Robust Optimization for Mitigating Reward Hacking with Correlated Proxies

Anonymous Author(s)
Affiliation
Address
email

Abstract

Designing robust reinforcement learning (RL) agents in the presence of imperfect reward signals remains a core challenge. In practice, agents are often trained with proxy rewards that only approximate the true objective, leaving them vulnerable to reward hacking, where high proxy returns arise from unintended or exploitative behaviors. Recent work formalizes this issue using r-correlation between proxy and true rewards, but existing methods like occupancy-regularized policy optimization (ORPO) optimize against a fixed proxy and do not provide strong guarantees against broader classes of correlated proxies. In this work, we formulate reward hacking as a robust policy optimization problem over the space of all r-correlated proxy rewards. We derive a tractable max-min formulation, where the agent maximizes performance under the worst-case proxy consistent with the correlation constraint. We further show that when the reward is a linear function of known features, our approach can be adapted to incorporate this prior knowledge, yielding both improved policies and interpretable worst-case rewards. Experiments across several environments show that our algorithms consistently outperform ORPO in worst-case returns, and offer improved robustness and stability across different levels of proxy–true reward correlation. These results show that our approach provides both robustness and transparency in settings where reward design is inherently uncertain.

1 Introduction

Real-world reinforcement learning (RL) systems often struggle with reward specification: it is notoriously difficult to craft a reward function that perfectly captures the intended goals in all scenarios [1, 2, 3]. In practice, designers rely on proxy rewards that approximate the true objective [4]. However, agents optimizing these imperfect proxies can lead to unintended exploitative behaviors, achieving high proxy returns while yielding poor true outcomes, a phenomenon known as reward hacking [5, 6, 7, 8]. Such reward hacking behaviors are not merely hypothetical; they have led to undesirable or even catastrophic consequences in safety-critical settings (e.g., autonomous driving) [9, 10] and are alarmingly common in real-world deployments [11, 12, 13, 14]. Beyond reward hacking, interpretability and transparency of RL policies are increasingly recognized as critical requirements for real-world acceptance [15, 16, 17]. Policymakers and practitioners in safety-critical domains require systems not only to be robust but also interpretable; they must understand which specific decision-making criteria lead to undesirable outcomes to effectively mitigate risks and ensure compliance with safety regulations [18, 19, 20]. These challenges highlight the need for RL algorithms to address two fundamental challenges: robustness to uncertain or poorly-specified rewards, and interpretability to facilitate oversight and compliance by human stakeholders, especially in high-stakes, real-world environments like traffic control [21], healthcare decision-making [22, 23], and pandemic response strategies [24].

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Recent work has begun to formalize reward hacking and develop principled mitigations. Laidlaw et al. [25] define a proxy reward to be *r-correlated* with the true reward if it maintains a correlation coefficient $r > 0$ on state-action pairs encountered by a certain reference (baseline) policy. Notably, their definition permits the proxy and true reward to diverge arbitrarily in parts of the state-action space not visited by the reference policy, precisely the regions an RL agent might exploit under intensive optimization. Using this framework, reward hacking is formalized as the situation in which optimizing an r-correlated proxy yields a policy with lower true reward than that of the reference policy. Building on this definition, Laidlaw et al. propose Occupancy-Regularized Policy Optimization (ORPO) as a mitigation strategy. ORPO augments the standard RL objective with a regularization term that penalizes deviations between the learned policy's occupancy measure (state-action visitation distribution) and that of the reference policy.

Despite significant progress, existing solutions to reward hacking show several limitations. First, their effectiveness relies heavily on the choice of the specific proxy reward. However, designing perfect proxies is challenging, and in real-world scenarios, reward proxies are often derived heuristically or empirically from noisy or limited data [26, 27], leading to uncertainty or variability in the exact correlation with true rewards. Therefore, robustness to variations in proxy rewards is crucial for dependable deployment. While the regularization method used by ORPO provides a lower bound on improvement in true reward, its guarantee on the worst-case performance against an adversarially chosen proxy is weak. Second, current methods like ORPO typically treat a reward function as a black box and learn a complex policy with no easily interpretable structure, making it hard to understand why the resulting policy avoids reward hacking or to trust its behavior in novel situations. Further, they cannot be easily adapted to incorporate prior knowledge of the true reward. These shortcomings underscore the need for a more robust and transparent approach to reward hacking in RL.

In this work, we formalize reward hacking as a robust RL problem under proxy reward uncertainty and develop new algorithms to address the above gaps. The key idea is to optimize against an adversarial proxy reward rather than trusting a single proxy. We assume the true reward could be any function that remains r-correlated with the proxy (per the reference policy), and we train the agent to perform well against the worst-case such proxy. This approach explicitly accounts for uncertainty in proxy design and guards against unintended exploitative behaviors. Concretely, we propose a max-min formulation in which the policy chooses its strategy to maximize its guaranteed true return while an adversary minimizes the true return by selecting a reward function from the set of all r-correlated proxies. By solving this problem, the agent learns a policy that is robust to all plausible deviations of the proxy reward within the correlation bound. We derive a closed-form solution for the adversary's worst-case reward assignment given any candidate policy, which allows efficient evaluation of the inner minimization and provides insight into how proxy reward flaws are most damaging. Building on this result, we introduce a practical algorithm for Max-Min Policy Optimization that iteratively updates the policy against this worst-case reward signal.

Moreover, to improve the tractability and transparency of the inner optimization, we introduce a Linear Max-Min variant of our method. In this variant, we assume the true reward lies in a class of linear functions over known features, allowing us to characterize the worst-case proxy reward as a sparse linear combination of those features. While the policy itself remains parameterized by general neural networks, the learned worst-case reward function becomes interpretable in terms of its feature weights. This provides insight into which aspects of the proxy reward space the policy is robust to or vulnerable against, making it valuable for applications where understanding the failure modes of the reward design is important.

Finally, we empirically evaluate the proposed approaches on several challenging environments. Across all domains, our Max-Min and Linear Max-Min policies outperform ORPO in terms of worst-case reward, indicating substantially improved robustness. Moreover, under a large range of proxy-true correlation scenarios, our methods exhibit higher average reward and lower variance compared to ORPO, meaning the performance of our policies remains more consistent and reliable. These findings demonstrate the practical significance of our robust formulation, paving the way for safer and more trustworthy RL deployment in real-world applications.

Our main contributions can be summarized as follows: 1) We propose a novel robust RL formulation that explicitly models reward hacking as a max-min optimization problem over proxy rewards constrained by correlation with the true rewards. 2) We develop a practical algorithm for the max-min problem, which is further extended to linear rewards with improved robustness and interpretability. 3)

Experiment results demonstrate improved robustness and worst-case rewards across four real-world inspired reward hacking environments.

