The Consequences of the Intrinsic Gap Between Reward Beliefs and MDP Rewards

Anonymous authors
Paper under double-blind review

Abstract

Deep neural policies have gained the ability to learn and execute sequences of decisions in MDPs that involve complex and high-dimensional states. Despite the growing use of reinforcement learning in diverse fields from language agents to medical and finance, a line of research has focused on constructing reward functions by observing how an optimal policy behaves, with the underlying premise that this will result in policies that are aligned with the intended outcome. In this line of research, several studies have proposed algorithms for learning a reward function or an optimal policy from observed optimal trajectories, with the goal of achieving sample-efficient, robust, and aligned policies. In this paper, we analyze the implications of learning with reward beliefs in high-dimensional state representation MDPs and we demonstrate that standard deep reinforcement learning yields more resilient and value-aligned policies when compared to learning from the behaviour of other policies in MDPs with complex state representations.

1 Introduction

Learning from raw high dimensional state observations became possible with the utilization of effective function approximators (Mnih et al., 2015; 2016; Kapturowski et al., 2023). This enhancement in the capabilities of reinforcement learning agents allowed these policies to be deployed in different fields from autonomous vehicles to healthcare (Hargrave et al., 2024) and language agents (Jaech et al., 2024; Trinh et al., 2024) with high-stake decision making tasks. One of the main limitations of reinforcement learning is to require a reward function to be able to learn an optimal policy for a given task. In some cases, constructing a reward function might be substantially more challenging than learning one, as in language models, i.e. reinforcement learning from human feedback, that requires to learn human rewards to align these models with human human values (Glaese et al., 2022; Jaech et al., 2024). To solve this problem a line of research focused on learning the reward function solely based on observing expert trajectories, and several proposed to directly learn the optimal policy from the expert demonstrations (Sammut et al., 1992; Hayes, 1994). Even further, recent work demonstrated that learning from demonstrations is substantially more sample-efficient than deep reinforcement learning (Garg et al., 2021).

While deep reinforcement learning policies are vastly utilized in manifold settings, several concerns have been raised on the robustness and adversarial weaknesses of these deep neural policies (Huang et al., 2017; Gleave et al., 2020; Korkmaz, 2022; 2023). In particular, it has been shown that deep reinforcement learning policies can be manipulated via adversarial perturbations introduced to their state observations. It has been discussed that the algorithms that target solving these issues further introduce computational hardness (Garg et al., 2020), decrease in accuracy (Bhagoji et al., 2019), induce fundamental trade-off between sensitivity and invariance (Tramèr et al., 2020), and generalization problems (Korkmaz, 2023). The non-robustness and volatilities of deep reinforcement learning limits the capabilities of policies trained in high dimensional state representation MDPs and raises safety concerns. The effects of some of these safety concerns can even be vividly observed in real time, in real life to have detrimental consequences. Therefore, in our paper we focus on analyzing robustness and seek answers to the following questions:

i. *How do the state-of-the-art algorithms that focus on learning via emulating affect the policy robustness compared to standard vanilla training algorithms?*

ii. What are the costs of learning from expert demonstrations compared to learning via exploration?

iii. Does learning without rewards in high-dimensional state representation MDPs yield learning non-robust features independent from both the MDP and algorithm?

Hence, to investigate these questions further in this paper we focus on the alignment problem and vulnerabilities of policies that can learn without a reward function within both the adversarial formulations and natural changes and make the following contributions:

- We investigate the robustness of the state-of-the-art deep neural policies that focus on learning via emulating against adversarial directions independent from both the algorithms the policies are trained with and the MDPs the policies are trained in. Our paper is the first one to focus on adversarial vulnerabilities of deep neural policies that can learn without a reward function (i.e. imitation learning and inverse reinforcement learning).
- We provide a theoretical analysis on what causes these adversarial vulnerabilities, and introduce a mathematically rigorous investigation that explains this phenomenon that learning from expert demonstrations comes with a great cost compared to learning from exploring. Learning without rewards causes sequential decision making processes to be extremely sensitive towards slight deviations from their optimal trajectories.
- We compare the robustness of standard vanilla trained policies and the policies that can learn without a reward function in high dimensional state representation MDPs. We demonstrate that vanilla trained deep reinforcement learning policies are significantly more robust and aligned compared to algorithms proposed to learn without the presence of a reward function.
- Finally, our results reveal that despite a significant decrease in performance, the inverse deep neural policy believes that it in fact received larger rewards than it did before, which demonstrates that the correlation between predicted rewards and true rewards obtained from the MDP is utterly broken, and there is a clear misalignment problem which reflecting that there is a gap between what the policy actually did and what the policy itself believed what it did.

2 Relevant Work and Background

In reinforcement learning the environment is given by a Markov decision process (MDP) $\mathcal{M} = \{S, A, \mathcal{P}, p_0, r, \gamma\}$ where S is the set of states, A is the set of actions, $\mathcal{P}(s' \mid s, a)$ is the probability of transitioning to state s' given that action a is taken in state s, p_0 is the initial state distribution, $r(s, a)$ is the reward received when taking action a in state s, and $0 < \gamma \leq 1$ is the discount factor. A policy $\pi(s, a)$ assigns a probability distribution on actions $a \in A$ to each state $s \in S$. Given a starting state distribution p_0, and transition probabilities $\mathcal{P}(\cdot \mid s_t, a_t)$, a policy π defines a probability distribution $\mathbb{P}_\pi$ on trajectories $\{s_t, a_t\}_{t\geq 0}$ where $s_0 \sim p_0$, $a_t \sim \pi(s_t, \cdot)$, and $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$. In particular, the distribution $\mathbb{P}_\pi$ satisfies $\mathbb{P}_\pi[s_0 = s] = p_0(s)$, $\mathbb{P}_\pi[a_t = a \mid s_t = s] = \pi(s, a)$, $\mathbb{P}_\pi[s_{t+1} = s' \mid s_t = s, a_t = a] = \mathcal{P}(s' \mid s, a)$. The goal in reinforcement learning is to learn a policy $\pi(s, a)$ that maximizes the expected discounted cumulative rewards $\sum_t \gamma^t \mathbb{E}_{s_t, a_t \sim \mathbb{P}_\pi} [r(s_t, a_t)]$. The occupancy measure ρ_π for a policy π is the distribution over states and actions visited when executing the policy given by $\rho_\pi(s, a) = \pi(s, a) \sum_t \gamma^t \mathbb{P}_\pi[s_t = s]$. In soft-$Q$ learning the policy is determined by learning the soft-Q function $Q(s, a)$. Given a policy π and a function $Q(s, a)$ the soft value function is given by $\mathcal{V}^\pi(s) = \mathbb{E}_{a\sim\pi(s,\cdot)}[Q(s, a) - \log \pi(s, a)]$, and the soft Bellman operator is

