00960.x. URL https://doi.org/10.1111/j.1740-9713.2016.00960.x. _eprint: https://academic.oup.com/jrssig/article-pdf/13/5/14/49106469/sign_13_5_14.pdf.

Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton, Boris Kovatchev, and Claudio Cobelli. The UVA/PADOVA Type 1 Diabetes Simulator. *Journal of Diabetes Science and Technology*, 8(1):26–34, January 2014. ISSN 1932-2968. doi: 10.1177/1932296813514502. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4454102/.

Debmalya Mandal, Stelios Triantafyllou, and Goran Radanovic. Performative Reinforcement Learning, February 2023. URL http://arxiv.org/abs/2207.00046. arXiv:2207.00046 [cs].

David Manheim and Scott Garrabrant. Categorizing Variants of Goodhart's Law, February 2019. URL http://arxiv.org/abs/1803.04585. arXiv:1803.04585 [cs, q-fin, stat].

Lev McKinney, Yawen Duan, David Krueger, and Adam Gleave. On The Fragility of Learned Reward Functions, January 2023. URL http://arxiv.org/abs/2301.03652. arXiv:2301.03652 [cs].

Smitha Milli, Luca Belli, and Moritz Hardt. From Optimizing Engagement to Measuring Value. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 714–722, March 2021. doi: 10.1145/3442188.3445933. URL http://arxiv.org/abs/2008.12623. arXiv:2008.12623 [cs, stat].

Samuel Mwalili, Mark Kimathi, Viona Ojiambo, Duncan Gathungu, and Rachel Mbogo. SEIR model for COVID-19 dynamics incorporating the environment and social distancing. *BMC Research Notes*, 13:352, July 2020. ISSN 1756-0500. doi: 10.1186/s13104-020-05192-1. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7376536/.

Alexander Nedergaard and Matthew Cook. k-Means Maximum Entropy Exploration, November 2023. URL http://arxiv.org/abs/2205.15623. arXiv:2205.15623 [cs].

Richard Ngo, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep learning perspective, September 2023. URL http://arxiv.org/abs/2209.00626. arXiv:2209.00626 [cs].

Andi Nika, Debmalya Mandal, Parameswaran Kamalaruban, Georgios Tzannetos, Goran Radanovic, and Adish Singla. Reward Model Learning vs. Direct Policy Optimization: A Comparative Analysis of Learning from Human Preferences. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 38145–38186. PMLR, July 2024. URL https://proceedings.mlr.press/v235/nika24a.html. ISSN: 2640-3498.

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, October 2019. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aax2342. URL https://www.science.org/doi/10.1126/science.aax2342.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. 2022. doi: 10.48550/ARXIV.2203.02155. URL https://arxiv.org/abs/2203.02155.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The Effects of Reward Misspecification: Mapping and Mitigating Misaligned Models, February 2022. URL http://arxiv.org/abs/2201.03544. arXiv:2201.03544 [cs, stat].

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, December 2019. URL http://arxiv.org/abs/1912.01703. arXiv:1912.01703 [cs, stat].

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging Offline Reinforcement Learning and Imitation Learning: A Tale of Pessimism, July 2023. URL http://arxiv.org/abs/2103.12021. arXiv:2103.12021 [cs, math, stat].

Siddharth Reddy, Anca Dragan, Sergey Levine, Shane Legg, and Jan Leike. Learning Human Objectives by Evaluating Hypothetical Behavior. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 8020–8029. PMLR, November 2020. URL https://proceedings.mlr.press/v119/reddy20a.html. ISSN: 2640-3498.

Julien Roy, Roger Girgis, Joshua Romoff, Pierre-Luc Bacon, and Christopher Pal. Direct Behavior Specification via Constrained Reinforcement Learning, June 2022. URL http://arxiv.org/abs/2112.12228. arXiv:2112.12228 [cs].

Stuart J. Russell, Peter Norvig, and Ernest Davis. *Artificial intelligence: a modern approach*. Prentice Hall series in artificial intelligence. Prentice Hall, Upper Saddle River, 3rd ed edition, 2010. ISBN 978-0-13-604259-4.

Dorsa Sadigh, Anca Dragan, Shankar Sastry, and Sanjit Seshia. Active Preference-Based Learning of Reward Functions. In *Robotics: Science and Systems XIII*. Robotics: Science and Systems Foundation, July 2017. ISBN 978-0-9923747-3-0. doi: 10.15607/RSS.2017.XIII.053. URL http://www.roboticsproceedings.org/rss13/p53.pdf.

John Schulman. Approximating KL Divergence, March 2020. URL http://joschu.net/blog/kl-approx.html.

Lior Shani, Aviv Rosenberg, Asaf Cassel, Oran Lang, Daniele Calandriello, Avital Zipori, Hila Noga, Orgad Keller, Bilal Piot, and Idan Szpektor. Multi-turn Reinforcement Learning from Preference Human Feedback. *arXiv preprint arXiv:2405.14655*, 2024. URL https://arxiv.org/abs/2405.14655.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and Characterizing Reward Hacking, September 2022. URL http://arxiv.org/abs/2209.13085. arXiv:2209.13085 [cs, stat].

Garry M. Steil. Algorithms for a Closed-Loop Artificial Pancreas: The Case for Proportional-Integral-Derivative Control. *Journal of Diabetes Science and Technology*, 7(6):1621–1631, November 2013. ISSN 1932-2968. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3876341/.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, September 2020. URL http://arxiv.org/abs/2009.01325. arXiv:2009.01325 [cs].

Jonathan Stray, Alon Halevy, Parisa Assar, Dylan Hadfield-Menell, Craig Boutilier, Amar Ashar, Lex Beattie, Michael Ekstrand, Claire Leibowicz, Connie Moon Sehat, Sara Johansen, Lianne Kerlin, David Vickrey, Spandana Singh, Sanne Vrijenhoek, Amy Zhang, McKane Andrus, Natali Helberger, Polina Proutskova, Tanushree Mitra, and Nina Vasan. Building Human Values into Recommender Systems: An Interdisciplinary Synthesis, July 2022. URL http://arxiv.org/abs/2207.10192. arXiv:2207.10192 [cs].

Jessica Taylor. Quantilizers: A Safer Alternative to Maximizers for Limited Optimization. March 2016. URL https://www.semanticscholar.org/paper/Quantilizers%3A-A-Safer-Alternative-to-Maximizers-for-Taylor/4e8ff3b4069a12a00196d62925bab8add7389742.

Jessica Taylor, Eliezer Yudkowsky, Patrick LaVictoire, and Andrew Critch. Alignment for Advanced Machine Learning Systems. In *Ethics of Artificial Intelligence*. Oxford University Press, August 2020. ISBN 978-0-19-090505-7. Google-Books-ID: 1yT3DwAAQBAJ.

Jeremy Tien, Jerry Zhi-Yang He, Zackory Erickson, Anca D. Dragan, and Daniel S. Brown. Causal Confusion and Reward Misidentification in Preference-Based Reward Learning, March 2023. URL http://arxiv.org/abs/2204.06601. arXiv:2204.06601 [cs].

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E*, 62(2):1805–1824, August 2000. ISSN 1063-651X, 1095-3787. doi: 10.1103/PhysRevE.62.1805. URL http://arxiv.org/abs/cond-mat/0002177. arXiv:cond-mat/0002177.

A. M. Turner, Logan Smith, Rohin Shah, Andrew Critch, and Prasad Tadepalli. Optimal Policies Tend To Seek Power. December 2019. URL https://www.semanticscholar.org/paper/Optimal-Policies-Tend-To-Seek-Power-Turner-Smith/46d4452eb041e33f1e58eab64ec8cf5af534b6ff.

Alexander Matt Turner, Neale Ratzlaff, and Prasad Tadepalli. Avoiding Side Effects in Complex Environments, October 2020. URL http://arxiv.org/abs/2006.06547. arXiv:2006.06547 [cs].

Ikechukwu Uchendu, Ted Xiao, Yao Lu, Banghua Zhu, Mengyuan Yan, Joséphine Simon, Matthew Bennice, Chuyuan Fu, Cong Ma, Jiantao Jiao, Sergey Levine, and Karol Hausman. Jump-Start Reinforcement Learning, July 2023. URL http://arxiv.org/abs/2204.02372. arXiv:2204.02372 [cs].

Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the Average: an Analysis of KL Regularization in RL, January 2021. URL http://arxiv.org/abs/2003.14089. arXiv:2003.14089 [cs, stat].

Eugene Vinitsky, Aboudy Kreidieh, Luc Le Flem, Nishant Kheterpal, Kathy Jang, Cathy Wu, Fangyu Wu, Richard Liaw, Eric Liang, and Alexandre M. Bayen. Benchmarks for reinforcement learning in mixed-autonomy traffic. In *Proceedings of The 2nd Conference on Robot Learning*, pp. 399–409. PMLR, October 2018. URL https://proceedings.mlr.press/v87/vinitsky18a.html. ISSN: 2640-3498.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. Mint: Evaluating llms in multi-turn interaction with tools and language feedback. *arXiv preprint arXiv:2309.10691*, 2023. URL https://arxiv.org/abs/2309.10691.

Cathy Wu, Aboudy Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M. Bayen. Flow: A Modular Learning Framework for Mixed Autonomy Traffic. *IEEE Transactions on Robotics*, 38 (2):1270–1286, April 2022. ISSN 1552-3098, 1941-0468. doi: 10.1109/TRO.2021.3087314. URL http://arxiv.org/abs/1710.05465. arXiv:1710.05465 [cs].

Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent Pessimism for Offline Reinforcement Learning, October 2023. URL http://arxiv.org/abs/2106.06926. arXiv:2106.06926 [cs, stat].

Shentao Yang, Yihao Feng, Shujian Zhang, and Mingyuan Zhou. Regularizing a Model-based Policy Stationary Distribution to Stabilize Offline Reinforcement Learning, June 2022. URL http://arxiv.org/abs/2206.07166. arXiv:2206.07166 [cs].

Yiming Zhang, Quan Vuong, and Keith W. Ross. First Order Constrained Optimization in Policy Space, October 2020. URL http://arxiv.org/abs/2002.06506. arXiv:2002.06506 [cs, stat].

Appendix

A Proofs and Additional Theoretical Results

A.1 Proof of Theorem 5.1

Theorem 5.1. *Suppose that $\tilde{R}$ is an r-correlated-proxy for the true reward function R, and let $\sigma_{\tilde{R}}$ and σ_R be defined as in Definition 4.1. Then for any policy π such that $\mu_\pi \ll \mu_{\pi_{base}}$ (i.e., $\mu_{\pi_{base}}(s,a) = 0 \Rightarrow \mu_\pi(s,a) = 0$), we have*

$$\frac{J(\pi, R) - J(\pi_{base}, R)}{\sigma_R} \geq \frac{1}{r}\left(\frac{J(\pi, \tilde{R}) - J(\pi_{base}, \tilde{R})}{\sigma_{\tilde{R}}} - \sqrt{(1-r^2)\chi^2\left(\mu_\pi \| \mu_{\pi_{base}}\right)}\right), \tag{4}$$

where $\chi^2\left(\mu_\pi \| \mu_{\pi_{base}}\right) = \mathbb{E}_{\mu_\pi}\left[\frac{\mu_\pi(s,a)}{\mu_{\pi_{base}}(s,a)} - 1\right]$ is the χ^2 divergence between μ_π and $\mu_{\pi_{base}}$.

