Adversarial Learning of Distributional Reinforcement Learning

Yang Sui [1] Yukun Huang [1] Hongtu Zhu [2] Fan Zhou [1]

Abstract

Reinforcement learning (RL) has made significant advancements in artificial intelligence. However, its real-world applications are limited due to differences between simulated environments and the actual world. Consequently, it is crucial to systematically analyze how each component of the RL system can affect the final model performance. In this study, we propose an adversarial learning framework for distributional reinforcement learning, which adopts the concept of influence measure from the statistics community. This framework enables us to detect performance loss caused by either the internal policy structure or the external state observation. The proposed influence measure is based on information geometry and has desirable properties of invariance. We demonstrate that the influence measure is useful for three diagnostic tasks: identifying fragile states in trajectories, determining the instability of the policy architecture, and pinpointing anomalously sensitive policy parameters.

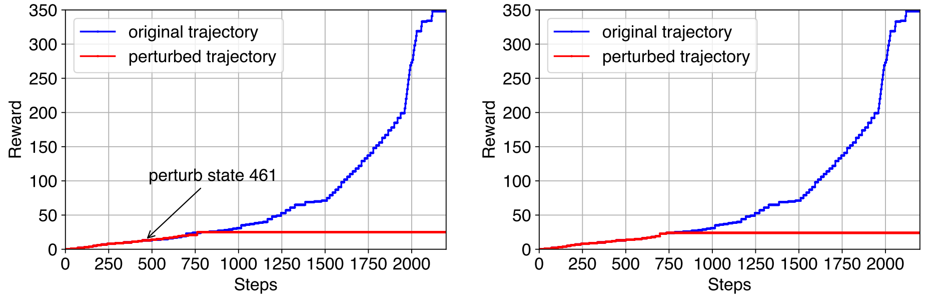
1. Introduction

Reinforcement learning (RL) has achieved great success in various artificial intelligence areas, such as video games (Lample & Chaplot, 2017; Li, 2017), large-scale strategy games (Silver et al., 2016; 2017), robot manipulation (Kober et al., 2013; Nguyen & La, 2019), and behavioral learning in social scenarios (Baker et al., 2019). Despite the advantages of RL in the virtual world, applying RL to real-world problems is challenging. First, offline RL methods usually lack a clear and understandable process for generating various design choices, from model architecture to algorithmic hyperparameters (Kumar et al., 2021). In addition, key features of potentially realistic applications such as partial observability, different action spaces, non-stationarity, and stochasticity make the practical applicability of offline RL algorithms difficult to assess (Gulcehre et al., 2020). Moreover, offline RL methods may often suffer from some reproducibility problems (Fujimoto et al., 2019; Peng et al., 2019). These issues create a gap between the offline data used to train the policy and the online environment where the policy will be applied, limiting the efficiency of RL in solving real-world problems. Efforts have been made to adapt the offline policy to the online environment through model, value learning, or importance sampling, see (Precup, 2000; Thomas et al., 2015; Jiang & Li, 2016; Kumar et al., 2021; Gulcehre et al., 2020), but the gap remains, and its detection in practice is not yet solved.

In industry, policies are trained offline or in simulators and then applied to the real world (He & Shin, 2019; Liang et al., 2021; Qin et al., 2020; Tang et al., 2021). Although offline simulators and online environments may appear to be almost the same, these offline-trained policies can perform poorly in the real world and the resulting decision trajectories can be quite different. For example, the order-dispatching policy of ride-sharing companies is usually trained using an offline simulator which is then applied to the real world afterwards (Xu et al., 2018; Tang et al., 2019; Zhou et al., 2021a;b). As the online environment keeps changing over time, there always exists a gap between the simulated and the real platforms which makes the trained policy perform poorly in practice. In particular, the same policy can make an undesirable decision when encountering a similar but slightly different state which has never been seen during the offline training. In addition, subtle changes in a specific parameter of the policy network can also lead to abnormal results. To further illustrate this issue, we carry out some empirical studies using the Atari 2600 platform. As Figure 1(a) shows, a small perturbation imposed on some certain state can significantly change the trajectory afterwards in the *Breakout* environment. Similarly, there is a huge gap between the re-generated trajectory and the original trajectory after applying a perturbation to some parameters in the policy network, see Figure 1(b). This suggests that for the entire RL system, small variations of many local components can lead to the performance difference between the online and offline situations. Unfortunately, there is currently no

[1] School of Statistics and Management, Shanghai University of Finance and Economics, Shanghai, China [2] Departments of Biostatistics, Statistics, Computer Science, and Genetics, The University of North Carolina at Chapel Hill, Chapel Hill, USA. Correspondence to: Fan Zhou <zhoufan@mail.shufe.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).



(a) perturb state

(b) perturb parameter

Figure 1. Comparison of trajectories before and after a small perturbation, (a) the resulting trajectory by slightly perturbing state 461, (b) the resulting trajectory by slightly perturbing some particular parameter in the policy network.

way to quantify the effects of the system variations on the model performance. Therefore, the main goal of this paper is to systematically develop a simple but general adversarial learning framework for RL that can be used to detect small perturbations of each component in the whole RL system and measure their effects on the model performance.