$$(\mathcal{T}^\pi Q)(s, a) = r(s, a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[\mathcal{V}^\pi(s', a)].$$

The soft Bellman operator is contractive and defines a unique soft-Q function satisfying the soft Bellman equation $Q = \mathcal{T}^\pi Q$. In soft-Q learning the goal is to learn a policy π which maximizes the entropy-regularized reward $\sum_t \gamma^t \mathbb{E}_{s_t, a_t\sim\mathbb{P}_\pi} [r(s_t, a_t) - \log(\pi(s_t, a_t))]$. The optimal policy is given by $\pi(s, a) = \frac{\exp Q(s,a)}{\sum_{a'} \exp Q(s,a')}$, where Q is the soft-Q function satisfying the soft Bellman equation

$$Q(s, a) = r(s, a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[\log \sum_{a'} Q(s', a')].$$

Robustness in Deep Reinforcement Learning. The first work focusing on adversarial robustness of deep reinforcement learning policies was conducted by Huang et al. (2017). In particular, the study conducted by Huang et al. (2017) utilizes the fast gradient sign method produced perturbations (Goodfellow et al., 2015) added to the state observations of the deep reinforcement learning policies. On this line of research while some studies focus on investigating more efficient ways of producing these adversarial perturbations introduced to state observations, some utilize adversarial formulations (Carlini & Wagner, 2017) to investigate frequency vulnerabilities and visualize the different patterns of non-robust features learnt by different deep reinforcement learning algorithms. Along these lines, some studies investigate the relationship between adversarial directions and natural directions intrinsic to the MDP and demonstrate that adversarial training results in worse generalization capabilities compared to vanilla training (Korkmaz, 2023). Following the research focusing on investigating vulnerabilities of deep reinforcement learning policies towards adversarial perturbations several other works focus on different strategies to make policies more robust towards these malicious perturbations. More in particular, Gleave et al. (2020); Pinto et al. (2017) propose to formulate the relationship between the adversary and the policy as a zero-sum Markov game. In some of these studies the adversary is restricted to change environment dynamics (Pinto et al., 2017), and in others the adversary is restricted to taking natural actions in the given environment (Gleave et al., 2020). Some studies focused on detection of these adversarial (i.e. non-robust) state observations by leveraging the curvature of the deep reinforcement learning manifold to make robust decisions (Korkmaz & Brown-Cohen, 2023). While several studies focused on trying to build robust deep reinforcement learning policies, quite recently several studies demonstrated that deep reinforcement learning policies learn shared adversarial features across MDPs including the state-of-the-art adversarially trained ones (Korkmaz, 2022; 2024).

Learning without Rewards. Reinforcement learning from human feedback (RLHF) which infers a reward function from human preferences, resulted in substantial progress in large language models. To ensure safety and aligned values with human preferences, currently RLHF is the main method used widely in large language models (Glaese et al., 2022; Menick et al., 2022; OpenAI, 2023; Trinh et al., 2024). Quite recent work demonstrated the connection between learning from preferences and inverse reinforcement learning, and further provided sample complexity bounds for these algorithms (Zhu et al., 2023). Inverse reinforcement learning focuses on learning a reward function from a set of expert trajectory observations. Hence, upon the construction of a reward function from observations an optimal policy can be learnt via reinforcement learning. Another line of research that centers on learning without a reward function focuses on the setting of learning a functioning policy from a given set of observed expert trajectories via emulating expert behaviour (Sammut et al., 1992). Quite recently, Garg et al. (2021) proposed to learn a single Q function from expert demonstrations to both represent the reward function and the policy. The proposal of learning a soft-Q function is currently the only algorithm that can achieve a performance level that can match deep reinforcement learning in high dimensional state representation MDPs. Furthermore, the authors argue that the fact that inferred rewards are highly correlated with the ground truth rewards shows that the proposed algorithms can also be used in inverse reinforcement learning. For this reason in the remainder of the paper we will refer to inverse reinforcement and imitation learning policies as inverse deep neural policies.

3 The Outcomes of Lack of Exploration in Learning without Rewards

In this section we fundamentally explain and theoretically motivate the results observed and reported in Section 5. Thus, this section is dedicated to explain how training via the state-of-the-art deep imitation learning algorithms leads to policies that are intrinsically non-robust. Let each state be given by a d-dimensional feature vector $s \in \mathbb{R}^d$, and for each action $a \in A$ there is a parameter vector $\theta_a \in \mathbb{R}^d$. The state-action value function is parameterized as $Q_\theta(s,a) = \langle \theta_a, s \rangle$. The inverse Q-learning objective is given by

$$\mathbb{E}_{(s,a)\sim\rho_E}\left[\phi\left(Q_\theta(s,a) - \gamma \mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}[\mathcal{V}_\theta(s')]\right)\right] - \mathbb{E}_{(s,a)\sim\mu}\left[\phi\left(\mathcal{V}_\theta(s) - \gamma \mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}[\mathcal{V}_\theta(s')]\right)\right] \quad (1)$$

where $\mathcal{V}_\theta(s) = \log \sum_a \exp Q_\theta(s,a)$, and let ρ_E be the occupancy measure of the expert policy, and μ any valid occupancy measure. We will focus on the offline setting where $\mu = \rho_E$ i.e. only samples from expert trajectories are used. Later we will discuss extensions to the online setting where μ is a mixture of expert trajectories and previously sampled states from a replay buffer. Expert trajectories will achieve higher rewards than the trajectories generated by an agent early on in the training process.