Proof. For simplicity of exposition, define

$$Z(s,a) = \frac{R(s,a) - J(\pi_{\text{base}}, R)}{\sigma_R} \qquad \text{and} \qquad \tilde{Z}(s,a) = \frac{\tilde{R}(s,a) - J(\pi_{\text{base}}, \tilde{R})}{\sigma_{\tilde{R}}}.$$

Using (3), we can rewrite (10) as

$$\mathbb{E}_{\mu_\pi}\left[\tilde{Z}(s,a) - r\,Z(s,a)\right] \leq \sqrt{(1-r^2)\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right)}.$$

Then, the left hand side can be rewritten as

$$\begin{aligned}
&\mathbb{E}_{\mu_\pi}\left[\tilde{Z}(s,a) - r\,Z(s,a)\right] \\
&\quad = \mathbb{E}_{\mu_\pi}\left[\tilde{Z}(s,a) - r\,Z(s,a)\right] - \mathbb{E}_{\mu_{\pi_{\text{base}}}}\left[\tilde{Z}(s,a) - r\,Z(s,a)\right] + \mathbb{E}_{\mu_{\pi_{\text{base}}}}\left[\tilde{Z}(s,a) - r\,Z(s,a)\right] \\
&\quad = \mathbb{E}_{\mu_\pi}\left[\tilde{Z}(s,a) - r\,Z(s,a)\right] - \mathbb{E}_{\mu_{\pi_{\text{base}}}}\left[\tilde{Z}(s,a) - r\,Z(s,a)\right],
\end{aligned}$$

since by definition $\mathbb{E}_{\mu_{\pi_{\text{base}}}}[\tilde{Z}(s,a)] = \mathbb{E}_{\mu_{\pi_{\text{base}}}}[Z(s,a)] = 0$. Applying the Cauchy-Schwartz inequality to this difference gives

$$\begin{aligned}
&\mathbb{E}_{\mu_\pi}\left[\tilde{Z}(s,a) - r\,Z(s,a)\right] - \mathbb{E}_{\mu_{\pi_{\text{base}}}}\left[\tilde{Z}(s,a) - r\,Z(s,a)\right] \\
&\quad = \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left[\tilde{Z}(s,a) - r\,Z(s,a)\right]\left[\mu_\pi(s,a) - \mu_{\pi_{\text{base}}}(s,a)\right] \\
&\quad = \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left[\sqrt{\mu_{\pi_{\text{base}}}(s,a)}\left(\tilde{Z}(s,a) - r\,Z(s,a)\right)\right]\left[\frac{\mu_\pi(s,a) - \mu_{\pi_{\text{base}}}(s,a)}{\sqrt{\mu_{\pi_{\text{base}}}(s,a)}}\right] \\
&\quad \leq \sqrt{\left(\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mu_{\pi_{\text{base}}}(s,a)\left[\tilde{Z}(s,a) - r\,Z(s,a)\right]^2\right)\left(\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{(\mu_\pi(s,a) - \mu_{\pi_{\text{base}}}(s,a))^2}{\mu_{\pi_{\text{base}}}(s,a)}\right)} \\
&\quad = \sqrt{\mathbb{E}_{\mu_{\pi_{\text{base}}}}\left[\left(\tilde{Z}(s,a) - r\,Z(s,a)\right)^2\right]\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right)}.
\end{aligned} \tag{9}$$

The expectation can be calculated as

$$\begin{aligned}
&\mathbb{E}_{\mu_{\pi_{\text{base}}}}\left[\left(\tilde{Z}(s,a) - r\,Z(s,a)\right)^2\right] \\
&= \text{Var}_{\mu_{\pi_{\text{base}}}}\left(\tilde{Z}(s,a) - r\,Z(s,a)\right) \\
&= \text{Var}_{\mu_{\pi_{\text{base}}}}\left(\tilde{Z}(s,a)\right) + r^2\,\text{Var}_{\mu_{\pi_{\text{base}}}}\left(Z(s,a)\right) - 2r\,\text{Cov}_{\mu_{\pi_{\text{base}}}}\left(\tilde{Z}(s,a), Z(s,a)\right) \\
&= 1 - r^2,
\end{aligned}$$

using the fact that both reward functions have unit variance under the base policy and that their correlation is r. Plugging this into (9) gives the desired result. □

A.1.1 Near-Optimal Base Policies

While Theorem 5.1 shows that optimizing the proxy reward with regularization can improve on the base policy's reward, it does not guarantee that the learned policy will be near-optimal. However, a simple corollary shows that if the base policy is near-optimal, then the learned policy will also be near-optimal.

Corollary A.1. *Suppose that $\tilde{R}$ is an r-correlated-proxy for the true reward function R, and let $\sigma_{\tilde{R}}$ and σ_R be defined as in Definition 4.1. Furthermore, suppose that the base policy π_{base} is near-optimal:*

$$J(\pi_{base}, R) \geq \max_{\pi^*} J(\pi^*, R) - \epsilon\sigma_R,$$

for some $\epsilon > 0$.

Then for any policy π such that $\mu_\pi \ll \mu_{\pi_{base}}$, we have

$$\frac{\max_{\pi^*} J(\pi^*, R) - J(\pi, R)}{\sigma_R} \leq \epsilon - \frac{1}{r}\left(\frac{J(\pi, \tilde{R}) - J(\pi_{base}, \tilde{R})}{\sigma_{\tilde{R}}} - \sqrt{(1-r^2)\chi^2\left(\mu_\pi \| \mu_{\pi_{base}}\right)}\right). \tag{10}$$

This result bounds the suboptimality gap—how close the learned policy is to optimal—in terms of the suboptimality gap of the base policy and the increase in the regularized proxy reward. The proof is straightforward and follows from Theorem 5.1.

A.1.2 Understanding the Lower Bound in Theorem 5.1

Denote by

$$L(\pi) = \frac{1}{r}\left(\frac{J(\pi, \tilde{R}) - J(\pi_{\text{base}}, \tilde{R})}{\sigma_{\tilde{R}}} - \sqrt{(1-r^2)\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right)}\right) \tag{11}$$

the lower bound on increase in the true reward which is the RHS of (10). One surprising observation is that this lower bound seems to be *increasing* as the proxy becomes *less* correlated with the true reward. This would suggest that a less correlated proxy leads to better optimization of the true reward. However, as the following lemma shows, $L(\pi)$ is actually decreasing in r.

Lemma A.2. *Under the same conditions as Theorem 5.1, the lower bound $L(\pi)$ satisfies*

$$L(\pi) \leq \frac{1 - \sqrt{1-r^2}}{r}\sqrt{\chi^2\left(\mu_\pi \| \mu_{\pi_{base}}\right)}. \tag{12}$$

This shows that the lower bound can be at most a factor of $\frac{1-\sqrt{1-r^2}}{r}$ times the divergence between the learned and base policies' occupancy measures. This factor is increasing in r and asymptotes to $r/2$ as $r \to 0$.

Proof. Using the same notation as the proof of Theorem 5.1, we can rewrite the lower bound as

$$L(\pi) = \frac{1}{r}\left(\mathbb{E}_\pi\left[\tilde{Z}(s,a)\right] - \sqrt{(1-r^2)\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right)}\right).$$

Following a similar argument to that in the proof of Theorem 5.1, we can write

$$\begin{aligned}
\mathbb{E}_\pi\left[\tilde{Z}(s,a)\right] &= \mathbb{E}_\pi\left[\tilde{Z}(s,a)\right] - \mathbb{E}_{\pi_{\text{base}}}\left[\tilde{Z}(s,a)\right] \\
&= \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left[\sqrt{\mu_{\pi_{\text{base}}}(s,a)}\tilde{Z}(s,a)\right]\left[\frac{\mu_\pi(s,a) - \mu_{\pi_{\text{base}}}(s,a)}{\sqrt{\mu_{\pi_{\text{base}}}(s,a)}}\right] \\
&\leq \sqrt{\left(\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mu_{\pi_{\text{base}}}(s,a)\tilde{Z}(s,a)^2\right)\left(\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{(\mu_\pi(s,a) - \mu_{\pi_{\text{base}}}(s,a))^2}{\mu_{\pi_{\text{base}}}(s,a)}\right)} \\
&= \sqrt{\mathbb{E}_{\mu_{\pi_{\text{base}}}}\left[\tilde{Z}(s,a)^2\right]\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right)} \\
&= \sqrt{\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right)}.
\end{aligned}$$

Combining this with the definition of $L(\pi)$ gives

$$\begin{aligned}
L(\pi) &\leq \frac{1}{r}\left(\sqrt{\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right)} - \sqrt{(1-r^2)\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right)}\right) \\
&= \frac{1-\sqrt{1-r^2}}{r}\sqrt{\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right)},
\end{aligned}$$

which completes the proof. □

A.1.3 Is the Lower Bound Optimizable?

While Theorem 5.1 shows that the increase in true reward over the base policy can be lower-bounded by the the proxy reward minus a regularization term, it is not clear if it is actually possible to increase the lower bound $L(\pi)$ as defined in (11). For example, if the base policy is already optimal with respect to the proxy reward, then clearly $L(\pi) \leq 0$. As another example, suppose it possible to improve π_{base} with respect to both the true and proxy rewards, but only by visiting a state-action pair never visited by π_{base}. In this case, it is also impossible to improve the lower bound in Theorem 5.1 while obeying the requirement that $\mu_\pi \ll \mu_{\pi_{\text{base}}}$.

As the following lemma shows, there are even more general cases when the lower bound cannot be optimized. However, as we show in our experiments, in many realistic environments it does appear that the lower bound is optimizable. Furthermore, Theorem 5.1 still allows for safe optimization of the proxy reward: even if the it not possible to increase the lower bound above zero, optimizing $L(\pi)$ will at least prevent reward hacking.

Lemma A.3. *Fix any $r \in (0,1)$. Then there is an MDP with a true reward function R, a proxy reward $\tilde{R}$, and a base policy π_{base} such that $\tilde{R}$ is an r-correlated proxy that can be improved upon in both true and proxy reward by a policy π^*:*

$$\begin{aligned}
J(\pi^*, R) &> J(\pi_{base}, R) \\
J(\pi^*, \tilde{R}) &> J(\pi_{base}, \tilde{R}) \\
\mu_{\pi^*} &\ll \mu_{\pi_{base}}.
\end{aligned}$$

However, for any policy π such that $\mu_\pi \ll \mu_{\pi_{base}}$,

$$L(\pi) = \frac{1}{r}\left(\frac{J(\pi,\tilde{R}) - J(\pi_{base},\tilde{R})}{\sigma_{\tilde{R}}} - \sqrt{(1-r^2)\chi^2\left(\mu_\pi \| \mu_{\pi_{base}}\right)}\right) \leq 0.$$

That is, Lemma A.3 shows that there is an MDP where there exists a policy π^* that improves on the

base policy in both true reward and proxy reward, but it is still not possible to increase the lower bound $L(\pi)$ above zero. This suggests that it is difficult to specify general conditions under which $L(\pi)$ can exceed zero. Thus, we rely on the empirical evidence from our experiments to show that the lower bound is often optimizable.

Proof. We consider MDPs with discount $\gamma = 0$, such that the transition probabilities are not relevant; only the initial state distribution μ_0 and the reward functions specify the MDP. We split the analysis into two cases depending on whether $r \leq 1/2$ or $r \geq 1/2$.

Case 1: $r \leq 1/2$**.** We define an MDP with two states s_1, s_2 and two actions a_1, a_2, with initial state distribution and rewards as follows:

$$\mu_0(s_1) = \frac{1}{1+r} \qquad R(s_1, a_1) = \sqrt{\frac{r}{1-r}} \qquad \tilde{R}(s_1, a_1) = \sqrt{\frac{r}{1-r}}$$

$$R(s_1, a_2) = -\sqrt{\frac{1-r}{r}} \qquad \tilde{R}(s_1, a_2) = 0$$

$$\mu_0(s_2) = \frac{r}{1+r} \qquad R(s_2, \cdot) = 0 \qquad \tilde{R}(s_2, \cdot) = -\sqrt{\frac{1-r}{r}}$$

Furthermore, we define $\pi_{\text{base}}(a_1 \mid s_1) = 1 - r$ and $\pi_{\text{base}}(a_2 \mid s_1) = r$, and $\pi_{\text{base}}(a_1 \mid s_2) = 1$. Based on this, simple algebra shows the following facts:

$$\mu_{\pi_{\text{base}}}(s_1, a_1) = \frac{1-r}{1+r} \qquad \mu_{\pi_{\text{base}}}(s_1, a_2) = \frac{r}{1+r}$$

$$J(\pi_{\text{base}}, R) = J(\pi_{\text{base}}, \tilde{R}) = 0 \qquad \sigma_R = \sigma_{\tilde{R}} = \sqrt{\frac{1}{1+r}}$$

$$\mathbb{E}_{\mu_{\pi_{\text{base}}}}\left[R(s,a)\tilde{R}(s,a)\right] = \frac{r}{1+r}.$$

Based on this, it is clear that $\tilde{R}$ is an r-correlated proxy. Furthermore, letting $\pi^*(a_1 \mid s_1) = 1$ and $\pi^*(a_1 \mid s_2) = 1$, we have

$$J(\pi^*, R) = \frac{1}{1+r}\sqrt{\frac{r}{1-r}} > 0 = J(\pi_{\text{base}}, R)$$

$$J(\pi^*, \tilde{R}) = \frac{r}{1+r}\sqrt{\frac{r}{1-r}} > 0 = J(\pi_{\text{base}}, \tilde{R})$$

$$\mu_{\pi^*} \ll \mu_{\pi_{\text{base}}},$$

satisfying the conditions in the lemma.