2 Preliminaries

Reinforcement Learning. A reinforcement learning (RL) problem can be formulated as an infinite-horizon Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, p, \mu_0, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $p(s' \mid s, a)$ is the transition probability from state s to s' given action a, and μ_0 is the initial state distribution. The agent interacts with the environment over discrete time steps $t = 0, 1, 2, \ldots$. At each time step, it selects an action $a_t \in \mathcal{A}$ based on the current state $s_t \in \mathcal{S}$ according to a policy $\pi(a \mid s)$, which defines a distribution over actions conditioned on the state. Upon taking action a_t, the agent receives a reward $R(s_t, a_t) \in \mathbb{R}$ and transitions to the next state s_{t+1} according to $p(s_{t+1} \mid s_t, a_t)$. The goal of the agent is to maximize the expected cumulative discounted return:

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

where $\gamma \in [0, 1)$ is the discount factor, and the expectation is taken over trajectories generated by following policy π. We define the *state-action occupancy measure* μ_π of a policy π as: $\mu_\pi(s, a) = (1-\gamma)\,\mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}\{s_t = s, a_t = a\}\right]$, which represents the discounted visitation frequency of each state-action pair under policy π. Using the occupancy measure, the return can be equivalently expressed as: $J(\pi, R) = \mathbb{E}_{(s,a)\sim\mu_\pi}[R(s, a)]$.

Correlated Proxies and Reward Hacking. Below we give an overview of the recently proposed r-correlated proxy framework proposed in [25] for detecting and mitigating reward hacking, which our work is built upon. A detailed discussion of related work on reward hacking and robust reinforcement learning is given in Appendix C.In particular, [25] considers a setting where the agent is given a reference policy π_{ref} and a proxy reward R_{proxy}, while the true reward is hidden. They further assume that the proxy reward is *r-correlated* with the true reward under the reference policy, that is:

$$\mathbb{E}_{\mu_{\pi_{\text{ref}}}}\left[\left(\frac{R_{\text{proxy}} - J(\pi_{\text{ref}}, R_{\text{proxy}})}{\sigma_{R_{\text{proxy}}}}\right)\left(\frac{R_{\text{true}} - J(\pi_{\text{ref}}, R_{\text{true}})}{\sigma_{R_{\text{true}}}}\right)\right] = r, \tag{2}$$

where $\sigma^2_{R_{\text{proxy}}} = \mathbb{E}_{\mu_{\pi_{\text{ref}}}}\left[\left(R_{\text{proxy}} - J(\pi_{\text{ref}}, R_{\text{proxy}})\right)^2\right]$ and $\sigma^2_{R_{\text{true}}} = \mathbb{E}_{\mu_{\pi_{\text{ref}}}}\left[\left(R_{\text{true}} - J(\pi_{\text{ref}}, R_{\text{true}})\right)^2\right]$ are the variances of the proxy and true rewards, respectively, under the reference policy. Reward hacking is said to occur when a policy π optimized for an r-correlated proxy reward achieves lower true reward than the reference policy: $J(\pi, R_{\text{true}}) < J(\pi_{\text{ref}}, R_{\text{true}})$. To mitigate reward hacking, [25] proposes Occupancy-Regularized Policy Optimization (ORPO) to optimize a regularized policy objective given below, which is shown to provide a lower bound on improvement in true reward:

$$\max_\pi\; J(\pi, R_{\text{proxy}}) - \lambda\sqrt{\chi^2(\mu_\pi \,\|\, \mu_{\pi_{\text{ref}}})}, \tag{3}$$

where $\chi^2(\mu_\pi \,\|\, \mu_{\pi_{\text{ref}}})$ denotes the χ^2-squared divergence between the occupancy measures of π and π_{ref}, and the regularization strength λ is set as: $\lambda = \sigma_{R_{\text{proxy}}}\sqrt{1-r^2}$. This encourages the learned policy to stay close to the reference distribution when the proxy reward is weakly correlated with the true reward.

3 Method

In this section, we discuss our robust policy optimization approach for mitigating reward hacking. In contrast to regularization-based methods such as ORPO, we consider a max-min formulation that identifies a robust policy with respect to the worst-case reward across all reward functions that are r-correlated with the proxy reward. We further extend our framework to settings where the reward function is a linear combination of known features with unknown weights. Our approach effectively leverages this structural information, when known a priori, to improve both robustness and interpretability, a task that is particularly challenging for regularization-based techniques.

3.1 Max-Min Policy Optimization

Similar to ORPO, we assume that the agent is given a proxy reward R_{proxy} and a reference policy π_{ref}, while the true reward is hidden. Rather than regularizing the policy under a fixed proxy reward, we consider the *entire space of rewards* $\mathcal{R}_{\text{corr}}$ that satisfy the correlation constraint with respect to a known proxy reward, as defined in Equation 4:

$$\mathcal{R}_{\text{corr}} = \left\{ R : (s,a) \to \mathbb{R} \;\middle|\; \mathbb{E}_{\mu_{\pi_{\text{ref}}}} \left[\frac{R - M}{V} \cdot R_{\text{proxy}} \right] = r,\ J(\pi_{\text{ref}}, R) = M,\ \sigma_R^2 = V^2 \right\}. \quad (4)$$

M and V denote the fixed mean and standard deviation of the reward function R under the reference policy π_{ref}. For simplicity, we define R_{proxy} to be the normalized proxy reward $R_{\text{proxy}}(s,a) := \frac{\tilde{R}_{\text{proxy}}(s,a) - J(\pi_{\text{ref}}, \tilde{R}_{\text{proxy}})}{\sigma_{\tilde{R}_{\text{proxy}}}}$, where $\tilde{R}_{\text{proxy}}$ is the original (unnormalized) proxy reward. After normalization, we have $J(\pi_{\text{ref}}, R_{\text{proxy}}) = 0$ and $\text{Var}_{\mu_{\pi_{\text{ref}}}}(R_{\text{proxy}}) = 1$, which simplifies the correlation constraint in Equation 4. The hyperparameter r controls the degree of alignment between the proxy and true reward. It allows us to interpolate between strong robustness (small r) and high proxy fidelity (large r), enabling a principled robustness-accuracy trade-off. We remark that it is without loss of generality to consider fixed M and V, which we will further elaborate on later.