Thus, we show that this corresponds to states in expert trajectories having larger projection onto a low-dimensional subspace.

Lemma 3.1 (*Low Dimensionality of Experts in High Dimensions*)**.** *Let θ^* be the parameters of the optimal soft-Q function $Q^*(s,a) = \langle \theta_a^*, s\rangle$, and let $\mathcal{V}_{\theta^*}$ be the corresponding soft value function. Let $\kappa > 1$ and $s, s' \in \mathbb{R}^d$ be states such that $\mathcal{V}_{\theta^*}(s) > \kappa \mathcal{V}_{\theta^*}(s')$. Then there is a subspace $W \subseteq \mathbb{R}^d$ of dimension at most $|A|$ and a linear transformation $\Psi : \mathbb{R}^d \to \mathbb{R}^{|A|}$ with kernel $W^\perp$ such that $\|\Psi s\| > \frac{\kappa}{\sqrt{|A|}} \|\Psi s'\| - \log |A|$*

Proof. Let W be the span of θ_a^* for $a \in A$. Let Ψ to be the linear map defined by the matrix with rows given by θ_a^*. Clearly the kernel of Ψ is the orthogonal complement $W^\perp$. Recalling that $\mathcal{V}_{\theta^*}(s) = \log \sum_a \exp Q_{\theta^*}(s,a)$ we have

$$\mathcal{V}_{\theta^*}(s') \geq \max_{a \in A} Q_{\theta^*}(s',a) \geq \frac{1}{\sqrt{|A|}} \sqrt{\sum_{a \in A} Q_{\theta^*}(s',a)^2} = \frac{1}{\sqrt{|A|}} \|\Psi s'\|.$$

On the other hand

$$\mathcal{V}_{\theta^*}(s) \leq \max_{a \in A} Q_{\theta^*}(s,a) + \log |A| \leq \sqrt{\sum_{a \in A} Q_{\theta^*}(s,a)^2} + \log |A| = \|\Psi s\| + \log |A|.$$

Combining the two inequalities with the assumption that $\mathcal{V}_{\theta^*}(s) > \kappa \mathcal{V}_{\theta^*}(s')$ completes the proof. □

Lemma 3.1 states that for $\kappa \gg \sqrt{|A|}$, the projection onto W of s is much larger than that of s' (as measured by the transformation Ψ). Thus, the intuitive conclusion is that states with higher value (e.g. states sampled from expert trajectories) have larger projection onto a low-dimensional subspace. The inverse Q-learning algorithm updates the parameters θ by stochastic gradient descent, where the average over ρ_E in Equation 1 is approximated by sampling from stored expert trajectories. In order to qualitatively demonstrate how the use of expert trajectories leads to lower robustness we take the conclusion of Lemma 3.1 and define subspace-contained expert trajectories.

Definition 3.2 (*Subspace Contained*)**.** The set of expert trajectories ρ_E is *subspace-contained* if there exists a subspace $W \subseteq \mathbb{R}^d$ of dimension $l < d$ such that, when started in $s_0 \in W$ the expert policy π_E always transitions to states $s \in W$. Consequently, $s \in W$ for all (s,a) in the support of ρ_E.

While Lemma 3.1 proves that in general there will exist a subspace where higher value states have larger projection than lower value states, Definition 3.2 posits the existence of a potentially lower-dimensional subspace which entirely contains all the states encountered on expert trajectories due to the fact that experts have higher values than an agent who started exploring a high-dimensional environment. Now will prove that Q-values for states in the orthogonal complement of W will remain unchanged during training with inverse Q-learning.

Proposition 3.3. *Let the initial weights be $\theta^{(0)} \in \mathbb{R}^d$. Let ρ_E be subspace-contained, and let $s \in W^\perp$ be a state in the orthogonal complement of W. Let θ be the weights after any number of steps of training with inverse Q-learning. Then $Q_\theta(s,a) = Q_{\theta^{(0)}}(s,a)$ for all $a \in A$.*

Proof. We first show that the (stochastic) gradient of the cost function in Equation 1 is contained in W. For any s the gradient of the state-action value function is

$$\nabla_{\theta_a} Q_\theta(s,a) = \nabla_{\theta_a} \langle \theta_a, s\rangle = s, \quad \nabla_{\theta_a} Q_\theta(s,a') = \nabla_{\theta_a} \langle \theta_{a'}, s\rangle = 0.$$

Further, the gradient of the value function is

$$\nabla_{\theta_a} \mathcal{V}_\theta^*(s) = \nabla_{\theta_a} \log \sum_a \exp Q_\theta(s,a) = s \frac{\exp Q_\theta(s,a)}{\sum_{a'} \exp Q_\theta(s,a')}.$$

Note that all the above gradients are multiples of the input state s. Next observe that the a component gradient of the objective in Equation 1 is a linear combination of $\nabla_{\theta_a} Q_\theta(s,a)$, $\nabla_{\theta_a} Q_\theta(s,a')$, and $\nabla_{\theta_a} \mathcal{V}_\theta^*(s)$, where s always lies in the support of ρ_E. Thus, by Definition 3.2, the gradient update in each step of inverse Q-learning lies in the subspace W, as it is a linear combination of states $s \in W$. Therefore, after any number of steps the weights θ will satisfy $\theta_a - \theta_a^{(0)} \in W$. To complete the proof observe that for $s \in W^\perp$ we have $Q_\theta(s,a) - Q_{\theta^{(0)}}(s,a) = \langle \theta_a - \theta_a^{(0)}, s\rangle = 0$. □

The initial weights $\theta^{(0)}$ are usually set randomly or to some default value, and thus one does not expect $\theta^{(0)}$ to have any reasonable relationship to the optimal weights. Therefore, the intuitive consequence of Proposition 3.3 is that the soft Q-function learned from expert trajectories assigns random or arbitrary values to actions in states $s \in W^{\perp}$. In fact, Proposition 3.3 leads directly to the following corollary which shows that the rewards estimated by inverse Q-learning become uncorrelated to the true rewards.