In this paper, we propose a general framework for adversarial learning that quantitatively measures the vulnerability of each component in a RL system, such as the parameters of the policy network and state observations, to small perturbations. This framework serves as a detection tool to pinpoint the specific parts of the RL system that are negatively impacting performance when small variations are imposed. Some of the key tasks that this framework can assist with include: detecting fragile states in trajectories generated by trained policies, determining the unstable parts of the policy network, and identifying anomalously sensitive model parameters. By focusing on these particularly vulnerable states, we can create adversarial examples or enhance the policy with data augmentation to improve the robustness of RL algorithms. Additionally, this framework can provide guidance for modifying the network architecture.

Specifically, we construct a perturbation manifold for any possible perturbation together with the associated geometric quantities, and then compute the influence measure of the perturbation on a given objective function of interest on this perturbation manifold. Our influence measure quantifies the degree of local influence of the perturbation to the objective function, and thus reflects the adversarial strength of each RL component to a subtle perturbation. Notably, our influence measure stands out from common influence measures, such as the change of the objective function after being perturbed, by possessing an intrinsic property that is entirely free of the constraints imposed by the perturbation. The whole framework we describe in this work is in the context of distributional reinforcement learning (DRL) but everything can be easily extended to other RL methods including the value-based and policy-based algorithms. We conduct numerous empirical studies to demonstrate the validity of our method. The most sensitive parts of the entire system detected by our method can be evaluated by comparing the trajectories with and without the small perturbation. The main contributions of this work are summarized as follows.

- To the best of our knowledge, this is the first systematic analysis of the reasons why a trained policy may fail when applied to a new but similar environment.
- We construct a framework for adversarial learning in order to evaluate the sensitivity of each component of the RL system, taking into account the impact of small perturbations on the overall system.
- We demonstrate that the proposed method is an efficient tool for detecting DRL systems, both from a theoretical and an empirical perspective.

2. Background

In the classical RL setting, an agent interacts with an environment via a standard Markov decision process (MDP), a five-tuple $(\mathcal{S}, \mathcal{A}, R, P, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the environment transition dynamics from state $\boldsymbol{s}$ to next state $\boldsymbol{s}'$ after taking action $\boldsymbol{a}$, and $\gamma \in (0, 1)$ is the discount factor.

From expectation to distribution. A stationary policy $\pi(\cdot|\boldsymbol{s})$ maps state $\boldsymbol{s}$ to a distribution over the action space $\mathcal{A}$. Given a policy π, the discounted sum of future rewards $\boldsymbol{Z}^{\pi}(\boldsymbol{s}, \boldsymbol{a}) = \sum_{t=0}^{\infty} \gamma^t R(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is a random variable along the agent's trajectory of interactions with the environment, where $\boldsymbol{s}_0 = \boldsymbol{s}$, $\boldsymbol{a}_0 = \boldsymbol{a}$, $\boldsymbol{s}_{t+1} \sim P(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_t)$, and $\boldsymbol{a}_t \sim \pi(\cdot|\boldsymbol{s}_t)$. Classic value-based RL methods usually focus on the state-value function $Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{E}[\boldsymbol{Z}^{\pi}(\boldsymbol{s}, \boldsymbol{a})]$, which is the expectation of $Z^{\pi}(\boldsymbol{s}, \boldsymbol{a})$. By contrast, DRL directly estimates the whole return distribution.

Distributional Bellman Operator. In expectation-based RL, the Q-function is updated via the Bellman operator

$$\mathcal{T}^{\pi} Q(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{E}[R(\boldsymbol{s}, \boldsymbol{a})] + \gamma \boldsymbol{E}_{\boldsymbol{s}' \sim p, \pi}[Q(\boldsymbol{s}', \boldsymbol{a}')].$$

Similarly, $\boldsymbol{Z}^{\pi}(\boldsymbol{s}, \boldsymbol{a})$ can be updated via the distributional Bellman operator for DRL,

$$\mathfrak{T}^{\pi} \boldsymbol{Z}(\boldsymbol{s}, \boldsymbol{a}) = R(\boldsymbol{s}, \boldsymbol{a}) + \gamma \boldsymbol{Z}(\boldsymbol{s}', \boldsymbol{a}'), \tag{1}$$

where $\boldsymbol{s}' \sim P(\cdot|\boldsymbol{s}, \boldsymbol{a})$ and $\boldsymbol{a}' \sim \pi(\cdot|\boldsymbol{s}')$. The distributional Bellman operator $\mathfrak{T}^{\pi}$ is contractive under certain distribution divergence metrics. There are two main categories of DRL algorithms relying on parametric approximations. One is the Categorical distributional RL (CDRL, Bellemare et al., 2017) which represents the return distribution $\boldsymbol{Z}$ with a categorical form. The other is the Quantile distributional RL (QDRL, Dabney et al., 2018; Zhou et al., 2020; 2021c) that represents the return distribution with a mixture of Diracs.