We propose a *worst-case optimization framework* where the policy is trained to maximize expected performance under the least favorable reward within $\mathcal{R}_{\text{corr}}$. Assuming that the true reward lies somewhere within this set, this approach improves robustness by ensuring that the policy does not overfit to any single optimistic interpretation of the proxy reward. Formally, the objective becomes a max-min problem:

$$\max_{\pi} \min_{R \in \mathcal{R}_{\text{corr}}} J(\pi, R) = \max_{\pi} \min_{R \in \mathcal{R}_{\text{corr}}} \mathbb{E}_{(s,a)\sim\mu_\pi}[R(s,a)]. \quad (5)$$

However, a challenge arises: the objective $\mathbb{E}_{\mu_\pi}[R(s,a)]$ depends on the state-action occupancy μ_π, whereas the constraints defining $\mathcal{R}_{\text{corr}}$ are expressed in terms of $\mu_{\pi_{\text{ref}}}$. This mismatch complicates direct optimization. To resolve this, we apply a *change-of-measure* technique [28, 29] to rewrite the expectation under $\mu_{\pi_{\text{ref}}}$. Specifically, let $L(s,a)$ denote the Radon-Nikodym derivative: $L(s,a) = \frac{\mu_\pi(s,a)}{\mu_{\pi_{\text{ref}}}(s,a)}$. By definition, $L(s,a) \geq 0$ and $\mathbb{E}_{\mu_{\pi_{\text{ref}}}}[L(s,a)] = 1$. Applying the change-of-measure formula, we can express the return as: $\mathbb{E}_{\mu_\pi}[R(s,a)] = \int_{\mathcal{S}\times\mathcal{A}} \mu_\pi(s,a) R(s,a)\, d(s,a) = \int_{\mathcal{S}\times\mathcal{A}} \mu_{\pi_{\text{ref}}}(s,a) \frac{\mu_\pi(s,a)}{\mu_{\pi_{\text{ref}}}(s,a)} R(s,a)\, d(s,a) = \mathbb{E}_{\mu_{\pi_{\text{ref}}}}[L(s,a)R(s,a)]$. Thus, both the objective and the constraints can be rewritten as expectations with respect to the reference distribution $\mu_{\pi_{\text{ref}}}$.

For notational simplicity, we will suppress the variables (s,a) and write for example, L to denote $L(s,a)$ and R to denote $R(s,a)$. Under this reparameterization, the inner minimization in Equation 5 can be reformulated as:

$$\min_{R \in \mathcal{R}_{\text{corr}}} \mathbb{E}_{\mu_{\pi_{\text{ref}}}}[L \cdot R]. \quad (6)$$

Although the feasible set in Problem 6 is not convex due to the equality constraint on the variance, we still derive an optimal solution using a Lagrangian formulation. Our approach leverages tools from duality theory, commonly used in robust optimization [30, 31]. We further justify the validity of our solution in Appendix D.2. Specifically, the Lagrangian functional associated with this problem is defined as: $l_0(\lambda_1, \lambda_2, \lambda_3, R) = \mathbb{E}_{\mu_{\pi_{\text{ref}}}}[L \cdot R - \lambda_1 \frac{R-M}{V} \cdot R_{\text{proxy}} - \lambda_2 R - \lambda_3 R^2] + \lambda_1 r + \lambda_2 M + \lambda_3 (M^2 + V^2)$, where $\lambda_1, \lambda_2, \lambda_3$ are the Lagrange multipliers corresponding to the correlation constraint, mean constraint, and variance constraint, respectively. Then the original problem in Equation 6 is equivalent to the following problem:

$$\max_{\lambda_1,\lambda_2,\lambda_3} \min_{R \in \mathcal{R}_{\text{corr}}} l_0(\lambda_1, \lambda_2, \lambda_3, R). \quad (7)$$

We now solve the inner minimization problem in Equation 7 by finding the optimal R for fixed dual variables $(\lambda_1, \lambda_2, \lambda_3)$. Taking the functional derivative of the Lagrangian l_0 with respect to $R(s,a)$ gives: $\frac{\partial l_0}{\partial R} = \mu_{\pi_{\text{ref}}}(s,a)[(L - \lambda_1 \frac{R_{\text{proxy}}}{V} - \lambda_2) - 2\lambda_3 R]$. When $\mu_{\pi_{\text{ref}}}(s,a) > 0$, setting the derivative of the Lagrangian to zero yields the optimal adversarial reward function:

$$R^*(s,a) = \frac{L(s,a) - \lambda_1 \frac{R_{\text{proxy}}}{V} - \lambda_2}{2\lambda_3}. \quad (8)$$

However, for state-action pairs where $\mu_{\pi_{\text{ref}}}(s,a) = 0$, i.e., those not visited under the reference policy, the correlation and moment constraints become vacuous. In these regions, the adversarial reward $R^*(s,a)$ can be driven arbitrarily poor, reflecting that no constraint prevents the adversary from assigning highly penalizing values to rarely visited or unobserved state-action pairs. Nevertheless, consider the case where $\mu_{\pi_{\text{ref}}}(s,a) > 0$, we can substitute the optimal R^* from Equation 8 into the Lagrangian l_0 and get the dual objective. After some process detailed in Appendix D.1, we get the optimal solution to the inner problem (6), so the original max-min problem (5) reduces to:

$$\max_{\pi} \; r \cdot V \cdot \mathbb{E}_{\mu_\pi}[R_{\text{proxy}}] - V \cdot \sqrt{1-r^2}\sqrt{\chi^2(\mu_\pi \,\|\, \mu_{\pi_{\text{ref}}}) - \mathbb{E}^2_{\mu_\pi}[R_{\text{proxy}}]} + M. \tag{9}$$

Thus, the final policy optimization objective becomes maximizing the proxy reward, regularized by a penalty that depends on the distributional shift between μ_π and $\mu_{\pi_{\text{ref}}}$ and the expectation of the current policy under proxy reward $\mathbb{E}_{\mu_\pi}[R_{\text{proxy}}]$, and the correlation strength r. We observe that the constants M and V do not affect the optimal policy: while they influence the absolute value of the worst-case reward for a given policy π, they only apply a linear transformation (scaling by V and shifting by M) and do not change the relative ordering of policies. Therefore, for simplicity, we set $V = 1$ and $M = 0$ in our implementation. This also provides a fair way to compare the worst-case rewards of different policies. Notice that the optimization objective in Equation 9 closely resembles the ORPO objective proposed in Equation 3. However, there are two key differences: (1) our regularization strength is $\frac{\sqrt{1-r^2}}{r}$ instead of $\sigma_{R_{\text{proxy}}}\sqrt{1-r^2}$, and (2) our penalty term is $\chi^2(\mu_\pi \,\|\, \mu_{\pi_{\text{ref}}}) - \mathbb{E}^2_{\mu_\pi}[R_{\text{proxy}}]$ rather than simply $\chi^2(\mu_\pi \,\|\, \mu_{\pi_{\text{ref}}})$. The proof that $\chi^2(\mu_\pi \,\|\, \mu_{\pi_{\text{ref}}}) - \mathbb{E}^2_{\mu_\pi}[R_{\text{proxy}}] \geq 0$ holds can be found in Appendix D.3. A detailed comparison between our policy gradient and that of ORPO is provided in Appendix D.8.