Corollary 3.4. *Suppose each coordinate of $\theta^{(0)}$ is an independent Gaussian random variable with mean 0 and variance 1, and that the feature vectors corresponding to each state are normalized i.e. $\|s\|_2 = 1$. Then for each $s \in W^{\perp}$ the expectation over $\theta^{(0)}$ of the rewards $r_\theta(s,a)$ estimated by inverse Q-learning will be independent of the state s.*

Proof. The reward estimates for inverse reinforcement learning are obtained by assuming that the learned Q-function satisfies the soft Bellman equation i.e. that $Q_\theta(s,a) = r(s,a) + \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[\mathcal{V}^{\pi_\theta}(s')]$. Thus, given the estimated optimal state-action value function Q_θ, one solves for the rewards $r(s,a)$ in the soft Bellman equation in order to obtain the reward estimate. Hence, the reward estimated by inverse Q-learning in a state $s \in W^{\perp}$ is

$$\begin{aligned}
r_\theta(s,a) = Q_\theta(s,a) - \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[\mathcal{V}^{\pi_\theta}(s')] &= Q_\theta(s,a) - \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[\log\sum_{a'}\exp Q_\theta(s',a')] \\
&= Q_{\theta^{(0)}}(s,a) - \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[\log\sum_{a'}\exp Q_{\theta^{(0)}}(s',a')]
\end{aligned}$$

where the last line follows from Proposition 3.3. Because $\theta^{(0)}$ has independent Gaussian coordinates, its distribution is rotationally invariant i.e. $U\theta_a^{(0)}$ has the same distribution as $\theta_a^{(0)}$ for any rotation matrix U. Let U be any rotation such that $U^\top$ sends s to the first standard basis vector e_1. It is always possible to choose such a rotation because $\|s\| = 1$. Then by rotational invariance $Q_{\theta^{(0)}}(s,a) = \langle\theta_a^{(0)}, s\rangle$ has the same distribution as $\langle U\theta_a^{(0)}, s\rangle = \langle\theta_a^{(0)}, U^\top s\rangle = \langle\theta_a^{(0)}, e_1\rangle = Q_{\theta^{(0)}}(e_1,a)$. Thus the expectation of the rewards estimated by inverse Q-learning is given by

$$\begin{aligned}
&\mathbb{E}_{\theta^{(0)}\sim\mathcal{N}(0,I)}\left[r_\theta(s,a)\right] = \mathbb{E}_{\theta^{(0)}\sim\mathcal{N}(0,I)}\left[Q_{\theta^{(0)}}(s,a)\right] \\
&- \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\left[\mathbb{E}_{\theta^{(0)}\sim\mathcal{N}(0,I)}\left[\log\sum_{a'\in\mathcal{A}}\exp Q_{\theta^{(0)}}(s',a')\right]\right] \\
&= \mathbb{E}_{\theta^{(0)}\sim\mathcal{N}(0,I)}\left[Q_{\theta^{(0)}}(e_1,a)\right] - \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\left[\mathbb{E}_{\theta^{(0)}\sim\mathcal{N}(0,I)}\left[\log\sum_{a'\in\mathcal{A}}\exp Q_{\theta^{(0)}}(e_1,a')\right]\right] \\
&= \mathbb{E}_{\theta^{(0)}\sim\mathcal{N}(0,I)}\left[Q_{\theta^{(0)}}(e_1,a)\right] - \mathbb{E}_{\theta^{(0)}\sim\mathcal{N}(0,I)}\left[\log\sum_{a'\in\mathcal{A}}\exp Q_{\theta^{(0)}}(e_1,a')\right].
\end{aligned}$$

This completes the proof as the above expression for $\mathbb{E}_{\theta^{(0)}\sim\mathcal{N}(0,I)}\left[r_\theta(s,a)\right]$ does not depend on the state s. □

In general, the inverse Q-learning policy still may perform well. Indeed, if following the learned policy causes the agent to only encounter states $s \in W$, then performance in the standard setting will be unaffected by inaccuracy in states $s \in W^{\perp}$. However, the robustness of the policy may still be affected, as a slight deviation from the optimal path may cause the policy to transition into $s \in W^{\perp}$ where the Q-function is completely untrained. We now formalize the above intuition regarding the consequences of Proposition 3.3. Let π^* be the optimal soft policy and let Q_θ be the soft Q-function obtained by training with inverse Q-learning with corresponding soft policy π_θ. We assume that $\pi_\theta(s,a) = \pi^*(s,a)$ for all $s \in W$ i.e. training with inverse Q-learning has succeeded in accurately learning the optimal soft policy in W. The optimal policy π^* started at $s_0 \in S$ never transitions out of S. However, π^* receives the same expected cumulative rewards R when started in either $s_0 \in W$ or $s' \in W^{\perp}$. Taking non-optimal actions in a p fraction of states $s \in W$ and utilizing π^* in all other states results in a transition to $s' \in W^{\perp}$. There are no transitions from $s' \in W^{\perp}$ to $s \in W$. We now show that under these assumptions, slight deviations from the optimal policy have no impact on the optimal policy π^*, but cause the inverse deep neural policy π_θ to perform at the same level as an untrained policy.