3. Case Study: State Perturbation in C51

We begin with a naive example to show how to build the influence measure for DRL and how the measure can be used to detect fragile states. We follow the idea of the classic CDRL approach C51 (Bellemare et al., 2017) and represent the return distribution $\boldsymbol{Z}$ with a categorical form $\boldsymbol{Z}(\boldsymbol{s}, \boldsymbol{a}) = \sum_{i=1}^{N} p_i(\boldsymbol{s}, \boldsymbol{a}) \delta_{\boldsymbol{z}_i}$, where $\delta_{\boldsymbol{z}}$ denotes the Dirac distribution at $\boldsymbol{z}$. The locations $\{\boldsymbol{z}_i = V_{\min} + i(V_{\max} - V_{\min})/(N-1) : 0 \leq i < N\}$ are evenly spaced, and $N = 51$ is a common choice. The parameters of the distribution are the probabilities p_i, represented as logits, associated with each location $\boldsymbol{z}_i$. The atom probabilities are determined by a parameter model $\boldsymbol{\theta} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^N$, $\boldsymbol{Z}_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{z}_i$.

To quantify the adversarial strength of the state, we use the first-order influence measure (**FI**), which essentially portrays the degree to which the objective function is affected by the perturbation. We take the Q-function: $f(\boldsymbol{\omega}) = \boldsymbol{E}[\boldsymbol{Z}] = \sum_{i=0}^{N-1} \boldsymbol{z}_i P(\boldsymbol{z}_i | \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta}, \boldsymbol{\omega})$, which is the expectation of $\boldsymbol{Z}$ as the objective function since the Q-function captures all future information but only depends on the current state $\boldsymbol{s}$ which is perturbed.

We provide a comprehensive discussion regarding the estimation error of the Q-function and its impact on our **FI** analysis framework. While errors in Q-function may arise, they do not undermine the significance of our method. Firstly, it is important to note that the Q-function serves as an illustrative example in our experiments. The **FI** method, however, is versatile and can effectively handle various functions of interest within the RL system. Secondly, the estimation error in Q-function has negligible influence on our analysis due to our use of actual Q-estimates during training, rather than relying solely on theoretical Q-values. Consequently, when evaluating the impact of perturbations on the trajectory, we incorporate the actual estimated Q-values. Thus, the potential error in the estimated Q-values does not affect our perturbation analysis. In fact, the primary objective of our method is to identify the components accountable for the RL system's underperformance, making it well-suited for the case we are addressing.

The **FI** for the state perturbation $\boldsymbol{\Delta s}$, where $\boldsymbol{\Delta s}_0 = \boldsymbol{0}$, can be computed as

$$\mathbf{FI}_{\boldsymbol{\Delta s}}(\boldsymbol{\Delta s}_0) = \boldsymbol{\nabla}^T_{f(\boldsymbol{\Delta s}_0)} \boldsymbol{G}^{-1}(\boldsymbol{\Delta s}_0) \boldsymbol{\nabla}_{f(\boldsymbol{\Delta s}_0)}, \tag{2}$$

where $\boldsymbol{\nabla}_{f(\boldsymbol{\Delta s})}$ and $\boldsymbol{G}(\boldsymbol{\Delta s})$ have the following forms, respectively,

$$\boldsymbol{\nabla}_{f(\boldsymbol{\Delta s})} = \frac{\partial f(\boldsymbol{\Delta s})}{\partial \boldsymbol{s}} = \sum_{i=0}^{N-1} \frac{\boldsymbol{z}_i \partial \log(p_i(\boldsymbol{\Delta s}))/\partial \boldsymbol{s}}{p_i(\boldsymbol{\Delta s})}, \tag{3}$$

$$\boldsymbol{G}(\boldsymbol{\Delta s}) = \sum_{i=0}^{N-1} p_i(\boldsymbol{\Delta s}) \frac{\partial^T \log p_i(\boldsymbol{\Delta s})}{\partial \boldsymbol{s}} \frac{\partial \log p_i(\boldsymbol{\Delta s})}{\partial \boldsymbol{s}}, \tag{4}$$

with $p_i = P(\boldsymbol{z}_i | \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta}, \boldsymbol{\Delta s})$. In practice, the gradient term $\partial \log(p_i(\boldsymbol{\Delta s}))/\partial \boldsymbol{s}$ can be easily computed via backpropagation (Abadi et al., 2016; Paszke et al., 2017).

FI indicates the influence level of a small local perturbation on the overall model performance, i.e. the Q-function in this case. A higher **FI** implies that the corresponding state has a greater effect on the Q-value after the perturbation, resulting in a more significant change of the trajectory to a greater extent. A lower **FI** indicates that the corresponding state is less sensitive to the imposed perturbation and does little damage to the overall RL system.

According to (2), (3) and (4), we compute **FI** for all states on an observed trajectory and perturb the high-**FI** states with the original policy being fixed. We record the change of Q-values and rebuild part of the trajectory starting from the state being perturbed. We find that perturbing states with high **FI** can significantly change the decision process which agrees with our assumptions above. For example, as Figure 1(a) shows, when perturbing the state 461 along the whole trajectory generated by a trained policy, the perturbed trajectory becomes quite different from the original one. These empirical findings suggest that the proposed **FI** is useful to detect potentially vulnerable states for DRL algorithms. More details are in the experiment part.

4. The Influence Measure

The case study in Section 3 tells that **FI** can accurately detect potentially fragile states. In this section, we propose a more general form of **FI** and introduce some important notations related to **FI** such as the perturbation manifold.