Now, consider any policy π such that $\mu_\pi \ll \mu_{\pi_{\text{base}}}$. We can calculate the lower bound $L(\pi)$ (ignoring the prefactor of $\frac{1}{r}$) as

$$J(\pi, \tilde{R})\sqrt{1+r} - \sqrt{(1-r^2)\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right)}. \tag{13}$$

Since $\mu_\pi \ll \mu_{\pi_{\text{base}}}$, $\pi(a_1 \mid s_2) = 1$, so it can only differ from π_{base} in state s_1. Let $\delta = \pi(a_1 \mid s_1) - \pi_{\text{base}}(a_1 \mid s_1)$. Then, we can write the first term of (13) as

$$J(\pi, \tilde{R})\sqrt{1+r} = \delta\frac{1}{1+r}\sqrt{\frac{r}{1-r}}\sqrt{1+r} = \delta\sqrt{\frac{r}{1-r^2}}.$$

Note that the χ^2 divergence between distributions μ and ν can be alternatively written as

$$\chi^2(\mu \| \nu) = \sum_{s,a} \frac{(\mu(s,a) - \nu(s,a))^2}{\nu(s,a)}.$$

Therefore, the χ^2 divergence between μ_π and $\mu_{\pi_{\text{base}}}$ can be lower bounded as

$$\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right) \geq \frac{\left(\mu_\pi(s_1,a_1) - \mu_{\pi_{\text{base}}}(s_1,a_1)\right)^2}{\mu_{\pi_{\text{base}}}(s_1,a_1)} = \frac{\left(\frac{1-r+\delta}{1+r} - \frac{1-r}{1+r}\right)^2}{\frac{1-r}{1+r}} = \frac{\delta^2}{1-r^2}.$$

This leads to the bound on (13):

$$rL(\pi) \leq \delta\sqrt{\frac{r}{1-r^2}} - \sqrt{(1-r^2)\frac{\delta^2}{1-r^2}} = \delta\sqrt{\frac{r}{1-r^2}} - |\delta|.$$

Clearly if $\delta \leq 0$ this is non-positive, and if $\delta > 0$ it is also non-positive since $\sqrt{r/(1-r^2)} < 1$ as long as $r \leq 1/2$. Thus, the lower bound cannot be increased above zero in this case.

Case 2: $r \geq 1/2$**.** In this case, we define an MDP with three states s_1, s_2, s_3 and two actions a_1, a_2, with initial state distribution and rewards as follows:

$$\begin{aligned}
\mu_0(s_1) &= \frac{2r^2-2r+1}{r^2-r+1} & R(s_1,a_1) &= \sqrt{\frac{1-r}{r}} & \tilde{R}(s_1,a_1) &= \sqrt{\frac{1-r}{r}} \\
& & R(s_1,a_2) &= -\sqrt{\frac{r}{1-r}} & \tilde{R}(s_1,a_2) &= 0 \\
\mu_0(s_2) &= \frac{(1-r)^2}{r^2-r+1} & R(s_2,\cdot) &= 0 & \tilde{R}(s_2,\cdot) &= -\sqrt{\frac{r}{1-r}} \\
\mu_0(s_3) &= \frac{(1-r)(2r-1)}{r^2-r+1} & R(s_3,\cdot) &= -\sqrt{\frac{r}{1-r}} & \tilde{R}(s_3,\cdot) &= -\sqrt{\frac{r}{1-r}}
\end{aligned}$$

We define the base policy π_{base} as follows:

$$\begin{aligned}
\pi_{\text{base}}(a_1 \mid s_1) &= \frac{r^2}{2r^2-2r+1} & \pi_{\text{base}}(a_2 \mid s_1) &= \frac{(1-r)^2}{2r^2-2r+1} \\
\pi_{\text{base}}(a_1 \mid s_2) &= 1 & \pi_{\text{base}}(a_1 \mid s_3) &= 1.
\end{aligned}$$

As above, we can show the following facts:

$$\begin{aligned}
\mu_{\pi_{\text{base}}}(s_1,a_1) &= \frac{r^2}{r^2-r+1} & \mu_{\pi_{\text{base}}}(s_1,a_2) &= \frac{(1-r)^2}{r^2-r+1} \\
J(\pi_{\text{base}}, R) = J(\pi_{\text{base}}, \tilde{R}) &= 0 & \sigma_R = \sigma_{\tilde{R}} &= \sqrt{\frac{r}{r^2-r+1}} \\
\mathbb{E}_{\mu_{\pi_{\text{base}}}}\left[R(s,a)\tilde{R}(s,a)\right] &= \frac{r^2}{r^2-r+1}.
\end{aligned}$$

Again, this shows that $\tilde{R}$ is an r-correlated proxy. Letting $\pi^*(a_1 \mid s_1) = \pi^*(a_1 \mid s_2) = \pi^*(a_1 \mid s_3) = 1$, we have

$$\begin{aligned}
J(\pi^*, R) &= \frac{1-r}{r^2-r+1}\sqrt{\frac{1-r}{r}} > 0 = J(\pi_{\text{base}}, R) \\
J(\pi^*, \tilde{R}) &= \frac{(1-r)^2}{r^2-r+1}\sqrt{\frac{1-r}{r}} > 0 = J(\pi_{\text{base}}, \tilde{R}) \\
\mu_{\pi^*} &\ll \mu_{\pi_{\text{base}}},
\end{aligned}$$

satisfying the conditions in the lemma.

Now, consider any policy π such that $\mu_\pi \ll \mu_{\pi_{\text{base}}}$. We can calculate the lower bound $L(\pi)$ (again

ignoring the prefactor of $\frac{1}{r}$) as

$$J(\pi, \tilde{R})\sqrt{\frac{r^2 - r + 1}{r}} - \sqrt{(1 - r^2)\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right)}. \tag{14}$$

As in the previous case, let $\delta = \pi(a_1 \mid s_1) - \pi_{\text{base}}(a_1 \mid s_1)$. Then, we can write the first term of (14) as

$$J(\pi, \tilde{R})\sqrt{\frac{r^2 - r + 1}{r}} = \delta\frac{2r^2 - 2r + 1}{r^2 - r + 1}\sqrt{\frac{1 - r}{r}}\sqrt{\frac{r^2 - r + 1}{r}} = \delta\frac{2r^2 - 2r + 1}{r}\sqrt{\frac{1 - r}{r^2 - r + 1}}.$$

The χ^2 divergence between μ_π and $\mu_{\pi_{\text{base}}}$ can be lower bounded as

$$\chi^2\left(\mu_\pi \| \mu_{\pi_{\text{base}}}\right) \geq \frac{\left(\mu_\pi(s_1, a_1) - \mu_{\pi_{\text{base}}}(s_1, a_1)\right)^2}{\mu_{\pi_{\text{base}}}(s_1, a_1)} = \frac{\left(\frac{2r^2 - 2r + 1}{r^2 - r + 1}\delta\right)^2}{\frac{r^2}{r^2 - r + 1}} = \delta^2\left(\frac{2r^2 - 2r + 1}{r}\right)^2\frac{1}{r^2 - r + 1}.$$

This leads to the bound on (14):

$$\begin{aligned} rL(\pi) &\leq \delta\frac{2r^2 - 2r + 1}{r}\sqrt{\frac{1 - r}{r^2 - r + 1}} - \sqrt{(1 - r^2)\delta^2\left(\frac{2r^2 - 2r + 1}{r}\right)^2\frac{1}{r^2 - r + 1}} \\ &= \frac{2r^2 - 2r + 1}{r\sqrt{r^2 - r + 1}}\left(\delta\sqrt{1 - r} - |\delta|\sqrt{1 - r^2}\right). \end{aligned}$$

The prefactor is positive, and as in the previous case, if $\delta \leq 0$ then the bound in non-positive. If $\delta > 0$, then the bound is also non-positive since $\sqrt{1 - r} < \sqrt{1 - r^2}$ for any $r \in (0, 1)$. Thus, the lower bound cannot be increased above zero in this case either, completing the proof. □

A.2 Failure of Action Distribution Regularization

As discussed in the main text, the OM regularization method we propose differs from the AD-based regularization found in previous work on RLHF. The following theorem shows that almost *any* form of policy optimization with action distribution regularization cannot guarantee an improvement in true reward over the base policy.

Theorem A.4. *Fix $r \in (0, 1)$. Consider a policy optimization objective regularized by any f-divergence between the action distributions of the learned policy and the base policy:*

$$\textit{maximize} \qquad L'(\pi) = J(\pi, \tilde{R}) - J(\pi_{base}, \tilde{R}) - g\left((1 - \gamma)\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t D_f\left(\pi(\cdot \mid s_t) \parallel \pi_{base}(\cdot \mid s_t)\right)\right]\right),$$

where $f : [0, \infty) \to \mathbb{R}$ is a convex, continuous function with $f(1) = 0$, $g : [0, \infty) \to [0, \infty)$ is a strictly increasing function with $g(0) = 0$, and D_f is the f-divergence:

$$D_f(P \parallel Q) = \mathbb{E}_{x \sim Q}\left[f\left(\frac{P(x)}{Q(x)}\right)\right].$$

Then there is an MDP, reward functions $R, \tilde{R}$, and base policy π_{base} such that $\tilde{R}$ is an r-correlated proxy for R, but there is a policy $\widetilde{\pi}$ such that

$$L'(\widetilde{\pi}) > 0 \qquad \textit{but} \qquad J(\widetilde{\pi}, R) < J(\pi_{base}, R).$$

Theorem A.4 concerns a general form of policy optimization with action distribution regularization; it considers *any* f-divergence between action distributions and *any* way of scaling the f-divergence using the function g. For instnce, g could incorporate linear scaling of the divergence as in the KL regularization used in previous work, square-root scaling as we use with χ^2 divergence in Theorem 5.1, or any other scaling. The theorem shows that no matter how the regularization is formulated, it cannot guarantee improvement in true reward over the base policy; there is a policy that increases the regularized objective but decreases the true reward.