Proposition 3.5. *Taking non-optimal actions in a p fraction of states does not change the reward received by the optimal policy π^*, but causes the inverse reinforcement learning policy π_θ to receive the same rewards received by $\pi_{\theta^{(0)}}$.*

Proof. After taking non-optimal actions in a p fraction of states, the optimal policy π^* transitions to $s' \in W^\perp$, and by assumption never transitions out. However, π^* receives the same rewards after transitioning to $s' \in W^\perp$ as it would have from remaining in W. The policy π_θ is equal to π^* on W. Thus, taking non-optimal actions in a p fraction of states causes a transition to $W^\perp$. However, by Proposition 3.3 $\pi_\theta(s', a) = \pi_{\theta^{(0)}}(s', a)$ for all $s \in W^\perp$. By assumption there are no transitions out of $W^\perp$, so π_θ receives the same rewards as $\pi_{\theta^{(0)}}$. $\square$

4 Probing the Inverse Reinforcement Learning Manifold via Adversarial Directions

Section 3 is dedicated to provide theoretical analysis on the causes of the adversarial vulnerabilities of inverse deep neural policies observed and reported in Section 4 and 5. Hence, in this section we describe the methods used to cause deviations from the optimal trajectory in order to understand the robustness of policies. To achieve this there are two main approaches we will take, one is based on moving along the adversarial directions in the deep neural policy manifold, and the second is a direct slight push to the policy from its optimal course of trajectory. These are described below in more detail. For the adversarial directions we utilize the methodology described in Korkmaz (2022).

Definition 4.1 (*Algorithm and MDP independent adversarial direction $\mathcal{A}^{random}_{alg+\mathcal{M}}$*)**.** Given a random state $s(\mathcal{M})$ sampled from a random episode of e of an MDP $\mathcal{M}$ from a policy $\pi(s(\mathcal{M}), \cdot)$, the minimum length adversarial direction $\boldsymbol{v}(s(\mathcal{M}), \pi(s(\mathcal{M}), \cdot))$ is computed satisfying, $\arg\max_a \pi(s(\mathcal{M}), a) \neq \arg\max_a \pi(s(\mathcal{M}) + \boldsymbol{v}(s(\mathcal{M}), \pi(s(\mathcal{M}), \cdot)), a)$. The computed adversarial direction $\boldsymbol{v}(s(\mathcal{M}), \pi(s(\mathcal{M}), \cdot))$ norm-bounded by $\kappa > 0$ is added to the visited states of another policy $\pi'(s(\mathcal{M}'), \cdot))$ trained with a completely different algorithm, in a distinct MDP $\mathcal{M}'$. Hence, the state obtained moving along the adversarial direction is

$$s_{\boldsymbol{v}} = s(\mathcal{M}') + \kappa \frac{\boldsymbol{v}(s(\mathcal{M}), \pi(s(\mathcal{M}), \cdot))}{\|\boldsymbol{v}(s(\mathcal{M}), \pi(s(\mathcal{M}), \cdot))\|}.$$

Note that if $\mathcal{M}_{\text{alg}} \neq \mathcal{M}'_{\text{alg}}$ this means that the policies are trained with the same algorithm but in different MDPs; thus, the adversarial direction is computed from $\mathcal{M}$ and transferred to a distinct MDP $\mathcal{M}'$. The setting of $\mathcal{M}_{\text{alg}} \neq \mathcal{M}'_{\text{alg}}$ will be referred as $\mathcal{A}^{\text{random}}_{\mathcal{M}}$.

Definition 4.2 (*The δ-deviation from the optimal trajectory*)**.** For a policy $\pi(s, a)$ and state s let $a_w(s) = \arg\min_a Q(s, a)$ denote the worst action in state s. The notation $\mathcal{A}_{a_w}$ refers to a setting in which in state s the policy is set to take action $a_w(s)$, rather than the optimal action selected by the policy $\pi(s, a)$ for a δ-fraction of the visited states where delta is $\delta \ll 1$. The notation $\mathcal{A}_{\text{random}}$ refers to the setting in which the policy $\pi(s, a)$ is set to take an action uniformly at random $a \sim \mathcal{U}_A$ in s for a δ-fraction of the visited states where delta is $\delta \ll 1$.

Both Definition 4.1 and Definition 4.2 will be used in Section 5 to lay out precise non-robustness of inverse deep neural policies and their comparison to vanilla trained deep reinforcement learning.

Experts and Alignment: Note that the initial and foundational objective of inverse reinforcement learning on inferring a reward function from observed trajectories opened a new channel on the value alignment problem (Ng & Russell, 2000; Russell, 1998). More precisely, the current unethical behaviour observed in large language models that are trained with RLHF (i.e. learning a reward function from human preferences to align language agents with human values) is one of the concrete substantial safety concerns that is tightly connected to what we highlight in our paper. Section 5 demonstrates that policies that can explore, i.e. standard deep reinforcement learning, are more robust and aligned compared to algorithms that are specifically focused on resolving the value-alignment problem. In particular, the fact that the policy's perception on the environmental (i.e. ground truth) rewards is broken as reported in Section 5.1 further demonstrates that the value-alignment problem is far beyond being solved, and in fact most critically, the claimed alignment is extremely fragile.

Table 1: The performance drop results with algorithm and MDP independent adversarial direction $\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$ and MDP independent adversarial direction $\mathcal{A}^{\text{random}}_{\mathcal{M}}$ for vanilla trained deep reinforcement learning policies and inverse deep neural policies in Seaquest.