Given a state $\boldsymbol{s}$, an action $\boldsymbol{a}$, and a trainable parameter $\boldsymbol{\theta}$ of the policy network, the distribution probability of the future return is represented as $P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta})$. Let $\boldsymbol{\omega}=(\omega_1,\ldots,\omega_p)^T$ be a perturbation vector, and $\boldsymbol{\omega}$ varies in an open subset $\Omega \subseteq \mathbb{R}^p$. The perturbations can be imposed on either the state observation $\boldsymbol{s}$ or the network parameter $\boldsymbol{\theta}$, and the perturbed model is denoted by $P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\omega})$ by introducing the perturbation $\boldsymbol{\omega}$, which has a natural geometrical structure (Amari, 2012).

Following Zhu et al. (2007; 2011), the perturbed model $M = \{P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\omega}) : \boldsymbol{\omega} \in \Omega\}$ can be regarded as p-dimensional manifold. Let $T_{\boldsymbol{\omega}}$ be the vector space of M at $\boldsymbol{\omega}$, which is spanned by p functions $\{\partial_i \ell(\boldsymbol{\omega}|\boldsymbol{z},\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta})\}_{i=1}^p$, where $\partial_i = \partial/\partial\omega_i$ and $\ell(\boldsymbol{\omega}|\boldsymbol{z},\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}) = \log P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\omega})$. The inner product of two basis operators ∂_i and ∂_j can be defined as

$$\begin{aligned} g_{ij}(\boldsymbol{\omega}) &= \langle \partial_i, \partial_j \rangle \\ &= \boldsymbol{E}_{\boldsymbol{\omega}}\left[\partial_i \ell(\boldsymbol{\omega}|\boldsymbol{z},\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}) \partial_j \ell(\boldsymbol{\omega}|\boldsymbol{z},\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta})\right], \end{aligned} \tag{5}$$

where $\boldsymbol{E}_{\boldsymbol{\omega}}$ denotes the expectation taken with respect to $P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\omega})$. The p^2 quantities $g_{ij}(\boldsymbol{\omega}), i,j=1,\ldots,p$ construct the metric tensor $\boldsymbol{G}(\boldsymbol{\omega}) \in \mathbb{R}^{p\times p}$ of the perturbation $\boldsymbol{\omega}$, which is generally assumed to be positive definite in a small neighborhood of $\boldsymbol{\omega}_0$.

Lemma 4.1. *Let $\boldsymbol{\phi}=(\phi_1,\ldots,\phi_p)=\boldsymbol{\phi}(\boldsymbol{\omega})$ be a new coordinate system of M, $k_a^i = \partial\omega_i/\partial\phi_a$, then the geometrical quantities of M in the coordinate system $\boldsymbol{\phi}$ can be written as $g_{ab}(\boldsymbol{\phi}) = \sum_{i,j} k_a^i k_b^j g_{ij}(\boldsymbol{\omega})$.*

The proof of Lemma 4.1 can be found in Amari (2012). Then, for any two tangent vectors $\boldsymbol{t}_i \in T_{\boldsymbol{\omega}}$ with the form of $\boldsymbol{t}_i(\boldsymbol{\omega}) = \boldsymbol{h}_i^T \partial^T \ell(\boldsymbol{\omega}|\boldsymbol{z},\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta})$, where $\boldsymbol{h}_i \in \mathbb{R}^p$ for $i=1,2$, we define the inner product $\langle \boldsymbol{t}_1(\boldsymbol{\omega}), \boldsymbol{t}_2(\boldsymbol{\omega})\rangle$ as

$$\langle \boldsymbol{t}_1(\boldsymbol{\omega}), \boldsymbol{t}_2(\boldsymbol{\omega})\rangle = \sum_{i,j}^{p} h_{1i} h_{2j} g_{ij}(\boldsymbol{\omega}) = \boldsymbol{h}_1^T \boldsymbol{G}(\boldsymbol{\omega}) \boldsymbol{h}_2. \tag{6}$$

Furthermore, the length of $\boldsymbol{t}_1(\boldsymbol{\omega})$ can be expressed as

$$\|\boldsymbol{t}_1(\boldsymbol{\omega})\| = \sqrt{\langle \boldsymbol{t}_1(\boldsymbol{\omega}), \boldsymbol{t}_1(\boldsymbol{\omega})\rangle} = [\boldsymbol{h}_1^T \boldsymbol{G}(\boldsymbol{\omega}) \boldsymbol{h}_1]^{1/2}. \tag{7}$$

Definition 4.2. We define the Riemannian metric tensor $\boldsymbol{G}(\boldsymbol{\omega})$ by (5) and the Riemannian manifold $M = \{P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\omega}) : \boldsymbol{\omega} \in \Omega\}$ with the inner product defined in (6) and (7) as the perturbation manifold around $\boldsymbol{\omega}_0$.