Proof. Define the inverse $g^{-1} : [0, \infty) \to [0, \infty]$ as

$$g^{-1}(x) = \sup \{ y \in [0, \infty) \mid g(y) \leq x \}.$$

Since f is continuous and $f(1) = 0$, there must be a radius $\rho > 0$ such that

$$|u - 1| \leq \rho \qquad \Rightarrow \qquad f(u) < \frac{2g^{-1}\left(\frac{1-r}{8}\right)}{1-r}.$$

We construct an MDP with discount factor

$$\gamma = \begin{cases} \max\left\{1 - \frac{2g^{-1}(\frac{1-r}{8})}{(1-r)f(2)}, \frac{1}{1+\rho}, \frac{1}{2}\right\} & f(2) > 0 \\ \max\left\{\frac{1}{1+\rho}, \frac{1}{2}\right\} & \text{otherwise.} \end{cases}$$

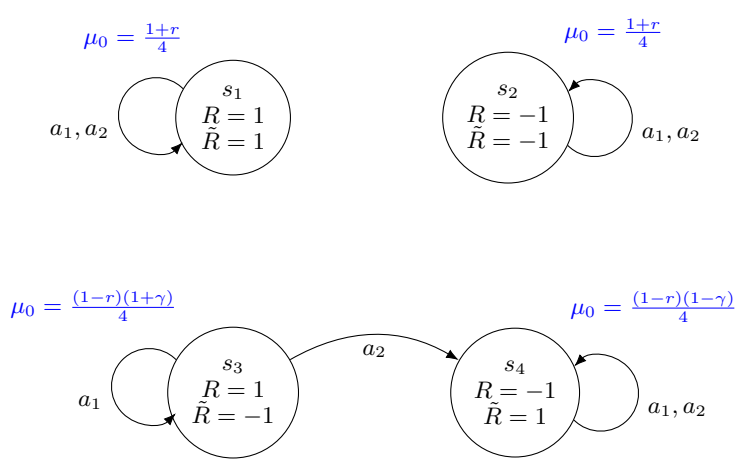
There are four states and two actions. The initial state distribution and transition probabilities are

$$\begin{aligned}
\mu_0(s_1) &= \frac{1+r}{4} & p(s_1 \mid s_1, a_1) &= 1 & p(s_1 \mid s_1, a_2) &= 1 \\
\mu_0(s_2) &= \frac{1+r}{4} & p(s_2 \mid s_2, a_1) &= 1 & p(s_2 \mid s_2, a_2) &= 1 \\
\mu_0(s_3) &= \frac{(1-r)(1+\gamma)}{4} & p(s_3 \mid s_3, a_1) &= 1 & p(s_4 \mid s_3, a_2) &= 1 \\
\mu_0(s_4) &= \frac{(1-r)(1-\gamma)}{4} & p(s_4 \mid s_4, a_1) &= 1 & p(s_4 \mid s_4, a_2) &= 1.
\end{aligned}$$

The reward functions only depend on the state and are given by

$$\begin{aligned}
R(s_1, \cdot) &= 1 & \tilde{R}(s_1, \cdot) &= 1 \\
R(s_2, \cdot) &= -1 & \tilde{R}(s_2, \cdot) &= -1 \\
R(s_3, \cdot) &= 1 & \tilde{R}(s_3, \cdot) &= -1 \\
R(s_4, \cdot) &= -1 & \tilde{R}(s_4, \cdot) &= 1.
\end{aligned}$$

In graphical form, the MDP is as follows:



We consider the base policy defined by

$$\begin{aligned}
\pi_{\text{base}}(a_1 \mid s_1) &= 1 & \pi_{\text{base}}(a_2 \mid s_1) &= 0 \\
\pi_{\text{base}}(a_1 \mid s_2) &= 1 & \pi_{\text{base}}(a_2 \mid s_2) &= 0 \\
\pi_{\text{base}}(a_1 \mid s_3) &= \gamma & \pi_{\text{base}}(a_2 \mid s_3) &= 1 - \gamma \\
\pi_{\text{base}}(a_1 \mid s_4) &= 1 & \pi_{\text{base}}(a_2 \mid s_4) &= 0.
\end{aligned}$$

To compute the occupancy measure of the base policy, we first can see that clearly,

$$\mu_{\pi_{\text{base}}}(s_1, a_1) = \mu_{\pi_{\text{base}}}(s_2, a_1) = \frac{1+r}{4}.$$

For s_3, the agent stays in the state until it takes action a_2, so we have

$$\begin{aligned}\mu_{\pi_{\text{base}}}(s_3) &= (1-\gamma)\mu_0(s_3)\left[1+\gamma^2+\gamma^4+\cdots\right] \\ &= (1-\gamma)\frac{(1-r)(1+\gamma)}{4}\frac{1}{1-\gamma^2} \\ &= \frac{1-r}{4}.\end{aligned}$$

This also implies that $\mu_{\pi_{\text{base}}}(s_4) = \frac{1-r}{4}$ since the occupancy measure sums to one. Thus, we can compute

$$\begin{aligned} J(\pi_{\text{base}}, R) &= 0 & J(\pi_{\text{base}}, \tilde{R}) &= 0 \\ \sigma_R &= 1 & \sigma_{\tilde{R}} &= 1 \\ \mathbb{E}_{\mu_{\pi_{\text{base}}}}[R(s,a)\tilde{R}(s,a)] &= r. \end{aligned}$$

This confirms that $\tilde{R}$ is an r-correlated proxy for R.

Now, we define $\widetilde{\pi}$ as

$$\begin{aligned} \pi_{\text{base}}(a_1 \mid s_1) &= 1 & \pi_{\text{base}}(a_2 \mid s_1) &= 0 \\ \pi_{\text{base}}(a_1 \mid s_2) &= 1 & \pi_{\text{base}}(a_2 \mid s_2) &= 0 \\ \pi_{\text{base}}(a_1 \mid s_3) &= 2\gamma - 1 & \pi_{\text{base}}(a_2 \mid s_3) &= 2(1-\gamma) \\ \pi_{\text{base}}(a_1 \mid s_4) &= 1 & \pi_{\text{base}}(a_2 \mid s_4) &= 0. \end{aligned}$$

Note that since $\gamma \geq 1/2$ by definition, the policy is well-defined. As for π_{base}, the occupancy measure of $\widetilde{\pi}$ at s_1 and s_2 is $\frac{1+r}{4}$. For s_3, we have

$$\begin{aligned}\mu_{\widetilde{\pi}}(s_3) &= (1-\gamma)\mu_0(s_3)\left[1+\gamma(2\gamma-1)+\gamma^2(2\gamma-1)^2+\cdots\right] \\ &= (1-\gamma)\frac{(1-r)(1+\gamma)}{4}\frac{1}{1-\gamma(2\gamma-1)} \\ &= \frac{(1-r)(1+\gamma)}{4(1+2\gamma)}.\end{aligned}$$

For s_4,

$$\begin{aligned}\mu_{\widetilde{\pi}}(s_4) &= 1 - \mu_{\widetilde{\pi}}(s_1) - \mu_{\widetilde{\pi}}(s_2) - \mu_{\widetilde{\pi}}(s_3) \\ &= \frac{1-r}{2} - \frac{(1-r)(1+\gamma)}{4(1+2\gamma)} \\ &= \frac{(1-r)(1+3\gamma)}{4(1+2\gamma)}.\end{aligned}$$

Based on this, we can compute the true reward of $\widetilde{\pi}$:

$$\begin{aligned} J(\widetilde{\pi}, R) &= \mathbb{E}_{\mu_{\widetilde{\pi}}}\left[R(s,a)\right] \\ &= \frac{1+r}{4} - \frac{1+r}{4} + \frac{(1-r)(1+\gamma)}{4(1+2\gamma)} - \frac{(1-r)(1+3\gamma)}{4(1+2\gamma)} \\ &= -\frac{\gamma(1-r)}{2(1+2\gamma)} \\ &\leq 0 = J(\pi_{\text{base}}, R). \end{aligned}$$

This verifies the claim that $J(\widetilde{\pi}, R) < J(\pi_{\text{base}}, R)$.

To show the second claim, we can compute the regularized objective $L'(\widetilde{\pi})$. Starting with the proxy reward term, we have

$$\begin{aligned} J(\widetilde{\pi}, \tilde{R}) &= \frac{1+r}{4} - \frac{1+r}{4} - \frac{(1-r)(1+\gamma)}{4(1+2\gamma)} + \frac{(1-r)(1+3\gamma)}{4(1+2\gamma)} \\ &= \frac{\gamma(1-r)}{2(1+2\gamma)} \\ &\geq \frac{1-r}{8}. \end{aligned} \tag{15}$$

Next, we compute the regularization term, which can be written as

$$\begin{aligned} & g\left((1-\gamma)\mathbb{E}_{\widetilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t D_f\Big(\widetilde{\pi}(\cdot \mid s_t) \parallel \pi_{\text{base}}(\cdot \mid s_t)\Big)\right]\right) \\ &= g\left(\mu(s_3) D_f\Big(\widetilde{\pi}(\cdot \mid s_3) \parallel \pi_{\text{base}}(\cdot \mid s_3)\Big)\right) \end{aligned}$$

since π_{base} and $\widetilde{\pi}$ only differ in state s_3. We can rewrite the above as

$$\begin{aligned} &= g\left(\frac{(1-r)(1+\gamma)}{4(1+2\gamma)}\left[\pi_{\text{base}}(a_1 \mid s_3) f\left(\frac{\widetilde{\pi}(a_1 \mid s_3)}{\pi_{\text{base}}(a_1 \mid s_3)}\right) + \pi_{\text{base}}(a_2 \mid s_3) f\left(\frac{\widetilde{\pi}(a_2 \mid s_3)}{\pi_{\text{base}}(a_2 \mid s_3)}\right)\right]\right) \\ &= g\left(\frac{(1-r)(1+\gamma)}{4(1+2\gamma)}\left[\gamma f\left(2-\frac{1}{\gamma}\right) + (1-\gamma) f(2)\right]\right). \end{aligned} \tag{16}$$

Note that since $\gamma \geq \frac{1}{1+\rho}$, we have

$$\begin{aligned} \frac{1}{\gamma} &\leq 1+\rho \\ 2-\frac{1}{\gamma} &\geq 1-\rho \\ f\left(2-\frac{1}{\gamma}\right) &< \frac{2g^{-1}\left(\frac{1-r}{8}\right)}{1-r}. \end{aligned}$$

Plugging this back into (16) along with the fact that $(1-\gamma)f(2) \leq \frac{2g^{-1}(\frac{1-r}{8})}{(1-r)}$, we get

$$\begin{aligned} &< g\left(\frac{(1-r)(1+\gamma)}{4(1+2\gamma)}\left[\gamma\frac{2g^{-1}\left(\frac{1-r}{8}\right)}{1-r} + \frac{2g^{-1}(\frac{1-r}{8})}{1-r}\right]\right) \\ &\leq g\left(\frac{1-r}{4} \times \frac{4g^{-1}\left(\frac{1-r}{8}\right)}{1-r}\right) \\ &\leq \frac{1-r}{8}. \end{aligned}$$

Combining this with (15), we have

$$\begin{aligned} L'(\pi) &= J(\pi, \tilde{R}) - J(\pi_{\text{base}}, \tilde{R}) - g\left((1-\gamma)\mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\gamma^t D_f\Big(\pi(\cdot \mid s_t) \parallel \pi_{\text{base}}(\cdot \mid s_t)\Big)\right]\right) \\ &> \frac{1-r}{8} - 0 - \frac{1-r}{8} = 0, \end{aligned}$$

which completes the proof. □

A.3 Action Distribution and Occupancy Measure Divergences in LLMs

As noted in the main text, in the current paradigm of using RLHF to train LLMs, we can show that action distribution divergence between two policies is equivalent to occupancy measure divergence.

In particular, RLHF for LLMs is usually modeled as a *contextual bandit*.

In our setting, a contextual bandit can be defined as an MDP with $\gamma = 0$; then, the return of the policy π under a reward function R is given by

$$J(\pi, R) = \mathbb{E}_{s\sim\mu_0(\cdot), a\sim\pi(\cdot|s)}[R(s,a)].$$

That is, a single state is sampled from the initial state distribution μ_0, and then a single action is sampled from the policy π conditioned on that state. RLHF for LLMs follows this setting as a prompt is sampled from a dataset, the LLM generates a response, and the reward is calculated based on the prompt and response.

In this setting, it is simple to show that the action distribution and occupancy measure divergences are equivalent.

Lemma A.5. *Let $D_f(P \parallel Q) = \mathbb{E}_{x\sim Q}\left[f\big(P(x)/Q(x)\big)\right]$ be the f-divergence between two distributions P and Q. Then, for any two policies π, π' in a contextual bandit, we have*

$$D_f(\mu_\pi \parallel \mu_{\pi'}) = \mathbb{E}_{s\sim\mu_0(\cdot)}\left[D_f\Big(\pi(\cdot \mid s) \parallel \pi'(\cdot \mid s)\Big)\right].$$

Lemma A.5 applies to any f-divergence, including the KL and χ^2 divergences we study in this paper.