Training Method	Adversarial Setting	RoadRunner	BankHeist	TimePilot
Deep Inverse-Q Learning	$\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$	0.93439±0.0075	0.9882±0.03457	0.95037±0.01454
Vanilla Trained	$\mathcal{A}^{\text{random}}_{\mathcal{M}}$	0.28496±0.09693	0.18527±0.14817	0.46938±0.08672
Training Method	Adversarial Setting	JamesBond	CrazyClimber	Gaussian
Deep Inverse-Q Learning	$\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$	0.916947±0.01371	0.76282±0.02146	0.05277±0.0821
Vanilla Trained	$\mathcal{A}^{\text{random}}_{\mathcal{M}}$	0.308188±0.12931	0.180897±0.13982	0.05536±0.12091

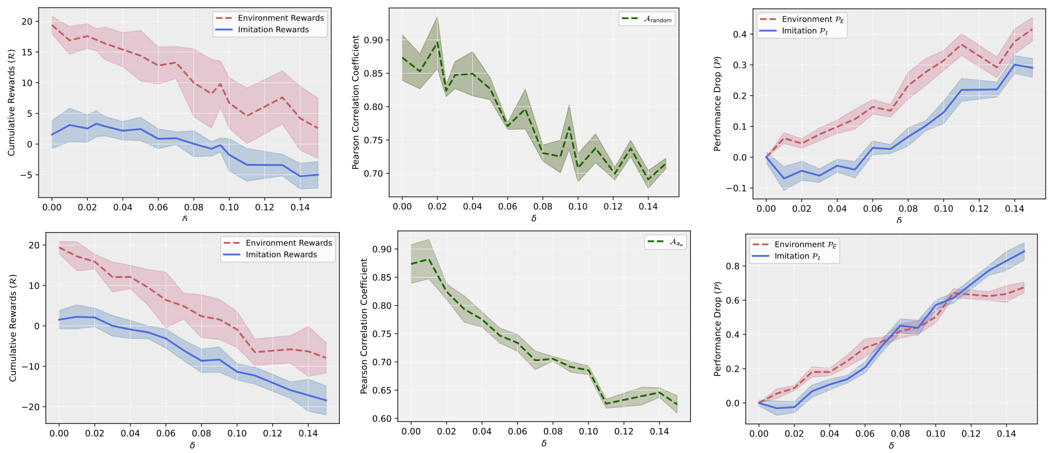
5 The Cost of Learning from Expert Demonstrations

In this paper the straightforward vanilla trained deep reinforcement learning policies are trained with Deep Double Q-Network (DDQN) initially proposed by Hasselt et al. (2016) with the architecture introduced in Wang et al. (2016). The state-of-the-art imitation and inverse reinforcement learning policy is trained via the inverse Q-learning algorithm described in Section 2. The experiments are conducted in the Arcade Learning Environment (ALE) (Bellemare et al., 2013) with OpenAI wrappers (Brockman et al., 2016). See supplementary material for further details. The results are averaged over 10 runs and the standard error of the mean is included in all the tables and figures presented in the paper. The normalized performance drop of the policies is computed as $\mathcal{P} = (\text{Score}_{\text{max}} - \text{Score}_{\text{set}})/(\text{Score}_{\text{max}} - \text{Score}_{\text{min}})$. Here $\text{Score}_{\text{max}}$ is the score obtained by the baseline trained policy following the learned policy with a clean run in a given environment, $\text{Score}_{\text{set}}$ is the score obtained by the policy in the test time, and $\text{Score}_{\text{min}}$ is the score obtained by the trained policy when the policy chooses the worst possible action in each state. Note that $\mathcal{A}_{\text{base}}$ refers to the unmodified run of the policy in an unmodified MDP.

As described in detail in Section 2 the inverse Q-learning algorithm learns both an optimal policy and a reward function from observed trajectories; thus, the fact that inverse-Q learning simultaneously learns both a reward function and an optimal policy is the reason that throughout the paper imitation learning and inverse reinforcement learning will be used interchangeably. Note that the inverse Q-learning algorithm is the only algorithm that can achieve equivalent performance with standard deep reinforcement learning policies in MDPs with high-dimensional observations. Table 2 and Table 1 report the performance drop results with the environment and algorithm independent adversarial direction $\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$ and the environment independent adversarial direction $\mathcal{A}^{\text{random}}_{\mathcal{M}}$ for vanilla trained deep reinforcement learning policies and inverse deep neural policies in Arcade Learning Environment (ALE). Recall that $\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$ represents the adversarial setting in which the adversarial direction is computed from a completely different MDP and from a policy trained with a completely different algorithm. Note that in these experiments the ℓ_2-norm bound κ level is set to the magnitude where simple Gaussian noise with the ℓ_2-norm κ has insignificant effect on the policy performance.

5.1 The Catastrophic Results of Small Divergence from the Optimal Policy

In this section we investigate effects of small deviations from the optimal policy followed by the inverse deep neural policy. This is achieved by $\mathcal{A}_{a_w}$ with $a_w = \arg\min_{a' \in \mathcal{A}} Q(s, a')$ for a random small fraction of visited states in a given episode, the $\mathcal{A}_{\text{random}}$ setting explained in Definition 4.2, and the $\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$ setting explained in Definition 4.1. Either moving along adversarial directions in the deep neural policy manifold or slight alterations in the actions taken by the policy causes slight deviation from the optimal course of the inverse deep neural policy. Figure 1 reports the true rewards obtained from the environment and the reward predictions of inverse Q-learning, the Pearson correlation coefficient between reward predictions and environment rewards, and the performance drop computed from environment rewards and the inverse Q-learning predictions with respect to δ-deviation from optimal trajectory with $\mathcal{A}_{a_w}$ and $\mathcal{A}_{\text{random}}$. Thus, the results in Figure 1 demonstrate the outcome of slight deviation of the optimal trajectory and its effects on the reward predictions and the true rewards obtained from the environment. The fact that small deviations from the optimal policy result in **significant decrease in both the rewards obtained, and the accuracy of the predicted rewards** for the inverse deep neural policy, provide empirical verification of Corollary 3.4. In particular, this lends credence to the claim that the lower exploration of state space and being limited to expert



Cumulative Rewards | Pearson Correlation Coefficient | Performance Drop

Figure 1: The effects of δ-deviation from the optimal trajectory with $\mathcal{A}_{\text{random}}$ and $\mathcal{A}_{a_w}$. Left: Cumulative rewards obtained from the environment, and inverse Q-learning reward predictions. Center: Pearson correlation coefficient. Right: Performance drop based on true environment rewards and inverse Q-learning reward predictions. Up: δ-deviation $\mathcal{A}_{\text{random}}$. Down: δ-deviation $\mathcal{A}_{a_w}$.