Let $f(\boldsymbol{\omega}) : \mathbb{R}^p \to \mathbb{R}^1$ be the objective function, defining the inference of interest for adversarial strength analysis. Let $C(t) : \boldsymbol{\omega}(t) = (\omega_1(t),\ldots,\omega_p(t))$ be a smooth curve on the manifold M connecting two points $\boldsymbol{\omega}_1 = \boldsymbol{\omega}(t_1)$ and $\boldsymbol{\omega}_2 = \boldsymbol{\omega}(t_2)$ with $\boldsymbol{\omega}(0) = \boldsymbol{\omega}_0$ and $d\boldsymbol{\omega}(t)/dt|_{t=0} = \boldsymbol{h}_{\boldsymbol{\omega}_0} \in T_{\boldsymbol{\omega}_0}$, then the distance between $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ along the curve $C(t)$ can be defined as

$$S_C(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = \int_{t_1}^{t_2} \sqrt{\frac{d\boldsymbol{\omega}(t)}{dt}^T \boldsymbol{G}(\boldsymbol{\omega}(t)) \frac{d\boldsymbol{\omega}(t)}{dt}}\, dt. \tag{8}$$

We can then define the first-order local influence measure ($\mathbf{FI}$) of $f(\boldsymbol{\omega})$ at $\boldsymbol{\omega}_0$ as

$$\mathbf{FI}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0) = \max_{C} \lim_{t\to 0} \frac{[f(\boldsymbol{\omega}(t)) - f(\boldsymbol{\omega}(0))]^2}{S_C^2(\boldsymbol{\omega}(t), \boldsymbol{\omega}(0))}, \tag{9}$$

where $[f(\boldsymbol{\omega}(t)) - f(\boldsymbol{\omega}(0))]^2 / S_C^2(\boldsymbol{\omega}(t), \boldsymbol{\omega}(0))$ can be interpreted as the ratio of the change of the objective function relative to the minimal distance between $P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\omega}(t))$ and $P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\omega}_0)$ on M. The maximum value $\mathbf{FI}$ of the ratio quantifies the extent to which $\boldsymbol{\omega}$ has a local influence on an objective function $f(\boldsymbol{\omega})$, with high $\mathbf{FI}$ representing that $f(\boldsymbol{\omega})$ is more vulnerable to $\boldsymbol{\omega}$ and low $\mathbf{FI}$ representing that $f(\boldsymbol{\omega})$ is less vulnerable to $\boldsymbol{\omega}$. Obviously, $|f(\boldsymbol{\omega}_t) - f(\boldsymbol{\omega}_0)|$ also serves as a measure, and we give a more detailed explanation of the relationship between $\mathbf{FI}$ and $|f(\boldsymbol{\omega}_t) - f(\boldsymbol{\omega}_0)|$. As defined in (9), the numerator is exactly $|f(\boldsymbol{\omega}_t) - f(\boldsymbol{\omega}_0)|$, which measures the change in the objective function from $\boldsymbol{\omega}_0$ (generally 0) to the perturbation $\boldsymbol{\omega}_t$. By dividing this difference by the distance between $\boldsymbol{\omega}_t$ and $\boldsymbol{\omega}_0$ in the denominator, and letting $\boldsymbol{\omega}$ approximate $\boldsymbol{\omega}_0$, we obtain the $\mathbf{FI}$. Therefore, $\mathbf{FI}$ is directly derived from $|f(\boldsymbol{\omega}_t) - f(\boldsymbol{\omega}_0)|$. However, compared to $|f(\boldsymbol{\omega}_t) - f(\boldsymbol{\omega}_0)|$, $\mathbf{FI}$ is more advanced and portrays the potential properties of the perturbation $\boldsymbol{\omega}$, completely free from the constraints of the perturbation $\boldsymbol{\omega}$. Meanwhile, the metric $|f(\boldsymbol{\omega}_t) - f(\boldsymbol{\omega}_0)|$ is still heavily constrained by the perturbation $\boldsymbol{\omega}$.

$\mathbf{FI}$ can be written in an explicit form and is invariant to reparameterization of $\boldsymbol{\omega}$. We now have the following result.

Theorem 4.3. *If $\boldsymbol{G}(\boldsymbol{\omega})$ is positive definite, we have the following results:*

(i) $\mathbf{FI}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0) = \boldsymbol{\nabla}_{f(\boldsymbol{\omega}_0)}^T \boldsymbol{G}^{-1}(\boldsymbol{\omega}_0) \boldsymbol{\nabla}_{f(\boldsymbol{\omega}_0)}$.

(ii) If $\boldsymbol{\phi}$ is a diffeomorphism of $\boldsymbol{\omega}$, then $\mathbf{FI}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0)$ is invariant with respect to any reparametrization corresponding to $\boldsymbol{\phi}$ and $\mathbf{FI}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0)$=$k^2\mathbf{FI}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0)$ holds for any k.