Proof. We have

$$\begin{aligned} D_f(\mu_\pi \parallel \mu_{\pi'}) &= \mathbb{E}_{s\sim\mu_0(\cdot), a\sim\pi'(\cdot|s)}\left[f\left(\frac{\mu_\pi(s,a)}{\mu_{\pi'}(a \mid s)}\right)\right] \\ &= \mathbb{E}_{s\sim\mu_0(\cdot)}\left[\mathbb{E}_{a\sim\pi'(\cdot|s)}\left[f\left(\frac{\mu_0(s)\pi(a \mid s)}{\mu_0(s)\pi'(a \mid s)}\right)\right]\right] \\ &= \mathbb{E}_{s\sim\mu_0(\cdot)}\left[D_f(\pi \parallel \pi')\right], \end{aligned}$$

which completes the proof. □

A.3.1 Autoregressive Environments

While most RLHF implementations use the contextual bandit formulation above for the purposes of KL regularization, one can also model training an LLM as a sequential problem where each token generated is a separate action. This formulation is no longer a contextual bandit, but we can show that the action distribution and occupancy measure KL divergences are still equivalent!

Lemma A.6. *Suppose that an environment satisfies the following conditions:*

- *It is deterministic: $\mu_0(s_0) = 1$ for exactly one state s_0, and for all $s_t, a_t \in \mathcal{S} \times \mathcal{A}$, $p(s_{t+1} \mid s_t, a_t) = 1$ for exactly one state s_{t+1}.*
- *Exactly one sequence of actions leads to each state: if following $a_0, \ldots, a_{t-1}$ leads to s, then no other sequence of actions (of any length) can also lead to s.*

Then, for any policies π, π', the action distribution and occupancy measure KL divergences between them are equal (removing the $(1-\gamma)$ prefactor on the action distribution divergence):

$$D_{KL}(\mu_\pi \parallel \mu_{\pi'}) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t D_{KL}(\pi(\cdot \mid s_t) \parallel \pi'(\cdot \mid s_t))\right].$$

Lemma A.6 applies to LLMs since one can treat the "state" of the environment after t timesteps as all the tokens generated so far $w_0 w_1 \ldots w_{t-1}$, and the actions as the next token w_t, which is then appended to the state:

$$a_t \sim \pi(a_t \mid s_t) = \pi(w_t \mid w_0 w_1 \ldots w_{t-1}).$$
$$p(s_{t+1} \mid s_t, a_t) = \mathbf{1}\{s_{t+1} = w_0 w_1 \ldots w_{t-1} w_t\} \qquad \text{where} \quad s_t = w_0 w_1 \ldots w_{t-1} \quad \text{and} \quad a_t = w_t.$$

Thus, regardless of the formalism used to train an LLM via RLHF, the action distribution and occupancy measure KL are equivalent. However, the conditions of Lemma A.6 are unlikely to be met by many other MDPs. Many MDPs are stochastic, violating the first assumption. Even among deterministic MDPs, it is very uncommon that only a single action sequence can lead to each state.

Proof. Given the assumptions about the environment, we can rewrite the log-occupancy measure of a state-action pair in terms of the sum of log action probabilties over the unique sequence of actions leading to that state. Suppose $a_0, \ldots, a_{t-1}$ is the unique action sequence leading to s and that this action sequence visits states $s_0, \ldots, s_{t-1}, s$. Then

$$\begin{aligned}
\log \mu_\pi(s, a) &= \log \left((1-\gamma) \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}\{s_t = s \wedge a_t = a\} \right] \right) \\
&= \log \big((1-\gamma) \gamma^t \mathbb{P}_\pi(s_t = s \wedge a_t = a) \big) \\
&= \log \left((1-\gamma) \gamma^t \prod_{i=0}^{t} \pi(a_i \mid s_i) \right) \\
&= \log(1-\gamma) + t \log \gamma + \sum_{i=0}^{t} \log \pi(a_i \mid s_i).
\end{aligned}$$

Using this, we can rewrite the occupancy measure KL divergence as

$$\begin{aligned}
D_{\mathrm{KL}}(\mu_\pi \parallel \mu_{\pi'}) &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \mu_\pi(s, a) \log \left(\frac{\mu_\pi(s, a)}{\mu_{\pi'}(s, a)} \right) \\
&= (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \sum_{a_0, \ldots, a_t \in \mathcal{A}^{t+1}} \mathbb{P}_\pi(a_0 \wedge \cdots \wedge a_t) \sum_{i=0}^{t} \Big(\log \pi(a_i \mid s_i) - \log \pi'(a_i \mid s_i) \Big) \\
&= (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \sum_{a_0, \ldots, a_t \in \mathcal{A}^{t+1}} \left(\prod_{j=0}^{t} \pi(a_i \mid s_i) \right) \sum_{i=0}^{t} \Big(\log \pi(a_i \mid s_i) - \log \pi'(a_i \mid s_i) \Big),
\end{aligned} \tag{17}$$

where s_i is the state reached by taking $a_0, \ldots, a_{i-1}$.

We will now show inductively that

$$\begin{aligned}
&\sum_{a_0, \ldots, a_t \in \mathcal{A}^{t+1}} \left(\prod_{j=0}^{t} \pi(a_j \mid s_j) \right) \sum_{i=0}^{t} \Big(\log \pi(a_i \mid s_i) - \log \pi'(a_i \mid s_i) \Big) \\
&\quad = \sum_{i=0}^{t} \sum_{s_i \in \mathcal{S}} \mathbb{P}_\pi(s_i) D_{\mathrm{KL}}(\pi(\cdot \mid s_i) \parallel \pi'(\cdot \mid s_i)).
\end{aligned} \tag{18}$$

Consider first if $t = 0$. Then

$$\begin{aligned}
&\sum_{a_0 \in \mathcal{A}} \pi(a_0 \mid s_0) \Big(\log \pi(a_0 \mid s_0) - \log \pi'(a_0 \mid s_0) \Big) \\
&= D_{\mathrm{KL}}(\pi(\cdot \mid s_0) \parallel \pi'(\cdot \mid s_0)) \\
&= \mathbb{P}_\pi(s_0) D_{\mathrm{KL}}(\pi(\cdot \mid s_0) \parallel \pi'(\cdot \mid s_0)).
\end{aligned}$$

Now suppose (18) holds for $t-1$. Then for t we have

$$
\begin{aligned}
&\sum_{a_0,\dots,a_t \in \mathcal{A}^{t+1}} \left(\prod_{j=0}^{t} \pi(a_j \mid s_j) \right) \sum_{i=0}^{t} \Big(\log \pi(a_i \mid s_i) - \log \pi'(a_i \mid s_i) \Big) \\
&= \sum_{a_0,\dots,a_{t-1} \in \mathcal{A}^{t}} \left(\prod_{j=0}^{t-1} \pi(a_j \mid s_j) \right) \sum_{a_t \in \mathcal{A}} \pi(a_t \mid s_t) \Bigg[\log \pi(a_t \mid s_t) - \log \pi'(a_t \mid s_t) \\
&\qquad\qquad + \sum_{i=0}^{t-1} \Big(\log \pi(a_i \mid s_i) - \log \pi'(a_i \mid s_i) \Big) \Bigg] \\
&= \sum_{a_0,\dots,a_{t-1} \in \mathcal{A}^{t}} \left(\prod_{j=0}^{t-1} \pi(a_j \mid s_j) \right) \Bigg[D_{\mathrm{KL}}(\pi(\cdot \mid s_t) \parallel \pi'(\cdot \mid s_t)) \\
&\qquad\qquad + \sum_{a_t \in \mathcal{A}} \pi(a_t \mid s_t) \sum_{i=0}^{t-1} \Big(\log \pi(a_i \mid s_i) - \log \pi'(a_i \mid s_i) \Big) \Bigg] \\
&= \sum_{a_0,\dots,a_{t-1} \in \mathcal{A}^{t}} \left(\prod_{j=0}^{t-1} \pi(a_j \mid s_j) \right) \Bigg[D_{\mathrm{KL}}(\pi(\cdot \mid s_t) \parallel \pi'(\cdot \mid s_t)) \\
&\qquad\qquad + \sum_{i=0}^{t-1} \Big(\log \pi(a_i \mid s_i) - \log \pi'(a_i \mid s_i) \Big) \Bigg] \\
&= \sum_{s_t \in \mathcal{S}} \mathbb{P}_\pi(s_t) D_{\mathrm{KL}}(\pi(\cdot \mid s_t) \parallel \pi'(\cdot \mid s_t)) \\
&\qquad + \sum_{a_0,\dots,a_{t-1} \in \mathcal{A}^{t}} \left(\prod_{j=0}^{t-1} \pi(a_j \mid s_j) \right) \sum_{i=0}^{t-1} \Big(\log \pi(a_i \mid s_i) - \log \pi'(a_i \mid s_i) \Big) \\
&\overset{\text{(i)}}{=} \sum_{s_t \in \mathcal{S}} \mathbb{P}_\pi(s_t) D_{\mathrm{KL}}(\pi(\cdot \mid s_t) \parallel \pi'(\cdot \mid s_t)) + \sum_{i=0}^{t-1} \sum_{s_i \in \mathcal{S}} \mathbb{P}_\pi(s_i) D_{\mathrm{KL}}(\pi(\cdot \mid s_i) \parallel \pi'(\cdot \mid s_i)) \\
&= \sum_{i=0}^{t} \sum_{s_i \in \mathcal{S}} \mathbb{P}_\pi(s_i) D_{\mathrm{KL}}(\pi(\cdot \mid s_i) \parallel \pi'(\cdot \mid s_i)),
\end{aligned}
$$

where (i) is from the inductive hypothesis.

Now, plugging (18) into (17) gives

$$
\begin{aligned}
&D_{\mathrm{KL}}(\mu_\pi \parallel \mu_{\pi'}) \\
&= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\sum_{i=0}^{t}\sum_{s_i\in\mathcal{S}}\mathbb{P}_\pi(s_i)D_{\mathrm{KL}}(\pi(\cdot \mid s_i) \parallel \pi'(\cdot \mid s_i)) \\
&= (1-\gamma)\sum_{i=0}^{\infty}\sum_{t=i}^{\infty}\gamma^t\sum_{s_i\in\mathcal{S}}\mathbb{P}_\pi(s_i)D_{\mathrm{KL}}(\pi(\cdot \mid s_i) \parallel \pi'(\cdot \mid s_i)) \\
&= (1-\gamma)\sum_{i=0}^{\infty}\sum_{s_i\in\mathcal{S}}\mathbb{P}_\pi(s_i)D_{\mathrm{KL}}(\pi(\cdot \mid s_i) \parallel \pi'(\cdot \mid s_i))\sum_{t=i}^{\infty}\gamma^t \\
&= (1-\gamma)\mathbb{E}_\pi\left[\sum_{i=0}^{\infty}D_{\mathrm{KL}}(\pi(\cdot \mid s_i) \parallel \pi'(\cdot \mid s_i))\sum_{t=i}^{\infty}\gamma^t\right] \\
&= (1-\gamma)\mathbb{E}_\pi\left[\frac{\gamma^i}{1-\gamma}\sum_{i=0}^{\infty}D_{\mathrm{KL}}(\pi(\cdot \mid s_i) \parallel \pi'(\cdot \mid s_i))\right] \\
&= \mathbb{E}_\pi\left[\gamma^i\sum_{i=0}^{\infty}D_{\mathrm{KL}}(\pi(\cdot \mid s_i) \parallel \pi'(\cdot \mid s_i))\right],
\end{aligned}
$$

which is the desired result. □

A.4 Learned Reward Functions are r-Correlated

Proxy reward functions are often *learned* from data like ratings or preference comparisons, including in the case of RLHF. Here, we show that a learned reward function with low mean-squared error—a common objective in supervised learning—is r-correlated with the true reward function.

Lemma A.7. *Let R be the true reward function and $\tilde{R}$ be a learned reward function. Suppose that $\mathbb{E}_{\mu_{\pi_{base}}}\left[(R(s,a)-\tilde{R}(s,a))^2\right] \leq \epsilon\sigma_R^2$. Then, the learned reward function is an r-correlated proxy with $r \geq 1-\epsilon$.*

The assumption in Lemma A.7 is that the mean-squared error over the occupancy measure of the base policy is small. This can be achieved, for example, by learning $\tilde{R}$ via least-squares regression over a training dataset of state-action pairs sampled from the base policy. Many results in learning theory show that this results in a small mean-squared error over the distribution the training data was sampled from, i.e., exactly the assumption in Lemma A.7 (Koltchinskii, 2006).