3.2 Structured Reward Spaces via Feature Linearization

A natural concern with worst-case optimization is *over-conservatism*: if the reward uncertainty set $\mathcal{R}_{\text{corr}}$ is too broad, the resulting policy may become overly cautious or deviate from realistic task objectives. Additionally, the learned worst-case rewards may themselves be implausible or uninterpretable. To address these issues, we introduce *structure* into the reward space by assuming that all rewards are *linear combinations of known features*. Specifically, we assume: $R(s,a) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(s,a)$, where $\boldsymbol{\phi}(s,a) = [\phi_1(s,a), \phi_2(s,a), \ldots, \phi_M(s,a)]^\top \in \mathbb{R}^M$ denotes a vector of M known or engineered feature functions, and $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_M]^\top \in \mathbb{R}^M$ represents the uncertain feature weights. The linearization yields two key benefits: 1) **Realism and Interpretability:** In many real-world tasks, reward functions are naturally approximated as linear combinations over interpretable features. For example, in a traffic control environment, features might include total commute time, vehicle speed, acceleration, and inter-vehicle headway distances. 2) **Better-Constrained Robustness:** By restricting uncertainty to structured, feature-based rewards, the worst-case optimization problem becomes more grounded and avoids pathological, unrealistic reward functions.

In this section, we assume that the agent is aware of the set of features but not their true weights. We show that our robust optimization framework can be naturally extended to incorporate the structure in rewards to improve robustness. In our experiments, we further demonstrate that linear rewards help interpret a policy's performance even when it is trained without such prior knowledge. Under our assumption, the uncertainty set reduces to the set of feature weights $\boldsymbol{\theta} \in \mathbb{R}^M$ satisfying:

$$\mathcal{R}^{\text{lin}}_{\text{corr}} = \left\{ \boldsymbol{\theta} \in \mathbb{R}^M \;\middle|\; \mathbb{E}_{\mu_{\pi_{\text{ref}}}}[\boldsymbol{\theta}^\top \boldsymbol{\phi} \cdot R_{\text{proxy}}] = r,\; \mathbb{E}_{\mu_{\pi_{\text{ref}}}}[\boldsymbol{\theta}^\top \boldsymbol{\phi}] = 0,\; \mathbb{E}_{\mu_{\pi_{\text{ref}}}}[(\boldsymbol{\theta}^\top \boldsymbol{\phi})^2] = 1 \right\}. \tag{10}$$

To simplify the analysis, we assume without loss of generality that the worst-case reward $R(s,a) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(s,a)$ is normalized to have zero mean and unit variance under the reference policy π_{ref}. This corresponds to setting $M = 0$ and $V = 1$, which, as shown in our earlier derivation, does not affect the resulting optimal policy. As before, R_{proxy} denotes the normalized proxy reward under π_{ref}, satisfying $\mathbb{E}_{\mu_{\pi_{\text{ref}}}}[R_{\text{proxy}}] = 0$ and $\text{Var}_{\mu_{\pi_{\text{ref}}}}[R_{\text{proxy}}] = 1$.

We now derive the corresponding max-min optimization under the structured reward assumption:

$$\max_{\pi} \min_{\boldsymbol{\theta} \in \mathcal{R}^{\text{lin}}_{\text{corr}}, \boldsymbol{\theta} \geq 0} \mathbb{E}_{(s,a)\sim\mu_\pi}\left[\boldsymbol{\theta}^\top \boldsymbol{\phi}(s,a)\right]. \tag{11}$$

Similar to previous steps, we introduce the Radon-Nikodym derivative $L(s,a) = \frac{\mu_\pi(s,a)}{\mu_{\pi_{\text{ref}}}(s,a)}$, use a change-of-measure, and define the Lagrangian functional for the inner minimization in Equation 11 as: $l_1(\lambda_1, \lambda_2, \lambda_3, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \left(\sum_{(s,a)} u_{\lambda_1,\lambda_2}(s,a)\boldsymbol{\phi}(s,a)\right) - \lambda_3 \boldsymbol{\theta}^\top Q \boldsymbol{\theta} + \lambda_1 r + \lambda_3$, where $u_{\lambda_1,\lambda_2} = \mu_\pi - \lambda_1 \mu_{\pi_{\text{ref}}} R_{\text{proxy}} - \lambda_2 \mu_{\pi_{\text{ref}}}$, $Q = \sum_{(s,a)} \mu_{\pi_{\text{ref}}}(s,a)\boldsymbol{\phi}(s,a)\boldsymbol{\phi}(s,a)^\top$. A detailed derivation can be found in Appendix D.4. Note that Q is positive semi-definite since it is a sum of outer products $\boldsymbol{\phi}(s,a)\boldsymbol{\phi}(s,a)^\top$ weighted by non-negative coefficients (occupancy measure of $\pi_{\text{ref}} \geq 0$). Then solving the inner minimization over $\boldsymbol{\theta}$ in Equation 11 is equivalent to solving:

$$\max_{\lambda_1,\lambda_2,\lambda_3} \min_{\boldsymbol{\theta}\geq 0} \quad l_1(\lambda_1, \lambda_2, \lambda_3, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \left(\sum u_{\lambda_1,\lambda_2}\boldsymbol{\phi}\right) - \lambda_3 \boldsymbol{\theta}^\top Q \boldsymbol{\theta} + \lambda_1 r + \lambda_3. \tag{12}$$