Proof. Note that $f(\boldsymbol{\omega}(t))$ is a function of $\boldsymbol{\omega}(t)$ defined on the perturbation manifold M. It follows from a Taylor series expansion that $f(\boldsymbol{\omega}(t)) = f(\boldsymbol{\omega}(0)) + \boldsymbol{\nabla}_{f(\boldsymbol{\omega}_0)}^T \boldsymbol{h}_{\boldsymbol{\omega}_0} t + \frac{1}{2}\left(\boldsymbol{h}_{\boldsymbol{\omega}_0}^T \boldsymbol{H}_{f(\boldsymbol{\omega}_0)} \boldsymbol{h}_{\boldsymbol{\omega}_0} + \boldsymbol{\nabla}_{f(\boldsymbol{\omega}_0)}^T d^2\boldsymbol{\omega}(0)/dt^2\right) t^2 + o(t^2)$, where $\boldsymbol{\nabla}_{f(\boldsymbol{\omega}_0)} = \partial f(\boldsymbol{\omega})/\partial\boldsymbol{\omega}|_{\boldsymbol{\omega}=\boldsymbol{\omega}_0}$ and $\boldsymbol{H}_{f(\boldsymbol{\omega}_0)} = \partial^2 f(\boldsymbol{\omega})/\partial\boldsymbol{\omega}\partial\boldsymbol{\omega}^T|_{\boldsymbol{\omega}=\boldsymbol{\omega}_0}$. By (8), $S_C^2(\boldsymbol{\omega}(t), \boldsymbol{\omega}(0)) = t^2 \boldsymbol{h}_{\boldsymbol{\omega}_0}^T \boldsymbol{G}(\boldsymbol{\omega}_0) \boldsymbol{h}_{\boldsymbol{\omega}_0} + o(t^2)$. Then, using l'Hôpital's rule, the influence measure defined in (9) can be rewritten as

$$\mathbf{FI}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0) = \max_{\boldsymbol{h}_{\boldsymbol{\omega}}} \frac{\boldsymbol{h}_{\boldsymbol{\omega}}^T \boldsymbol{\nabla}_{f(\boldsymbol{\omega}_0)} \boldsymbol{\nabla}_{f(\boldsymbol{\omega}_0)}^T \boldsymbol{h}_{\boldsymbol{\omega}}}{\boldsymbol{h}_{\boldsymbol{\omega}}^T \boldsymbol{G}(\boldsymbol{\omega}_0) \boldsymbol{h}_{\boldsymbol{\omega}}} = \boldsymbol{\nabla}_{f(\boldsymbol{\omega}_0)}^T \boldsymbol{G}^{-1}(\boldsymbol{\omega}_0) \boldsymbol{\nabla}_{f(\boldsymbol{\omega}_0)}.$$

Assuming $\boldsymbol{\omega} = \boldsymbol{\omega}(\boldsymbol{\phi})$ and $\boldsymbol{\phi} = \boldsymbol{\phi}(\boldsymbol{\omega})$, the Jacobian matrices are $\boldsymbol{\Phi} = \partial\boldsymbol{\phi}/\partial\boldsymbol{\omega}$ and $\boldsymbol{\Psi} = \partial\boldsymbol{\omega}/\partial\boldsymbol{\phi}$. Differentiating the identities $\boldsymbol{\phi}[\boldsymbol{\omega}(\boldsymbol{\phi})] = \boldsymbol{\phi}$ and $\boldsymbol{\omega}[\boldsymbol{\phi}(\boldsymbol{\omega})] = \boldsymbol{\omega}$ with respect to $\boldsymbol{\phi}$ and $\boldsymbol{\omega}$, respectively, leads to $\boldsymbol{\Phi}\boldsymbol{\Psi} = \boldsymbol{\Psi}\boldsymbol{\Phi} = \boldsymbol{I}_p$. By lemma 4.1, we have $\boldsymbol{G}(\boldsymbol{\phi}) = \boldsymbol{\Psi}^T\boldsymbol{G}(\boldsymbol{\omega})\boldsymbol{\Psi}$. Moreover, $\boldsymbol{\nabla}_{f(\boldsymbol{\phi}_0)} = \boldsymbol{\Psi}^T\boldsymbol{\nabla}_{f(\boldsymbol{\omega}_0)}$ and $\boldsymbol{h}_{\boldsymbol{\phi}_0} = \partial\boldsymbol{\phi}(t)/dt|_{t=0} = \boldsymbol{\Phi}\boldsymbol{h}_{\boldsymbol{\omega}_0}$, where $\boldsymbol{\phi}_0 = \boldsymbol{\phi}(\boldsymbol{\omega}_0)$. Using (*i*), we can prove (*ii*). □

Theorem 4.3 indicates that $\mathbf{FI}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0)$ is associated with the first derivative of $f(\boldsymbol{\omega}(t))$ on M evaluated at $t = 0$ and invariant to any reparameterization of $\boldsymbol{\omega}(t)$. In contrast, the conventionally used Cook measure (Cook, 1986) changes with the transformation of $\boldsymbol{\omega}$, which can cause issues, especially when there is scale heterogeneity between parameters to which the perturbation is imposed (Zhu et al., 2011).