Proof. Throughout the proof, all expectations, variances, and covariances are with respect to $\mu_{\pi_{\text{base}}}$. We can rewrite the assumption using the bias-variance decomposition as

$$
\begin{aligned}
&\mathbb{E}\left[(R(s,a)-\tilde{R}(s,a))^2\right] \\
&\quad= \mathrm{Var}\left[R(s,a)-\tilde{R}(s,a)\right] + \left(\mathbb{E}\left[R(s,a)\right] - \mathbb{E}\left[\tilde{R}(s,a)\right]\right)^2 \\
&\quad= \mathrm{Var}\left[R(s,a)\right] + \mathrm{Var}\left[\tilde{R}(s,a)\right] - 2\,\mathrm{Cov}\left[R(s,a),\tilde{R}(s,a)\right] + \left(\mathbb{E}\left[R(s,a)\right] - \mathbb{E}\left[\tilde{R}(s,a)\right]\right)^2 \\
&\quad= \sigma_R^2 + \sigma_{\tilde{R}}^2 - 2\,\mathrm{Cov}\left[R(s,a),\tilde{R}(s,a)\right] + \left(\mathbb{E}\left[R(s,a)\right] - \mathbb{E}\left[\tilde{R}(s,a)\right]\right)^2 \\
&\quad\leq \epsilon\sigma_R^2.
\end{aligned}
$$

Note that $\left(\mathbb{E}\left[R(s,a)\right] - \mathbb{E}\left[\tilde{R}(s,a)\right]\right)^2 > 0$, so we can rewrite the inequality as

$$
2\mathrm{Cov}\left[R(s,a),\tilde{R}(s,a)\right] \geq (1-\epsilon)\sigma_R^2 + \sigma_{\tilde{R}}^2 \geq (1-\epsilon)\left(\sigma_R^2 + \sigma_{\tilde{R}}^2\right).
$$

Dividing both sides by $2\sigma_R\sigma_{\tilde{R}}$ gives

$$\frac{\mathrm{Cov}\left[R(s,a),\tilde{R}(s,a)\right]}{\sigma_R\sigma_{\tilde{R}}} \geq \frac{1-\epsilon}{2}\frac{\sigma_R^2+\sigma_{\tilde{R}}^2}{\sigma_R\sigma_{\tilde{R}}}.$$

By the AM-GM inequality, $\frac{\sigma_R^2+\sigma_{\tilde{R}}^2}{2} \geq \sigma_R\sigma_{\tilde{R}}$, so

$$\frac{\mathrm{Cov}\left[R(s,a),\tilde{R}(s,a)\right]}{\sigma_R\sigma_{\tilde{R}}} \geq 1-\epsilon,$$

which is the desired result. □

A.5 State-Only Occupancy Measures

We can also consider state-only occupancy measures, defined as

$$\mu_\pi(s) = (1-\gamma)\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t \mathbb{1}\{s_t = s\}\right].$$

In many environments, the reward functions only depend on the state, i.e., $R(s,a) = R(s)$ and $\tilde{R}(s,a) = \tilde{R}(s)$. In this case, Theorem 5.1 holds for state-only occupancy measures as well. The proof is identical to the proof of Theorem 5.1, but replacing expectations and sums over state-action pairs with expectations over states.

B Derivation of Occupancy-Regularized Policy Optimization

In this appendix section, we show how to derive the approximations used for Occupancy-Regularized Policy Optimization (ORPO). As a reminder, we would like to optimize

$$J(\pi_\theta,\tilde{R}) - \lambda\sqrt{\chi^2\left(\mu_{\pi_\theta}\|\mu_{\pi_{\text{base}}}\right)}.$$

We can rewrite its gradient as

$$\begin{aligned}
&\nabla_\theta\left(J(\pi_\theta,\tilde{R}) - \lambda\sqrt{\chi^2\left(\mu_{\pi_\theta}\|\mu_{\pi_{\text{base}}}\right)}\right)\\
&\quad= \nabla_\theta J(\pi_\theta,\tilde{R}) - \frac{\lambda\nabla_\theta\chi^2\left(\mu_{\pi_\theta}\|\mu_{\pi_{\text{base}}}\right)}{2\sqrt{\chi^2\left(\mu_{\pi_\theta}\|\mu_{\pi_{\text{base}}}\right)}}\\
&\quad= \nabla_\theta\left(\sum_{s,a}\mu_{\pi_\theta}(s,a)\tilde{R}(s,a)\right) - \frac{\lambda}{2\sqrt{\chi^2\left(\mu_{\pi_\theta}\|\mu_{\pi_{\text{base}}}\right)}}\nabla_\theta\left(\sum_{s,a}\frac{\mu_{\pi_\theta}(s,a)^2}{\mu_{\pi_{\text{base}}}(s,a)} - 1\right)\\
&\quad= \sum_{s,a}\left[\nabla_\theta\mu_{\pi_\theta}(s,a)\tilde{R}(s,a) - \frac{\lambda}{2\sqrt{\chi^2\left(\mu_{\pi_\theta}\|\mu_{\pi_{\text{base}}}\right)}}\nabla_\theta\left(\frac{\mu_{\pi_\theta}(s,a)^2}{\mu_{\pi_{\text{base}}}(s,a)} - 1\right)\right]\\
&\quad= \sum_{s,a}\left[\nabla_\theta\mu_{\pi_\theta}(s,a)\tilde{R}(s,a) - \frac{\lambda}{2\sqrt{\chi^2\left(\mu_{\pi_\theta}\|\mu_{\pi_{\text{base}}}\right)}}\frac{2\mu_{\pi_\theta}(s,a)}{\mu_{\pi_{\text{base}}}(s,a)}\nabla_\theta\mu_{\pi_\theta}(s,a)\right]\\
&\quad= \sum_{s,a}\left(\nabla_\theta\mu_{\pi_\theta}(s,a)\right)\left(\tilde{R}(s,a) - \frac{\lambda}{\sqrt{\chi^2(\mu_{\pi_\theta}\|\mu_{\pi_{\text{base}}})}}\frac{\mu_{\pi_\theta}(s,a)}{\mu_{\pi_{\text{base}}}(s,a)}\right).
\end{aligned}$$

As described in the main text, policy gradient algorithms can approximate this type of gradient by using an augmented reward function

$$R'(s,a) = \tilde{R}(s,a) - \frac{\lambda}{\sqrt{\chi^2(\mu_{\pi_\theta}\|\mu_{\pi_{\text{base}}})}}\frac{\mu_{\pi_\theta}(s,a)}{\mu_{\pi_{\text{base}}}(s,a)}. \tag{19}$$

However, two terms in (19) cannot be computed directly: the current χ^2 divergence between occupancy measures, and the ratio of the occupancy measures $\mu_{\pi_\theta}(s,a)/\mu_{\pi_{\text{base}}}(s,a)$. We first show how to approximate the latter using a discriminator network $\hat{d}_\phi(s,a)$, trained to optimize

$$\phi = \arg\min_{\phi} \; \mathbb{E}_{\mu_{\pi_\theta}}\Big[\, \log(1+e^{-\hat{d}_\phi(s,a)})\Big] + \mathbb{E}_{\mu_{\pi_{\text{base}}}}\Big[\, \log(1+e^{\hat{d}_\phi(s,a)})\Big]. \tag{20}$$

It is well-known that exactly optimizing the loss function in (20) gives

$$\hat{d}_\phi(s,a) = \log \frac{\mu_{\pi_\theta}(s,a)}{\mu_{\pi_{\text{base}}}(s,a)}. \tag{21}$$

Furthermore, the OM χ^2 divergence is given by

$$\chi^2\left(\mu_{\pi_\theta} \| \mu_{\pi_{\text{base}}}\right) = \mathbb{E}_{\mu_{\pi_\theta}}\left[\frac{\mu_{\pi_\theta}(s,a)}{\mu_{\pi_{\text{base}}}(s,a)} - 1\right] = \mathbb{E}_{\mu_{\pi_\theta}}\left[e^{\hat{d}_\phi(s,a)} - 1\right] =: \widehat{\chi^2}. \tag{22}$$

Combining (21) and (22) shows that the augmented reward in (19) can be rewritten as

$$R'(s,a) = \tilde{R}(s,a) - \frac{\lambda}{\sqrt{\widehat{\chi^2}}} e^{\hat{d}_\phi(s,a)}.$$

Putting all the steps together, the following algorithm formalizes ORPO:

Algorithm 1 Occupancy-Regularized Policy Optimization (ORPO).

1: **for** iteration $i = 1, \ldots, I$ **do**
2: Collect a set of n trajectories $\mathcal{D}_\pi$ from π_θ.
3: Collect a set of n trajectories $\mathcal{D}_{\pi_{\text{base}}}$ from π_{base}.
4: Optimize ϕ via SGD to minimize

$$\mathbb{E}_{\mathcal{D}_\pi}\Big[\, \log(1+e^{-\hat{d}_\phi(s,a)})\Big] + \mathbb{E}_{\mathcal{D}_{\pi_{\text{base}}}}\Big[\, \log(1+e^{\hat{d}_\phi(s,a)})\Big]$$

5: Calculate $\widehat{\chi^2} = \mathbb{E}_{\mathcal{D}_\pi}\left[e^{\hat{d}_\phi(s,a)} - 1\right]$.
6: Transform $\mathcal{D}_\pi$ to $\mathcal{D}'_\pi$ by replacing the rewards with $R'(s,a) = \tilde{R}(s,a) - \frac{\lambda}{\sqrt{\widehat{\chi^2}}}\left(e^{\hat{d}_\phi(s,a)} - 1\right)$.
7: Optimize θ via SGD to minimize the proximal policy optimization (PPO) loss $L_{\text{PPO}}\left(\mathcal{D}'_\pi\right)$.
8: **end for**

A similar approach can also be used to optimize the proxy reward regularized by KL divergence

$$J(\pi_\theta, \tilde{R}) - \lambda\, D_{\text{KL}}\left(\mu_{\pi_\theta} \| \mu_{\pi_{\text{base}}}\right),$$

by changing the augmented reward in Line 6 of Algorithm 1 to

$$R'(s,a) = \tilde{R}(s,a) - \lambda\, \hat{d}_\phi(s,a).$$

How accurate is the discriminator-based approximation?. To determine whether using the discriminator results in an accurate approximation of χ^2 and KL OM divergences, we plot the output of the discriminator in the glucose environment versus the theoretically correct value in Figure 6. The results suggest that the discriminator accurately approximates the log ratio of the occupancy measures, which in turn allows for accurate approximations of the OM divergences.

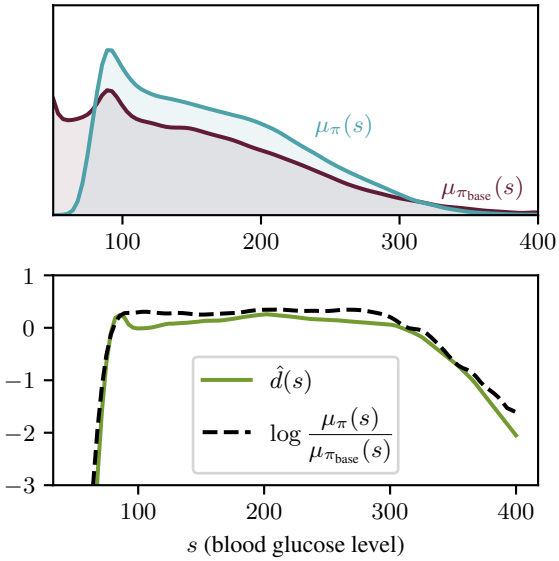


Figure 6: The top plot shows the state occupancy measures of the base policy π_{base} and a policy π optimized with ORPO in the glucose environment. The bottom plot shows the exact log ratio of the occupancy measures $\log \mu_\pi(s)/\mu_{\pi_{\text{base}}}(s)$ versus the discriminator output $\hat{d}_\phi(s)$, which attempts to approximate it. We find the discriminator output to be a good approximation of the log ratio.