Notice that $l_1(\lambda_1, \lambda_2, \lambda_3, \boldsymbol{\theta})$ is a convex quadratic function of $\boldsymbol{\theta}$ (assuming $\lambda_3 \leq 0$) subject to linear inequality constraints $\boldsymbol{\theta} \geq 0$. Thus, the original problem is a standard convex quadratic program (QP) with non-negativity constraints [32]. However, it is not possible to derive a universal closed-form solution for the optimal $\boldsymbol{\theta}^*$ under arbitrary Q. To further simplify the problem and obtain a closed-form solution, we transform the feature vector $\boldsymbol{\phi}$ into a whitened version $\tilde{\boldsymbol{\phi}}$ such that the matrix Q becomes the identity matrix I and we formally show this in Appendix D.5. Specifically, we perform a whitening transformation using the Cholesky decomposition [32]. Let $W = Q^{-\frac{1}{2}}, \tilde{\boldsymbol{\phi}}(s,a) = W\boldsymbol{\phi}(s,a)$, where $Q^{-\frac{1}{2}}$ denotes a matrix square root of Q^{-1} (which exists since Q is positive semi-definite and non-singular assuming $\exists(s,a)$ such that $\mu_{\pi_{\text{ref}}}(s,a) > 0$). Then the original problem in Equation 12 can be further simplified into:

$$\max_{\lambda_1,\lambda_2,\lambda_3} \min_{\tilde{\boldsymbol{\theta}}\geq 0} \quad l_1(\lambda_1, \lambda_2, \lambda_3, \tilde{\boldsymbol{\theta}}) = \tilde{\boldsymbol{\theta}}^\top \left(\sum_{(s,a)} u_{\lambda_1,\lambda_2}(s,a)\tilde{\boldsymbol{\phi}}(s,a)\right) - \lambda_3 \tilde{\boldsymbol{\theta}}^\top \tilde{\boldsymbol{\theta}} + \lambda_1 r + \lambda_3. \tag{13}$$

where we now optimize over the parameter $\tilde{\boldsymbol{\theta}}$ using the transformed features $\tilde{\boldsymbol{\phi}}$. For notational simplicity, we will drop the tilde and henceforth use $\boldsymbol{\phi}$ to represent the whitened feature $\tilde{\boldsymbol{\phi}}$, and $\boldsymbol{\theta}$ to represent the whitened weights $\tilde{\boldsymbol{\theta}}$. Then we can get a closed-form solution (we detail the steps in Appendix D.6) for optimal $\boldsymbol{\theta}^*$ as: $\boldsymbol{\theta}^* = \max\left(0, -\frac{\sum_{(s,a)} u_{\lambda_1,\lambda_2}(s,a)\boldsymbol{\phi}(s,a)}{2\lambda_3}\right)$, where the $\max(\cdot, 0)$ is applied elementwise. Details for solving the outer maximization in Equation 13 can be found in Appendix D.7. After obtaining the optimal dual variables $(\lambda_1^*, \lambda_2^*, \lambda_3^*)$, we can substitute them back into Equation 11 and solve the outer maximization over the policy π using standard reinforcement learning algorithms, such as PPO [33].

ORPO with Linear Awards. While ORPO provides a general guarantee based on occupancy measure regularization, it does not exploit any structural assumptions about the reward function. In particular, even when the true reward is linear in a set of features, ORPO does not explicitly incorporate this structure into its policy optimization or theoretical analysis. While the lower bound (Theorem 5.1 in [25]) continues to hold, it is unclear how to leverage this structure to obtain a tighter lower bound or to guide policy updates more effectively. This suggests a missed opportunity: by explicitly modeling the reward as a linear function, it becomes possible to derive stronger guarantees, interpret worst-case reward directions, and efficiently optimize against them. Our Linear Maxmin method fills this gap by parameterizing reward uncertainty directly in the space of reward weights, enabling both robustness and greater transparency.

3.3 Implementation Details and Algorithms

A core step in both our algorithms and ORPO is to estimate the Radon-Nikodym derivative $L(s,a)$. To this end, we follow prior works [25, 34, 35] and train a discriminator network. Specifically, we use a discriminator architecture identical to that in [25], denoted by $d_\phi(s,a)$, which is optimized according to:

$$\phi = \arg\min_\phi \; \mathbb{E}_{\mu_{\pi_{\text{ref}}}}[\log(1 + e^{d_\phi(s,a)})] + \mathbb{E}_{\mu_\pi}[\log(1 + e^{-d_\phi(s,a)})]. \tag{14}$$

It is known that the optimal discriminator satisfies $d^*(s,a) = \log \frac{\mu_\pi(s,a)}{\mu_{\pi_{\text{ref}}}(s,a)}$ and we estimate $L(s,a)$ as $\tilde{L}(s,a) = \exp d_\phi(s,a)$ with $d_\phi(s,a) \approx d^*(s,a)$. As discussed in Section 3.1, if the policy π visits state-action pairs that the reference policy π_{ref} rarely or never visits, the adversarial reward can be arbitrarily poor. In theory, the estimated $\tilde{L}(s,a)$ is expected to grow arbitrarily large in this

case, which should discourage the learned policy from exploiting such regions. However, we observe empirically (Section 4.2) that the ORPO policy still visits some of these low-coverage regions under π_{ref}. This is because in the original ORPO implementation[1], the discriminator is not fully optimized during policy learning. Specifically, the discriminator receives only a small number of gradient updates per reinforcement learning iteration, resulting in underfitting and inaccurate estimates of the Radon-Nikodym derivative $\tilde{L}(s, a)$. To address this, we substantially increase the number of gradient updates per iteration and carefully tune the learning rate. Our goal is to strike a practical balance between training time and discriminator quality, which we further discuss in Appendix E.1.

To compute the final objective for our Max-Min policy in Equation 9, we estimate the χ^2 divergence, the normalized proxy reward R_{proxy}, and the first and second moments $\mathbb{E}_{\mu_\pi}[R_{\text{proxy}}]$ and $\mathbb{E}^2_{\mu_\pi}[R_{\text{proxy}}]$. These components together define the robust optimization objective used to update the policy. A simplified Max-Min policy optimization procedure is outlined in Algorithm 1. We provide detailed descriptions of each estimation step, as well as the complete algorithmic implementation in Appendix E.2. Corresponding derivations and implementation details for the Linear Max-Min variant are included in Appendix E.3.