Note that the above calculation of the influence measure requires the positive definiteness of $\boldsymbol{G}(\boldsymbol{\omega})$. However, this condition is not always satisfied in many environments, such as Atari games where the state is mostly a high-dimensional image. Motivated by Shu & Zhu (2019), we transform $\boldsymbol{\omega}$ to a vector $\boldsymbol{\nu}$ such that $\boldsymbol{G}(\boldsymbol{\nu})$ is positive definite in a small neighborhood of $\boldsymbol{\nu}_0$ that corresponds to $\boldsymbol{\omega}_0$. Specifically, we apply cSVD $\boldsymbol{G}(\boldsymbol{\omega}_0) = \boldsymbol{U}_0^T\boldsymbol{U}_0$, with $\boldsymbol{U}_0 = [P^{1/2}(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\omega})\partial^T\ell(\boldsymbol{\omega}|\boldsymbol{Z},\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta})/\partial\boldsymbol{\omega}] = \boldsymbol{V}_0\boldsymbol{W}_0$ and $\boldsymbol{W}_0\boldsymbol{W}_0^T = \boldsymbol{R}_0\boldsymbol{\Gamma}_0\boldsymbol{R}_0^T$. $\boldsymbol{V}_0$ and $\boldsymbol{R}_0$ are orthogonal matrices and $\boldsymbol{\Gamma}_0$ is a diagonal matrix. We can then introduce the following proposition, whose proof is similar to that in Shu & Zhu (2019).

Proposition 4.4. *Under the transformation* $\boldsymbol{\nu} = \boldsymbol{\Gamma}_0^{1/2}(\boldsymbol{V}_0\boldsymbol{R}_0)^T\boldsymbol{\omega}$, $\mathbf{FI}_{\boldsymbol{\nu}}(\boldsymbol{\nu}_0)$ *has the form of*

$$\begin{aligned}\mathbf{FI}_{\boldsymbol{\nu}}(\boldsymbol{\nu}_0) &= \boldsymbol{\nabla}_{f(\boldsymbol{\nu}_0)}^T\boldsymbol{\nabla}_{f(\boldsymbol{\nu}_0)} \\ &= \boldsymbol{\nabla}_{f(\boldsymbol{\omega}_0)}^T(\boldsymbol{V}_0\boldsymbol{R}_0)^T\boldsymbol{\Gamma}_0^{-1}(\boldsymbol{V}_0\boldsymbol{R}_0)\boldsymbol{\nabla}_{f(\boldsymbol{\omega}_0)}.\end{aligned} \tag{10}$$

We now summarize the three key steps in carrying out **our proposed adversarial learning framework for DRL**.

- Step 1. Construct a perturbation manifold $M = \{P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\omega}) : \boldsymbol{\omega} \in \Omega\}$ as defined in Definition 4.2.
- Step 2. Given the perturbation manifold, we calculate the geometric quantities, such as $g_{ij}(\boldsymbol{\omega})$.
- Step 3. Choose an objective function $f(\boldsymbol{\omega})$ and calculate **FI** by (9) when the positive definiteness of $\boldsymbol{G}(\boldsymbol{\omega})$ is fulfilled; otherwise, we first transform $\boldsymbol{\omega}$ and calculate **FI** by (10).

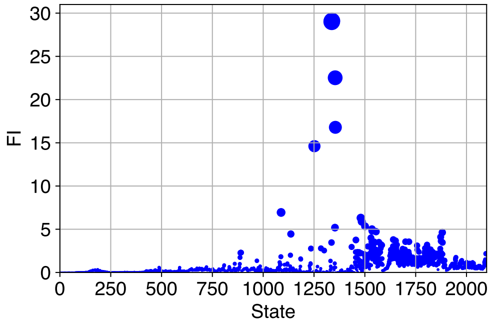


Figure 2. The distribution of **FI**s of all states along the trajectory. The horizontal axis represents the index of state and the circle size represents the magnitude of the **FI** value of each state.

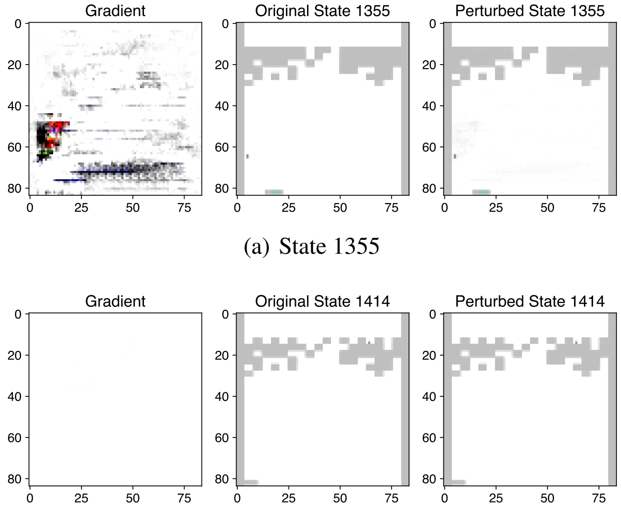
5. Experiments

In this section, we perform numerical studies on the Atari 2600 platform to evaluate the proposed method. Specifically, we focus on the *breakout* environment and the C51 algorithm while everything can be extended to other games and DRL algorithms. Additional experimental results can be found in the Appendix.