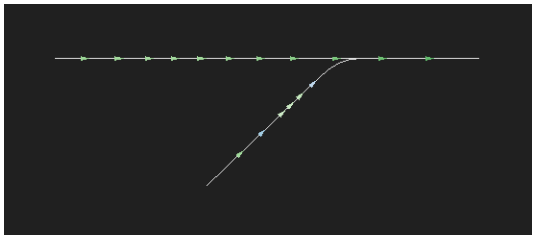
C Environment Details

Here, we discuss the details of the five reward-hacking environments we study.

C.1 Traffic Control

The traffic control environment, based on the Flow simulator (Wu et al., 2022), simulates a group of human-controlled and RL-controlled vehicles on an on-ramp attempting to merge into traffic on a highway. The true reward prioritizes a small mean commute time, while the proxy reward is the average velocity of all cars. When reward hacking, the RL controlled vehicle on the on-ramp stops indefinitely and lets cars continue forward at high speeds on the highway, which maximizes the proxy reward but increases the commute times of cars on the on-ramp infinitely. As the base policy for the traffic environment we used the Intelligent Driver Model (IDM), a standard approximation of human driving behavior (Treiber et al., 2000). In practice, base policies are often learned via imitation learning, so to simulate this we generate data from the IDM controller and train a behavioral cloning (BC) policy using the generated data.

Here, the green cars are controlled by the human driver model IDM controller, and the blue cars are controlled by RL:

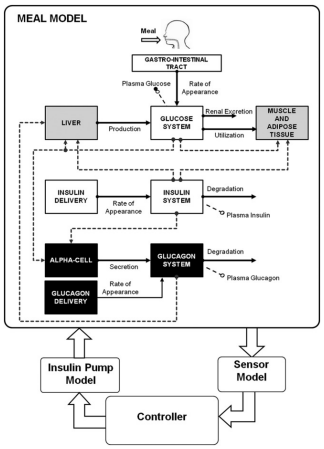


This particular frame showcases reward hacking behavior. The blue RL-controlled vehicle has stopped completely on the on-ramp, blocking cars behind it. This increases the average velocity of all vehicles in the simulation, as the cars on the straightway are able to continue speeding along the road without having to wait for merging cars. However, the true reward (negative average commute

time) decreases endlessly as the cars on the on-ramp wait.

C.2 GLUCOSE MONITORING

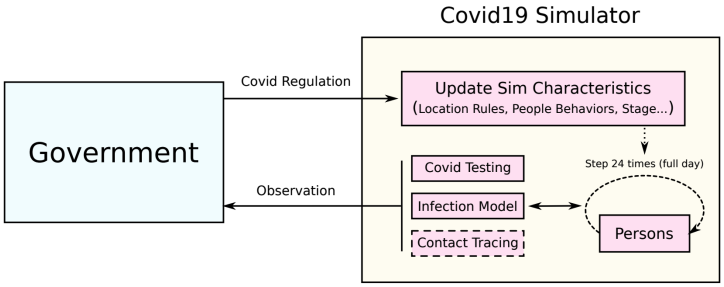
The SimGlucose blood glucose monitoring environment is an extension of the FDA-approved glucose monitoring simulator proposed by Man et al. (2014) for Type 1 Diabetes patients (Fox et al., 2020):



The RL agent (bottom) controls the insulin administered to a simulated patient in order to maintain healthy glucose levels. The true reward is a standard measure of health risk for the patient, but the proxy reward is misaligned and prioritizes the monetary cost of insulin. As the safe baseline policy, we train a BC policy based on data generated by a PID controller with parameters tuned by the original designers of the simulator (Steil, 2013).

C.3 PANDEMIC MITIGATION

PandemicSimulator (Kompella et al., 2020) simulates a population's infection dynamics using a COVID-specific SEIR model (Mwalili et al., 2020):



The RL agent chooses the level of lockdown restrictions placed on the population by observing the results of testing. The proxy reward function omits the political cost associated with certain decisions. Our base policy is trained via BC on a combination of hand-specified and real-world strategies employed by governments during the pandemic, which were also used by Kompella et al. (2020) as baselines.

C.4 RLHF

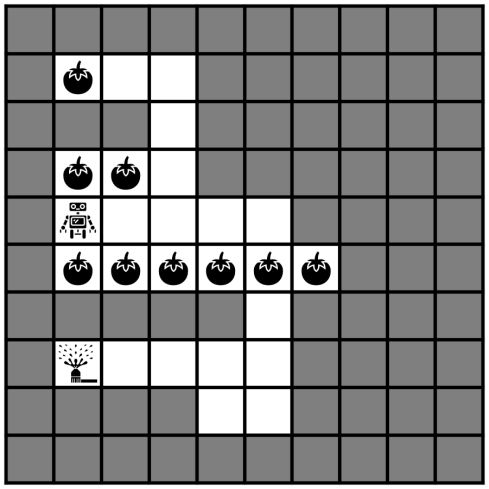
We base our RLHF environment on the work of Coste et al. (2024), who study overoptimization of LLM-based reward models. The proxy reward model we use is fine-tuned from Pythia-70M (Biderman et al., 2023) on the AlpacaFarm (Dubois et al., 2023) preference dataset. As a true reward model, we use the Llama 3 Tulu V2 8B RM from AI2 (Ivison et al., 2024). As a base policy, we use

Coste et al.'s SFT policy, which was fine-tuned from Pythia-1.4B on the AlpacaFarm SFT data. We evaluate policies' true and proxy rewards on responses sampled for a held out set of prompts.

C.5 Tomato-Watering Gridworld

The tomato environment contains a sprinkler state where the agent perceives all tomatoes as being watered and thus receives high proxy reward but no true reward. We train a base policy using the true reward function, and then add a 10% chance of taking a random action to ensure there is room to improve upon it.

The gray squares in the environment represent walls, and the white squares represent open spaces where the agent can travel:



The sprinkler state is down a narrow hallway, and on the other end a tomato is down another narrow hallway.

D Experiment Details

D.1 Non-LLM Experiments

We implement ORPO using RLLib (Liang et al., 2018) and PyTorch (Paszke et al., 2019).

Network architectures For the pandemic, traffic, and tomato-watering environments, we use policy networks based on fully-connected networks with 2 layers of 128 units, 4 layers of 512 units, and 4 units of 512 units, respectively. The policy model for the glucose environment is a basic LSTM network with 3 layers of 64 units each. We made this choice since the observation of the environment contains continuous historical information about the patient's blood glucose levels and previously administered insulin.

The discriminator model for all four non-LLM environments is a fully connected network with 2 layers of 256 units each. We found that the discriminator architecture did not need to be tuned to each environment.

Policy initialization Initializing using an imitation learning policy has been shown to effectively speed up the learning process (Laidlaw et al., 2023; Uchendu et al., 2023) and is used in practice for RLHF (Stiennon et al., 2020), so we initialize our policies using the specified π_{base} for the more realistic traffic, glucose, and pandemic environments.

Hyperparameters Some hyperparameters for the traffic environment were tuned by Pan et al. (2022). We primarily tuned the hyperparameters listed below in order to ensure that the proxy reward

would be properly optimized and reward hacking would occur without regularization. This enables to see if the various regularization methods actually succeed at preventing reward hacking.

Hyperparameter	**Tomato**	**Traffic**	**Glucose**	**Pandemic**
Training iterations	500	250	500	260
Batch size	3000	40000	100000	3860
SGD minibatch size	128	16384	1024	64
SGD epochs per iteration	8	5	4	5
Optimizer	Adam	Adam	Adam	Adam
Learning rate	1e-3	5e-5	1e-4	0.0003
Gradient clipping	0.1	None	10	10
Discount rate (γ)	0.99	0.99	0.99	0.99
GAE coefficient (λ)	0.98	0.97	0.98	0.95
Entropy coefficient (start)	0.01	0.01	0.01	0.1
Entropy coefficient (end)	0.01	0.01	0.01	0.01
Entropy schedule horizon	0	0	0	500000
KL target	0.001	0.02	1e-3	0.01
Value function loss clipping	10	10,000	100	20
Value function loss coefficient	0.1	0.5	0.0001	0.5
Share value function layers	F	T	T	T

Table 2: PPO/ORPO hyperparameters.

Hyperparameter	**Tomato**	**Traffic**	**Glucose**	**Pandemic**
Discriminator reward clipping	1000	10	1e10	0.1
Regularization coefficient (λ)	Varied	Varied	Varied	Varied
$\sigma_{\tilde{R}}$	0.05	0.0002	0.05	0.08

Table 3: ORPO-specific hyperparameters.

ORPO details We found that a couple of tricks were useful to ensure that ORPO remained stable. First, we clip the discriminator term added to the reward functions to a range $[-\delta, \delta]$, since sometimes it can blow up and cause numerical issues. Second, when estimating $\widehat{\chi^2}$, we use a trimmed mean (trimmed by 1% in each tail) to reduce the effect that outliers have on the estimate. These are both particularly important for χ^2 divergence, where the output of the discriminator is exponentiated for both the reward discriminator term and for estimating $\widehat{\chi^2}$.

D.2 RLHF Experiments

We train LLMs via the RLHF implementation used by Coste et al. (2024), which is based on OpenAssistant and trlX (Havrilla et al., 2023). To implement χ^2 or KL regularization, we directly add a loss term to the PPO loss:

$$\chi^2 \text{ divergence:} \quad \lambda\left(\frac{\pi_\theta(a \mid s)}{\pi_{\text{base}}(a \mid s)} + \frac{\pi_{\text{base}}(a \mid s)}{\pi_\theta(a \mid s)} - 2\right)$$

$$\text{KL divergence:} \quad \lambda\left(\log\frac{\pi_\theta(a \mid s)}{\pi_{\text{base}}(a \mid s)} + \frac{\pi_{\text{base}}(a \mid s)}{\pi_\theta(a \mid s)} - 1\right)$$

where s is the prompt and a is the sampled response. Intuitively, both loss terms have a unique minimum when $\pi_\theta(a \mid s) = \pi_{\text{base}}(a \mid s)$ and in expectation are equivalent to the correct divergence. Schulman (2020) suggests that these are particularly low-variance estimates, and we find that they work well in practice.

E Additional Experiments and Results

In this appendix, we the full results from our main experiments as well as ablations of ORPO.

E.1 Win Rates for RLHF

Besides calculating the true reward for RLHF models with the Llama 3 Tulu V2 8B RM, we also calculate the win rates for the best coefficient of KL and χ^2 regularization. We use AlpacaEval (Dubois et al., 2024) with GPT4o-mini to compute the win rate between the RLHF policy and the SFT policy. We find that the median win rate for χ^2 divergence is higher, and it is also more consistent across random seeds. In contrast, using KL divergence leads to reward hacking for one seed and a lower median win rate.

Divergence	Coefficient	Median win rate	Win rate range
χ^2	0.0008	**52.83**	51.98 – 53.98
KL	0.025	51.50	11.93 – 53.75

Table 4: Win rates for RLHF-trained models using KL divergence vs. χ^2 divergence. The median win rate and range of win rates are reported across five seeds.

E.2 Results for All Regularization Coefficients

Here, we present the results of training with AD and OM regularization using χ^2 and KL divergence across all regularization coefficients. Each table shows the median and the standard deviation of the true rewards achieved by the learned policy across 5 random seeds.