Algorithm 1 Max-Min Policy Optimization (Simplified)

```
Initialize policy parameters θ
Initialize reference policy π_ref and collect trajectories
Estimate mean and variance of the proxy reward under π_ref
for each iteration do
    Collect trajectories from current policy π_θ
    Normalize the proxy rewards for state-action pairs in the collected trajectories
    Estimate the expected proxy reward and its second moment under the current policy
    Estimate the discriminator using Equation (14) and χ² divergence between μ_π and μ_π_ref
    Update the policy using PPO to maximize the Max-Min objective in Equation (9)
end for
```

4 Experiment

4.1 Experiment Setup

We evaluate our method across four realistic benchmark environments: *Traffic*, *Pandemic*, *Glucose Monitoring*, and *Tomato Watering GridWorld*. These environments were originally proposed in [36, 5] and represent diverse forms of proxy reward hacking, including misweighting, ontological mismatch, and scope misalignment [36]. Each setting presents unique challenges in reward specification and policy robustness. A detailed description of the environments and their respective reward structures is provided in Appendix E.4. In each of the four environments, we train policies using both our `Max-Min` and `Linear Max-Min` optimization algorithms. For baselines, we compare against the `ORPO` policy. To isolate the impact of discriminator training, we also include an ablation: `ORPO*`, where we train the ORPO policy using the same full discriminator training schedule as in our algorithms. This variant shares the same architecture and optimization settings as the original ORPO, differing only in the extent of discriminator training. Including this baseline allows us to evaluate the specific contribution of discriminator optimization to policy robustness. We include more detailed experimental settings in Appendix E.5 and a discussion of training time and complexity of all algorithms in Appendix E.6.

As for evaluation metrics, we report both the expected proxy and true rewards, along with the expected worst-case reward as described in Section 3.1. Note that some policies may visit state-action pairs that are not covered by the reference policy π_{ref}. In such cases, we exclude those trajectories and report the occupancy measure of the unseen state-action pairs. Additionally, we evaluate each policy using two variants of the expected linear worst-case reward introduced in Section 3.2. The first uses only the features present in the proxy reward, while the second variant, denoted *Linear Worst**, leverages features from the true reward, some of which remain unseen during training. This setup mimics a more realistic real-world scenario in which the true reward function may depend on features not explicitly modeled at training time. Comparing performance under this setting allows us to assess the robustness of each policy to unseen or misaligned reward structures. All rewards are normalized with respect to the reference policy π_{ref} to ensure a consistent scale across metrics, enabling fair and

[1] `https://github.com/cassidylaidlaw/orpo/tree/main`

Table 1: Evaluation results on Traffic and Pandemic environments. All policies are trained using **only the proxy reward**. In Traffic, the proxy reward is based on *vel*, *accel*, *headway* (1, 1, 0.1), while the true reward uses *commute*, *accel*, *headway* (1, 1, 0.1). In Pandemic, the proxy reward includes *infection*, *lower stage*, *smooth changes* (10, 0.1, 0.01), while the true reward additionally includes *political* with weight 10 after *infection*. We report $\boldsymbol{\theta}$ in the same order as feature weights. **Occ** denotes total occupancy over state-action pairs unseen by π_{ref}, where discriminator outputs infinity.

Env	**Traffic**					
Method	**True**	**Proxy**	**Worst**	**Linear Worst (θ)**	**Linear Worst* (θ)**	**Occ ↓**
ORPO	16.93	3.31	-1.97e+04	-0.68 (0.71, 0.21, 0.69)	-0.81 (0.63, 0.12, 0.97)	3.71e-04
ORPO*	10.31	1.32	-1.33e+04	-0.42 (0.46, 0.18, 0.86)	-0.44 (0.58, 0.06, 0.81)	1.90e-05
Max-Min	12.64	3.64	-270.84	-0.07 (0.01, 0.02, 0.96)	-0.07 (0.001, 0.02, 0.99)	0
Linear Max-Min	16.36	2.53	-1.19e+04	0.21 (0.64, 0.07, 0.76)	-0.12 (0.91, 0.01, 0.67)	0
Env	**Pandemic**					
Method	**True**	**Proxy**	**Worst**	**Linear Worst (θ)**	**Linear Worst* (θ)**	
ORPO	-0.91	1.81	-5.30e+06	-2.42 (0.23, 0.95, 0.17)	-2.63 (0.02, 0.95, 0.92, 0.08)	
ORPO*	1.24	1.24	-4.42e+06	-1.35 (0.25, 0.97, 0.13)	-1.35 (0.25, 0, 0.97, 0.13)	
Max-Min	1.15	1.15	-65.69	-1.11 (0.14, 0.99, 0.01)	-1.11 (0.14, 0, 0.99, 0.01)	
Linear Max-Min	2.61	7.56	-6.83e+05	0.66 (0.001, 0.23, 0.02)	-0.17 (0.01, 0.97, 0.22, 0.09)	

meaningful comparisons. Note that all worst-case rewards are reported using the fixed correlation level r specified during training: $r = 0.3$ for Traffic, $r = 0.7$ for Pandemic, with values for other environments provided in Appendix E.5.

4.2 Results

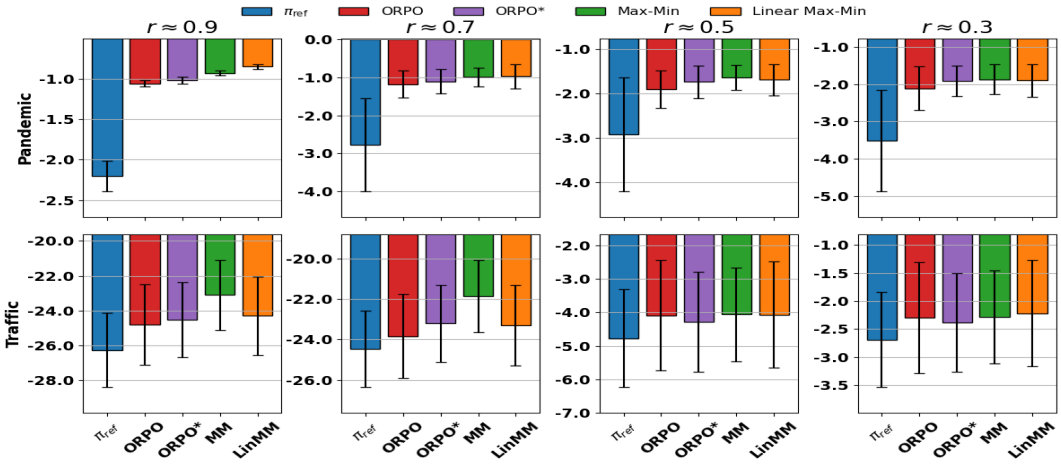

Figure 1: Mean reward and standard deviation under sampled θ and true reward features at different proxy–true reward correlation levels r for the Traffic and Pandemic environments. Our methods (`Maxmin` and `Linear Maxmin`) yield more stable and higher average performance across all choices of r.