Table 2: The performance drop results with algorithm and MDP independent adversarial direction $\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$ and MDP independent adversarial direction $\mathcal{A}^{\text{random}}_{\mathcal{M}}$ for vanilla trained deep reinforcement learning policy and inverse deep neural policy in Pong.

Training Method	Adversarial Setting	RoadRunner	BankHeist	TimePilot
Deep Inverse-Q Learning	$\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$	1.0±0.0	0.9153±0.01128	0.9880±0.0112
Vanilla Trained	$\mathcal{A}^{\text{random}}_{\mathcal{M}}$	0.0819±0.0500	0.0265±0.01284	0.2963±0.0573
Training Method	Adversarial Setting	JamesBond	CrazyClimber	Gaussian
Deep Inverse-Q Learning	$\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$	1.0±0.0	0.9857±0.006	0.04285±0.01572
Vanilla Trained	$\mathcal{A}^{\text{random}}_{\mathcal{M}}$	-0.0024±0.0037	0.06024±0.02970	0.0451±0.0182

trajectories cause the inverse deep neural policies to overfit to the expert's beliefs and experience significant performance loss under subtle departures from the policy's optimal course.

Notably, when δ is initially increased from zero, the performance drop of the inverse Q-learning reward predictions is negative. That is, despite a **significant decrease in performance** caused by the deviation, the **inverse deep neural policy believes it will actually receive larger rewards** than it did before. In fact, as the true environment rewards decrease, their Pearson correlation with the predicted rewards also decreases, indicating that in general small deviations cause the policy to form an inaccurate view of the rewards it will obtain. The cost of learning from experts instead of exploring is revealed in Table 2 where the foundational reason of this vulnerability is that experts in fact limit the ability of an agent to develop more comprehensive knowledge of the MDP as shown in Figure 1.

Black-Box Adversaries. The results reported in Table 2 and 1 are for an adversarial direction that is computed from a vanilla trained policy in one MDP $\mathcal{M}$, and added to the observations in a different MDP $\mathcal{M}'$ of the inverse Q-learning policy and the vanilla trained policy respectively. This corresponds to the $\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$ setting for the inverse Q-learning policy, and $\mathcal{A}^{\text{random}}_{\mathcal{M}}$ for the vanilla policy. Of particular significance is the fact that, even though the perturbation is specifically computed from a vanilla trained policy and for a vanilla trained policy, the impact of the adversarial direction is much larger on the inverse-Q learning policy than on the vanilla policy. Thus, the results in Table 2 and 1 demonstrate that the inverse Q-learning policies are more susceptible to the shared adversarial directions across both MDPs and algorithms compared to vanilla trained reinforcement learning policies. These results demonstrate that not only are inverse deep neural policies are vulnerable, they can be attacked via black-box adversarial attacks.

Breaking the Link Between Imitation and Inverse Reinforcement Learning. Table 3 shows the Pearson and Spearman correlation coefficients between cumulative rewards obtained from the environment and the cumulative reward prediction of the deep imitation learning policy for the algorithm and MDP independent adversarial direction $\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$, δ-deviation $\mathcal{A}_{a_w}$, and δ-deviation $\mathcal{A}_{\text{random}}$ in Seaquest, BeamRider and Breakout MDPs with δ-deviation from the optimal trajectory

Table 3: Pearson and Spearman correlation coefficient between true cumulative rewards obtained from the environment and cumulative reward prediction made by the state-of-the-art inverse deep neural policy for a baseline run $\mathcal{A}_{\text{base}}$, with $\mathcal{A}_{a_w}$, δ-deviation $\mathcal{A}_{\text{random}}$, and with $\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$ where $\delta = 0.003$.

Seaquest	$\mathcal{A}_{\text{base}}$	δ-deviation $\mathcal{A}_{a_w}$	δ-deviation $\mathcal{A}_{\text{random}}$	$\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$
Pearson	0.857880±0.025528	-0.44787±0.140356	0.324616±0.22191	0.202436±0.24088
Spearman	0.688300±0.1257206	-0.35129±0.20131	0.023051±0.12057	0.023701±0.20764
BeamRider	$\mathcal{A}_{\text{base}}$	δ-deviation $\mathcal{A}_{a_w}$	δ-deviation $\mathcal{A}_{\text{random}}$	$\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$
Pearson	0.654375±0.005274	-0.27414±0.026496	-0.10209±0.076554	-0.22879±0.199385
Spearman	0.688300±0.1257206	-0.34725±0.022020	-0.11846±0.092472	-0.43719±0.115118
Breakout	$\mathcal{A}_{\text{base}}$	δ-deviation $\mathcal{A}_{a_w}$	δ-deviation $\mathcal{A}_{\text{random}}$	$\mathcal{A}^{\text{random}}_{\text{alg}+\mathcal{M}}$
Pearson	0.850592±0.013398	-0.12104±0.083332	0.248250±0.046503	-0.09089±0.092005
Spearman	0.7665362±0.04679	-0.13360±0.030187	0.243274±0.012255	-0.25481±0.089637

setting with $\delta = 0.003$. The results in Table 3 demonstrate that even a slight change in the trajectory (i.e. a change in the trajectory with frequency of 0.003) is more than enough to break the correlation between true rewards obtained from the environment and the reward predictions of the inverse deep neural policy. Hence, without the correlation between true rewards and reward predictions the decisions of the policy will be random, i.e. untrained.

Misalignment with the Intended Objective: These results further verify the theoretical predictions of Corollary 3.4, where the estimated rewards for states that deviate subtly from expert trajectories are uncorrelated with their true rewards. Inverse deep neural policy is designed so that there is a tight coupling between the trained policy and the estimated rewards. In particular, the rewards are derived in such a way that the policy is optimal for those rewards. Thus, decoupling of the estimated rewards and environment rewards implies that the learned policy is optimized for a very different objective than intended. The fact that subtle deviations from the optimal trajectory break the beliefs of the inverse deep neural policy on the MDP rewards, raises significant questions regarding misalignment.