χ^2 divergence

Coefficient	AD	State OM	State-action OM
0.000002	-60.79 $\pm$ 8.40	-61.04 $\pm$ 2.20	-59.11 $\pm$ 4.53
0.000004	-62.01 $\pm$ 8.58	-61.85 $\pm$ 2.22	-58.21 $\pm$ 5.52
0.00001	-60.43 $\pm$ 6.61	-59.82 $\pm$ 2.85	-54.35 $\pm$ 2.81
0.00002	-62.01 $\pm$ 4.54	-56.53 $\pm$ 10.66	-1.35 $\pm$ 30.28
0.00004	-50.84 $\pm$ 6.28	-42.24 $\pm$ 22.01	-1.15 $\pm$ 0.06
0.0001	-1.29 $\pm$ 0.12	-2.18 $\pm$ 0.42	-1.38 $\pm$ 0.29
0.0002	-1.70 $\pm$ 0.12	-2.46 $\pm$ 0.65	-1.85 $\pm$ 0.29

KL divergence

Coefficient	AD	State OM	State-action OM
0.000001	-56.20 $\pm$ 4.02	-61.59 $\pm$ 2.72	-61.18 $\pm$ 2.87
0.0000025	-59.58 $\pm$ 4.84	-54.59 $\pm$ 3.16	-59.59 $\pm$ 2.36
0.000005	-57.24 $\pm$ 2.62	-59.03 $\pm$ 5.41	-61.24 $\pm$ 2.01
0.00001	-54.84 $\pm$ 3.15	-57.62 $\pm$ 3.13	-58.85 $\pm$ 2.73
0.000025	-55.10 $\pm$ 2.64	-59.32 $\pm$ 1.37	-56.86 $\pm$ 5.48
0.00005	-49.99 $\pm$ 4.04	-59.96 $\pm$ 1.65	-53.39 $\pm$ 23.77
0.0001	-45.72 $\pm$ 9.02	-1.34 $\pm$ 25.30	-1.25 $\pm$ 0.07
0.00025	-1.33 $\pm$ 0.05	-1.47 $\pm$ 0.20	-1.51 $\pm$ 0.10
0.0005	-1.52 $\pm$ 0.04	-1.76 $\pm$ 0.22	-1.99 $\pm$ 0.35
0.001	-1.73 $\pm$ 0.07	-1.76 $\pm$ 0.26	-2.30 $\pm$ 1.14
0.0025	-1.98 $\pm$ 0.07	-1.76 $\pm$ 0.51	-1.94 $\pm$ 0.30
0.005	-2.15 $\pm$ 0.05	-1.90 $\pm$ 0.83	-2.08 $\pm$ 0.61
0.01	-2.11 $\pm$ 0.05	-2.12 $\pm$ 1.00	-2.14 $\pm$ 0.56

Table 5: All traffic control results ($\times 10^3$).

χ^2 divergence

Coefficient	**AD**	**State OM**	**State-action OM**
0.0008	-17.60 ± 1.78	-12.59 ± 1.63	-27.16 ± 12.83
0.0016	-36.16 ± 3.25	-11.25 ± 10.08	-11.17 ± 0.19
0.004	-34.16 ± 12.90	-10.88 ± 0.91	-11.65 ± 2.12
0.008	-12.45 ± 0.25	-10.99 ± 3.13	-12.02 ± 0.18
0.016	-12.31 ± 0.08	-10.68 ± 0.17	-12.18 ± 0.46
0.04	-12.29 ± 0.05	-10.78 ± 0.12	-12.34 ± 0.10
0.08	-12.39 ± 0.04	-10.73 ± 0.84	-12.20 ± 0.11

KL divergence

Coefficient	**AD**	**State OM**	**State-action OM**
0.00006	-21.23 ± 9.74	-33.59 ± 9.89	-30.96 ± 22.42
0.00012	-30.39 ± 22.16	-41.96 ± 12.70	-19.67 ± 6.43
0.0003	-23.10 ± 5.63	-35.29 ± 10.32	-27.56 ± 7.50
0.0006	-21.85 ± 19.03	-34.56 ± 11.97	-22.40 ± 9.84
0.0012	-25.17 ± 10.24	-31.28 ± 7.83	-31.77 ± 6.01
0.003	-23.51 ± 6.91	-35.76 ± 10.53	-23.90 ± 12.81
0.006	-12.26 ± 11.82	-58.08 ± 46.98	-29.42 ± 25.37
0.012	-12.30 ± 9.27	-10.60 ± 0.87	-11.88 ± 0.81
0.03	-12.28 ± 0.14	-11.03 ± 6.86	-11.73 ± 0.21
0.06	-12.20 ± 0.07	-10.71 ± 0.18	-12.23 ± 14.25
0.12	-12.33 ± 0.04	-10.24 ± 0.61	-12.09 ± 0.38
0.3	-12.35 ± 0.04	-11.02 ± 0.57	-12.11 ± 0.25
0.6	-12.40 ± 0.04	-10.61 ± 0.36	-12.11 ± 0.28
1.2	-12.33 ± 0.03	-10.50 ± 0.26	-12.02 ± 0.28

Table 6: All pandemic mitigation results.

χ^2 divergence

Coefficient	**AD**	**State OM**	**State-action OM**
0.0005	-580.7 ± 73.8	-484.2 ± 56.6	-164.0 ± 3.5
0.001	-94.2 ± 14.0	-263.9 ± 19.2	-127.0 ± 5.9
0.0025	-97.1 ± 8.2	-146.4 ± 18.1	-93.3 ± 9.3
0.005	-76.6 ± 10.5	-109.7 ± 10.3	-55.2 ± 0.7
0.01	-84.7 ± 8.5	-72.8 ± 8.8	-47.5 ± 0.6
0.025	-85.6 ± 7.9	-54.3 ± 1.3	-50.9 ± 2.3
0.05	-74.8 ± 13.1	-57.1 ± 3.5	-113.3 ± 32.8

KL divergence

Coefficient	**AD**	**State OM**	**State-action OM**
0.00003	-598.4 ± 39.7	-604.1 ± 10.7	-583.8 ± 61.2
0.00006	-600.6 ± 12.4	-589.0 ± 260.9	-594.7 ± 3.3
0.00015	-592.3 ± 51.3	-607.9 ± 11.1	-577.1 ± 11.2
0.0003	-593.6 ± 6.0	-592.0 ± 29.3	-497.9 ± 11.2
0.0006	-590.0 ± 7.5	-593.1 ± 5.4	-364.3 ± 5.8
0.0015	-459.9 ± 114.1	-511.1 ± 21.1	-181.6 ± 7.5
0.003	-270.0 ± 39.7	-332.3 ± 41.0	-101.2 ± 5.0
0.006	-154.5 ± 5.5	-158.7 ± 28.8	-61.9 ± 5.2
0.015	-84.1 ± 6.8	-82.9 ± 5.6	-48.9 ± 0.5
0.03	-98.3 ± 8.4	-58.4 ± 3.8	-49.6 ± 1.2
0.06	-88.6 ± 12.8	-59.0 ± 7.2	-78.3 ± 10.2
0.15	-82.1 ± 11.6	-75.9 ± 4.6	-106.6 ± 19.6
0.3	-73.4 ± 9.2	-98.1 ± 16.1	-127.3 ± 24.7
0.6	-88.6 ± 5.6	-112.5 ± 16.6	-118.5 ± 10.7

Table 7: All glucose monitoring results ($\times 10^3$).

Coefficient	**AD χ^2**	**AD KL**
0.00008	9.20 ± 0.68	8.80 ± 2.24
0.00025	14.05 ± 3.27	9.48 ± 1.02
0.0008	16.94 ± 0.07	8.84 ± 0.42
0.0025	16.84 ± 0.08	14.22 ± 2.81
0.008	16.71 ± 0.11	12.73 ± 2.75
0.025	16.59 ± 0.11	16.81 ± 0.27
0.08	16.43 ± 0.07	16.52 ± 0.08
0.25	16.22 ± 0.10	16.33 ± 0.04
0.8	16.13 ± 0.10	16.25 ± 0.13

Table 8: All results for RLHF.

χ^2 divergence

Coefficient	AD	State OM	State-action OM
0.0005	2.65 ± 0.67	0.64 ± 0.19	0.53 ± 0.29
0.001	3.87 ± 0.25	0.65 ± 4.63	0.60 ± 0.16
0.0025	6.24 ± 0.10	9.06 ± 0.17	9.04 ± 4.71
0.005	6.17 ± 0.03	9.06 ± 0.11	9.16 ± 0.09
0.01	6.16 ± 0.04	9.07 ± 0.07	9.17 ± 0.12
0.025	6.19 ± 0.04	8.64 ± 0.12	8.61 ± 0.18
0.05	6.14 ± 0.00	7.95 ± 0.05	7.99 ± 0.22

KL divergence

Coefficient	AD	State OM	State-action OM
0.0008	2.52 ± 0.18	2.32 ± 0.86	2.31 ± 0.08
0.0016	2.98 ± 0.33	2.31 ± 0.07	1.11 ± 0.89
0.004	4.59 ± 0.19	2.23 ± 0.06	2.01 ± 0.23
0.008	6.10 ± 0.15	1.30 ± 0.19	1.25 ± 0.32
0.016	6.33 ± 0.13	0.82 ± 0.22	0.84 ± 0.21
0.04	6.26 ± 0.05	1.17 ± 2.92	1.81 ± 0.31
0.08	6.26 ± 0.04	7.62 ± 0.06	7.32 ± 0.28
0.16	6.21 ± 0.05	7.20 ± 0.12	7.12 ± 0.11
0.4	6.16 ± 0.03	6.89 ± 0.14	6.84 ± 0.19
0.8	6.19 ± 0.03	7.07 ± 0.12	6.86 ± 0.19
1.6	6.14 ± 0.04	6.90 ± 0.13	6.61 ± 0.31
4	6.13 ± 0.03	6.80 ± 0.16	6.79 ± 0.12
8	6.13 ± 0.01	6.81 ± 0.28	6.80 ± 0.06
16	6.13 ± 0.00	6.94 ± 0.10	6.83 ± 0.25

Table 9: All results for the tomato-watering gridworld.

E.3 Ablations

Here, we present the results of two ablations of ORPO. For each of the non-RLHF environments, we fix the optimal coefficient for state-action OM χ^2 regularization and modify other hyperparameters. We do not ablate the RLHF experiments because RLHF is a contextual bandit (Appendix A.3) and so it isn't actually necessary to run ORPO for RLHF. The results are shown in Table 10 below.

Order of training policy and discriminator networks. First, we experiment with modifying ORPO to train the discriminator *after* the policy. In Algorithm 1, the discriminator $\hat{d}_\phi$ is optimized, then the rewards are updated with the discriminator outputs, and then the policy is trained with the updated rewards. An alternative is to wait to train the discriminator until after the policy has been updated, i.e., put Line 4 after Line 7. We experimented with this and found that in in most environments there is not too much difference between the two orders, although training the discriminator first gives slightly better results. However, in the pandemic mitigation environment, we found that it training the discriminator second gave results with much higher variance. This suggests that it is best to train the discriminator before augmenting the rewards to train the policy.

Discriminator reward clipping. Second, we experiment with modifying the discriminator reward clipping parameter δ of ORPO. We found that removing the clipping parameter entirely led to NaN errors and training could not complete, so we do not report those results. However, to test sensitivity to this parameter, we tried training with a clipping parameter $10\times$ larger and $10\times$ smaller in each environment. We found that the results did not vary by much across different clipping parameters. This suggests that ORPO is relatively robust to the hyperparameter, so it does not need to be tuned precisely.

Method	**Environment**			
	Traffic control ($\times 10^3$)	Pandemic mitigation	Glucose monitoring ($\times 10^3$)	AI safety gridworld
Default parameters	-1.15 $\pm$ 0.06	-12.18 $\pm$ 0.46	-47.5 $\pm$ 0.6	9.17 $\pm$ 0.12
Train policy before discriminator	-1.24 $\pm$ 0.07	-12.23 $\pm$ 2.71	-48.2 $\pm$ 0.7	8.97 $\pm$ 0.13
Discriminator reward clipping $\times 0.1$	-1.18 $\pm$ 0.14	-12.22 $\pm$ 0.21	-47.7 $\pm$ 0.8	9.21 $\pm$ 0.18
Discriminator reward clipping $\times 10$	-1.24 $\pm$ 0.03	-12.11 $\pm$ 0.29	-47.9 $\pm$ 0.7	9.20 $\pm$ 0.14

Table 10: Results of our ablations of ORPO. We report the median and standard deviation of the true reward across five random seeds for the four non-RLHF environments. The top row shows the results of state-action occupancy measure regularization with the optimal coefficient for χ^2 divergence. The other rows show ablations with the same coefficient. We find that ORPO is mostly robust to different hyperparameters but that it is probably best to train the discriminator network before the policy network.