Worst-Case Performance. Table 1 presents the evaluation results on the Traffic and Pandemic environments. Additional results for other environments are provided in Appendix F. Our `Max-Min` and `Linear Max-Min` policies achieve better expected worst-case performance under both general and linear adversarial rewards, while remaining competitive with baselines in terms of expected true and proxy rewards. Notably, the `Max-Min` policy attains the highest expected worst-case return, followed by `Linear Max-Min`. Conversely, `Linear Max-Min` yields the highest expected linear worst-case reward, followed by `Max-Min`, demonstrating the robustness of both approaches under worst-case scenarios. For the Linear Worst* evaluation, which uses reward features unseen during training, we observe minimal degradation in `Max-Min` policy's performance, indicating its strong robustness to feature variation. In contrast, the performance of `Linear Max-Min` declines in this case, suggesting its advantage diminishes when prior assumptions about feature structure are inaccurate.

We find that `ORPO*` exhibits better worst-case performance than the original `ORPO`. In particular, training the discriminator more thoroughly significantly reduces the occupancy of state-action pairs that are not visited by the reference policy, indicating that more accurate estimation of the Radon–Nikodym derivative leads to improved policy robustness. Notably, in the Pandemic environment, we observe no such unvisited state-action pairs, and the discriminator outputs remain small across all policies. This

could be due to either the discriminator network not being fully optimized or its inability to capture rare events that fall outside the support of π_{ref}. Developing more reliable techniques for handling such rare or unseen state-action pairs remains an open direction for future work.

We also report the adversarial weight vectors $\boldsymbol{\theta}$ for each policy. These weights reveal which features are most vulnerable to proxy exploitation under the learned policy and can be used to diagnose and revise the proxy reward function, thereby improving robustness. This highlights the interpretability benefits of our framework. Moreover, several patterns emerge from the results. In the Traffic environment, first, we observe a clear dominance of the headway feature, with all methods assigning it the highest weight. This suggests that headway is the most critical component exposed to reward hacking under correlation constraints. Second, the acceleration feature is consistently downweighted across all methods. This indicates that acceleration may be less prone to exploitation or already well aligned with the reference policy. Third, the velocity feature is moderately emphasized by `Linear Max-Min` and `ORPO` (e.g., 0.64 and 0.71), while `Max-Min` nearly suppresses it (0.01). This contrast suggests that `Linear Max-Min` anticipates some vulnerability from velocity deviations, while `Max-Min` focuses almost entirely on headway. In the Pandemic environment, first, both `ORPO*` and `Max-Min` assign zero weight to the political feature. This occurs because the expected feature value under their policies is exactly zero, making the correlation constraint inactive for that dimension. Interestingly, this feature plays a significant role in the adversarial rewards for both `ORPO` and `Linear Max-Min`, with their corresponding $\boldsymbol{\theta}$ assigning non-negligible weight to it (e.g., 0.95 and 0.97 respectively). This suggests that these policies expose themselves to vulnerability in feature dimensions that are entirely ignored by `Max-Min` and `ORPO*`. Second, the lower stage feature consistently receives the highest weight across all methods, indicating it is the most sensitive component under proxy misalignment.

Robustness Across Correlation Levels. To further assess the robustness of each policy across a broader range of proxy–true correlation scenarios, we also compute the Linear Worst* for each policy under varying r values. Specifically, for each r, we sample 1000 vectors $\boldsymbol{\theta}$ such that $\boldsymbol{\theta} \in \mathcal{R}^{\text{lin}}_{\text{corr}}$, and report the average return and variance achieved by each policy over these sampled rewards. Importantly, the variation in r is applied **only during evaluation**; all policies are fixed and trained using the specific r values reported in Appendix E.5. Unlike evaluations that only consider several reward functions, this approach evaluates policy performance across the entire reward set $\mathcal{R}^{\text{lin}}_{\text{corr}}$, providing a more comprehensive measure of robustness and better reflecting real-world scenarios where the true reward and correlation r are unknown.

Figure 1 shows the average reward and variance achieved by each method under different levels of proxy–true reward correlation r. As expected, the base policy π_{base} (blue) performs the worst across all correlation levels in both environments. In Traffic, its variance is relatively small, suggesting consistently poor but stable behavior. In contrast, variance is highest in the Pandemic environment, indicating increased policy fragility. Notably, `ORPO*` (purple) consistently achieves lower variance than `ORPO` (red) across both environments and outperforms it in terms of average reward at $r \approx 0.9$ and $r \approx 0.7$ in Traffic, and across nearly all r values in Pandemic. This underscores the importance of accurate discriminator training for improving both stability and robustness. `Max-Min` (green) demonstrates the highest average reward and lowest variance across a wide range of r values in both environments, showing strong resilience to reward misspecification. While `Linear Max-Min` (orange) achieves the best performance at specific correlation levels, particularly $r \approx 0.3$ in Traffic and $r \approx 0.7$–0.9 in Pandemic. As r decreases and the proxy becomes less informative, differences in average reward among methods shrink, while variance increases. These results highlight the significance of variance control in low-correlation regimes and demonstrate that `Max-Min` and `Linear Max-Min` offer robust and stable performance under high uncertainty.

5 Conclusion

In this work, we propose a robust policy optimization framework that explicitly accounts for reward hacking by training policies against the worst-case proxy reward drawn from a correlation-constrained uncertainty set. Our approach formalizes reward hacking as a robust optimization problem and introduces both a Max-Min formulation with a closed-form adversarial reward and a Linear Max-Min variant that further improves interpretability and tractability. We develop efficient algorithms and empirically validate our methods across diverse environments known to exhibit reward hacking behavior. Our results demonstrate that both Max-Min and Linear Max-Min policies achieve stronger worst-case performance and improved stability compared to prior baselines such as ORPO.

References

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[2] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. In *Advances in Neural Information Processing Systems*, 2018.

[3] Jonathan Stray, Alon Halevy, Parisa Assar, Dylan Hadfield-Menell, Craig Boutilier, Amar Ashar, Chloe Bakalar, Lex Beattie, Michael Ekstrand, Claire Leibowicz, et al. Building human values into recommender systems: An interdisciplinary synthesis. *ACM Transactions on Recommender Systems*, 2(3):1–57, 2024.

[4] 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. In *International Conference on Learning Representations*, 2023.

[5] Jan Leike, Miljan Martic, Victoria Krakovna, Pedro Ortega, Tom Everitt, Ryan Lefrancq, Laurent Orseau, and Shane Legg. AI safety gridworlds. *arXiv preprint arXiv:1711.09883*, 2017.

[6] Tom Everitt, Victoria Krakovna, Laurent Orseau, Marcus Hutter, and Shane Legg. Reinforcement learning with a corrupted reward channel. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[7] Tom Everitt, Marcus Hutter, Ramana Kumar, and Victoria Krakovna. Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective. *Synthese*, 198(Suppl 27):6435–6467, 2021.