MDP Rewards and Agent's Beliefs: The agent's beliefs on the MDP rewards define what the task is and how the task should be solved. The fact that the deep imitation learning policy's beliefs experience an extreme shift under slight deflections from its optimal path is evidence that the policy has a completely different vision on what the objective is and how the task should be completed (i.e. the misalignment problem) (Wiener, 1960; Good, 1965). While these results may raise questions and concerns on the safety and AI-alignment of the deep imitation and deep inverse reinforcement learning policies respectively, one of our main objectives is to layout the exact fundamental trade-off made with learning from expert demonstrations instead of exploring the MDP.

6 Conclusion

In this paper we study the vulnerabilities of policies trained without rewards in high-dimensional state representation MDPs, and seek answers for the following questions: *(i) Does learning without a reward function in complex state representation MDPs cause learning non-robust features independent from the MDP it is trained in and the algorithms it is trained with? (ii) How is the policy robustness affected by the state-of-the-art algorithms that can learn without rewards compared to vanilla trained deep reinforcement learning? (iii) What is the cost of learning from expert demonstrations instead of learning purely from exploration?* To answer these questions we first provide a theoretical analysis that shows that learning from experts instead of from exploration comes with a cost. Moving along the adversarial directions independent from both the MDP and algorithm in the neural policy manifold, we demonstrate that vanilla trained deep reinforcement learning policies are more robust compared to the state-of-the-art algorithms that can learn without rewards. We further demonstrate that even though there is a significant decrease in performance, the inverse deep neural policy believes that it in fact received larger rewards than it did before, which demonstrates that the correlation between predicted rewards and true rewards obtained from the MDP is completely broken. These results provide a clear demonstration of the violation of alignment where there is a gap between what the policy actually did and what the policy itself believed that it did. Our work highlights that there is a misalignment problem and the algorithms proposed to solve this problem are learning extremely fragile policies with misaligned values.

REFERENCES

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research.*, pp. 253–279, 2013.

Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. Lower bounds on adversarial robustness from optimal transport. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pp. 7496–7508, 2019.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv:1606.01540*, 2016.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *In 2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017.

Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *Neural Information Processing Systems (NeurIPS) [Spotlight Presentation]*, 2021.

Sanjam Garg, Somesh Jha, Saeed Mahloujifar, and Mohammad Mahmoody. Adversarially robust learning could leverage computational hardness. In *Algorithmic Learning Theory, ALT 2020, 8-11 February 2020, San Diego, CA, USA*, volume 117 of *Proceedings of Machine Learning Research*, pp. 364–385. PMLR, 2020.

Amelia Glaese, Nat McAleese, Maja Trebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin J. 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 Mokrá, 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. *CoRR*, 2022.

Adam Gleave, Michael Dennis, Cody Wild, Kant Neel, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. *International Conference on Learning Representations ICLR*, 2020.

Irving John Good. Speculations concerning the first ultraintelligent machine. In *In Advances in Computers*. Cambridge MA: Academic Press, 1965.

Ian Goodfellow, Jonathan Shelens, and Christian Szegedy. Explaning and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

Mason Hargrave, Alex Spaeth, and Logan Grosenick. Epicare: A reinforcement learning benchmark for dynamic treatment regimes. In *Conference on Neural Information Processing Systems, NeurIPS*, 2024.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *In Thirtieth AAAI conference on artificial intelligence*, 2016.

Gillian Hayes. A controller using learning by imitation. *In Proceedings Second International Symposium on Intelligent Systems*, 1994.

Sandy Huang, Nicholas Papernot, Yan Goodfellow, Ian an Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *Workshop Track of the 5th International Conference on Learning Representations*, 2017.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, and Ignasi Clavera Gilaberte and. Openai o1 system card. *CoRR*, 2024.

Steven Kapturowski, Victor Campos, Ray Jiang, Nemanja Rakicevic, Hado van Hasselt, Charles Blundell, and Adrià Puigdomènech Badia. Human-level atari 200x faster. 2023.

Ezgi Korkmaz. Deep reinforcement learning policies learn shared adversarial features across MDPs. *AAAI Conference on Artificial Intelligence*, 2022.

Ezgi Korkmaz. Adversarial robust deep reinforcement learning requires redefining robustness. *AAAI Conference on Artificial Intelligence*, 2023.

Ezgi Korkmaz. Understanding and diagnosing deep reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning ICML*, Proceedings of Machine Learning Research (PMLR). PMLR, 2024.

Ezgi Korkmaz and Jonah Brown-Cohen. Detecting adversarial directions in deep reinforcement learning to make robust decisions. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17534–17543. PMLR, 2023.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, H. Francis Song, Martin J. Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. Teaching language models to support answers with verified quotes. *CoRR*, abs/2203.11147, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, arc G Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518: 529–533, 2015.

Volodymyr Mnih, Adria Badia Puigdomenech, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *In International Conference on Machine Learning*, pp. 1928–1937, 2016.

Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In Pat Langley (ed.), *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pp. 663–670, 2000.

OpenAI. Gpt-4 technical report. *CoRR*, 2023.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. *International Conference on Learning Representations ICLR*, 2017.

Stuart Russell. Learning agents for uncertain environments (extended abstract). In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, Madison, Wisconsin, USA, July 24-26, 1998*, pp. 101–103. ACM, 1998.

Claude Sammut, Scott Hurst, Dana Kedzier, and Donald Michie. Learning to fly. *In Proceedings of the Ninth Conference on Machine Learning*, 1992.

Florian Tramèr, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Jörn-Henrik Jacobsen. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9561–9571. PMLR, 2020.

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 2024.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando. De Freitas. Dueling network architectures for deep reinforcement learning. *Internation Conference on Machine Learning ICML.*, pp. 1995–2003, 2016.

Norbet Wiener. Some moral and technical consequences of automation. In *Science*, pp. 1355–1358, 1960.

Banghua Zhu, Michael I. Jordan, and Jiantao Jiao. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 43037–43067. PMLR, 2023.