[8] Jack Koch, Lauro Langosco, Jacob Pfau, James Le, and Lee Sharkey. Objective robustness in deep reinforcement learning. *arXiv preprint arXiv:2105.14111*, 2, 2021.

[9] Victoria Krakovna, Laurent Orseau, Ramana Kumar, Miljan Martic, and Shane Legg. Penalizing side effects using stepwise relative reachability. *arXiv preprint arXiv:1806.01186*, 2018.

[10] W Bradley Knox, Alessandro Allievi, Holger Banzhaf, Felix Schmitt, and Peter Stone. Reward (mis) design for autonomous driving. *Artificial Intelligence*, 316:103829, 2023.

[11] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. The challenge of understanding what users want: Inconsistent preferences and engagement optimization. *Management Science*, 70(9):6336–6355, 2024.

[12] Matt Franchi, JD Zamfirescu-Pereira, Wendy Ju, and Emma Pierson. Detecting disparities in police deployments using dashcam data. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 534–544, 2023.

[13] Smitha Milli, Luca Belli, and Moritz Hardt. From optimizing engagement to measuring value. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 714–722, 2021.

[14] 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, 2019.

[15] George A Vouros. Explainable deep reinforcement learning: state of the art and challenges. *ACM Computing Surveys*, 55(5):1–39, 2022.

[16] Erika Puiutta and Eric MSP Veith. Explainable reinforcement learning: A survey. In *International cross-domain conference for machine learning and knowledge extraction*, pages 77–95. Springer, 2020.

[17] Rahul Iyer, Yuezhang Li, Huao Li, Michael Lewis, Ramitha Sundar, and Katia Sycara. Transparency and explanation in deep reinforcement learning neural networks. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 144–150, 2018.

[18] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

[19] Jeff Druce, Michael Harradon, and James Tittle. Explainable artificial intelligence (xai) for increasing user trust in deep reinforcement learning driven autonomous systems. *arXiv preprint arXiv:2106.03775*, 2021.

[20] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *stat*, 1050:2, 2017.

[21] 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 *Conference on Robot Learning*, pages 399–409. PMLR, 2018.

[22] Ian Fox, Joyce Lee, Rodica Pop-Busui, and Jenna Wiens. Deep reinforcement learning for closed-loop blood glucose control. In *Machine Learning for Healthcare Conference*, pages 508–536. PMLR, 2020.

[23] Andreas Holzinger, Chris Biemann, Constantinos S Pattichis, and Douglas B Kell. What do we need to build explainable ai systems for the medical domain? *arXiv preprint arXiv:1712.09923*, 2017.

[24] V Kompella, R Capobianco, S Jong, J Browne, S Fox, L Meyers, P Wurman, P Stone, et al. Reinforcement learning for optimization of covid-19 mitigation policies. In *CEUR WORKSHOP PROCEEDINGS*, volume 2884. CEUR-WS, 2020.

[25] Cassidy Laidlaw, Shivam Singhal, and Anca Dragan. Correlated proxies: A new definition and improved mitigation for reward hacking. In *International Conference on Learning Representations*, 2025.

[26] Hong Jun Jeon, Smitha Milli, and Anca Dragan. Reward-rational (implicit) choice: A unifying formalism for reward learning. In *Advances in Neural Information Processing Systems*, 2020.

[27] Dorsa Sadigh, Anca Dragan, Shankar Sastry, and Sanjit Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems*, 2017.

[28] Zhaolin Hu and L Jeff Hong. Kullback-leibler divergence constrained distributionally robust optimization. *Available at Optimization Online*, 1(2):9, 2013.

[29] Henry Lam. Robust sensitivity analysis for stochastic systems. *Mathematics of Operations Research*, 41(4):1248–1275, 2016.

[30] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

[31] Joel Goh and Melvyn Sim. Distributionally robust optimization and its tractable approximations. *Operations research*, 58(4-part-1):902–917, 2010.

[32] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

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

[34] Bingyi Kang, Zequn Jie, and Jiashi Feng. Policy optimization with demonstrations. In *International Conference on Machine Learning (ICML)*, 2018.

[35] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, 2016.

[36] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*, 2022.

[37] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. Inverse reward design. In *Advances in Neural Information Processing Systems*, 2017.

[38] Rohin Shah, Vikrant Varma, Ramana Kumar, Mary Phuong, Victoria Krakovna, Jonathan Uesato, and Zac Kenton. Goal misgeneralization: Why correct specifications aren't enough for correct goals. *arXiv preprint arXiv:2210.01790*, 2022.

[39] Garud Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

[40] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[41] Jun Morimoto and Kenji Doya. Robust reinforcement learning. *Neural Computation*, 17(2):335–359, 2005.

[42] Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning (ICML)*, pages 6215–6224, 2019.

[43] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

[44] Andreas Schlaginhaufen and Maryam Kamgarpour. Identifiability and generalizability in constrained inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2023.

[45] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine learning (ICML)*, 2004.

[46] Dotan Di Castro, Aviv Tamar, and Shie Mannor. Policy gradients with variance related risk criteria. In *International Conference on Machine learning (ICML)*, 2012.

[47] Tengyang Xie, Bo Liu, Yangyang Xu, Mohammad Ghavamzadeh, Yinlam Chow, Daoming Lyu, and Daesub Yoon. A block coordinate ascent algorithm for mean-variance optimization. In *Advances in Neural Information Processing Systems*, 2018.

[48] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

[49] Jorge J. Moré. The levenberg-marquardt algorithm: Implementation and theory. Technical Report ANL-80-20, Argonne National Laboratory, Argonne, IL, 1978. Lecture Notes in Mathematics, vol. 630.

[50] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.

[51] Cathy Wu, Abdul Rahman 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, 2021.

[52] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805, 2000.

[53] Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton, Boris Kovatchev, and Claudio Cobelli. The uva/padova type 1 diabetes simulator: new features. *Journal of diabetes science and technology*, 8(1):26–34, 2014.

[54] 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, 2013.

[55] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[56] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph E Gonzalez, Michael I Jordan, and Ion Stoica. Rllib: Abstractions for distributed reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2018.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claim is supported by the theoretical and experiment results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitation can be found in the Appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide detailed proof in the Appendix

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We give experiment design details in the paper and the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in the supplemental material?

Answer: [Yes]

Justification: We will provide our code in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide details in the experiment section and the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report error bars in Figure 1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Appendix E.6

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, our research conforms the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We mentioned it in the Appendix

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly cited every previous work we build upon.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.