5.1. Detection of fragile states

In this part, we apply adversarial learning to the states, following the experimental setup outlined in Section 3. We compute **FI** scores for all the states along a trajectory generated by a trained policy. As Figure 2 shows, the states with the large **FI** scores are between steps 1330 and 1360. Specifically, we select two states, a high-**FI** state 1355 and a low-**FI** state 1414, and assess the changes in the Q-values after imposing a small perturbation. The perturbation used in this work takes the form of $c\boldsymbol{\nabla}_{f(\boldsymbol{\Delta s})}$, which is proportional to the gradient of the objective function and c is an extremely small constant, as shown in Figure 3. The original Q-values at state 1355 are [6.2796354, 6.5242834, 4.1185102, 6.4103985] for the four actions, and the four numbers become [0.6616837, 0.5192522, 2.8266487, 0.08896617] after the perturbation. In this case, the optimal action, determined by selecting the action with the maximum Q-value, has changed. However, for state 1414 with low **FI**, the Q-values change from [9.973947, 9.993811, 9.985605, 9.992829] to [9.977762, 9.995044, 9.987992, 9.99405] and the difference is negligible. This indicates that the Q-values of states with high **FI** exhibit more significant changes compared to the states with low **FI** after perturbation.

Notice that the gradients corresponding to the states with high and low **FI** are quite different. To be fair, we also impose the same small Gaussian noise to these two states. In this case, the Q-values of state 1355 become [5.110825,



(a) State 1355

(b) State 1414

Figure 3. Visualization of two states before and after perturbation. Gradients of the objective function are in the first column, the original states are in the second column, and the states after perturbation are in the third column.

5.184224, 5.6336656, 4.8403316] and those of state 1414 become [9.763626, 9.86359, 9.777034, 9.81879]. State 1355 still suffers a larger change in the Q-values and the optimal action to take differs from that before perturbation. Comparing the two different kinds of perturbations, the simple Gaussian noise requires the determination of the mean and the variance, and the criteria for perturbation are not clear. Thus, we prefer the gradient perturbation $\boldsymbol{\nabla}_{f(\boldsymbol{\Delta s})}$ for the following experiments.

Moreover, we are interested in the change of the trajectory after perturbation. As Figure 4 illustrates, there exist significant differences between the perturbed and original trajectories for three high-**FI** states. Taking state 1355 as an example, we observe that the optimal action taken at this state changes from *FIRE* to *RIGHT* after perturbation. This alteration leads to a deviation from the original trajectory, ultimately resulting in an 11% decrease in the final score compared to the original trajectory. As Figure 5 shows, the sphere is falling to the lower left and *RIGHT* is clearly not a reasonable movement, contributing to the decline in the final score following the perturbation.

We carry out some further analyses by changing the hyperparameter γ from 0.98 to 1. We summarize the **FI**s of all states in Figure 6 and perturb the states with high **FI** as shown in Figure 7. Figure 6 shows that the states with high **FI** are mainly around the 500-th and 1500-th time steps, which is similar to $\gamma = 0.98$. However, as depicted in Figure 7, the changes in the trajectories after perturbation are much more significant than those in the $\gamma = 0.98$ scenario. Figure 8 provides a potential explanation for this phenomenon. Although the trajectory does not change too much immediately after the perturbed state 462, the deviation becomes larger after 300 steps. As Figure 8 shows, the sphere is moving to the left without landing on the board while the agent stops selecting the right action *FIRE* again which results in a premature stop of the trajectory compared with the original trajectory. This result demonstrates that **FI** is an effective tool for detecting potentially vulnerable states that have a substantial negative impact on model performance when being perturbed, even if the policy remains unchanged.

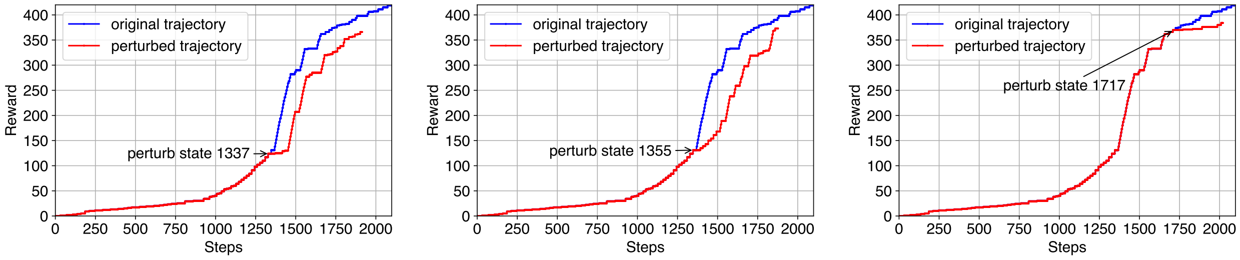
Table 1. The **FI**s of all layers in the policy network.

TRAINABLE LAYER	**FI**
CON2VD1 KERNEL	0.06668868
CON2VD1 BIAS	0.06669269
CON2VD2 KERNEL	0.06676830
CON2VD2 BIAS	0.06674466
CON2VD3 KERNEL	0.06677952
CON2VD3 BIAS	0.06676660
DENSE1 KERNEL	0.06950542
DENSE1 BIAS	0.08280935
DENSE2 KERNEL	233.681872
DENSE2 BIAS	233.575852

5.2. Adversarial learning analysis of policy networks

The proposed **FI** also allows us to quantify the adversarial strength of the parameters in the policy network. Table 1 presents the **FI** values for different layers in the policy network. The **FI** values for all three convolutional layers and the first dense layer are remarkably small, indicating that these layers are less susceptible to perturbations. On the other hand, the **FI** values for the second dense layer are considerably larger, which is expected as this layer is located in the later part of the network and is thus more sensitive to changes.

Furthermore, we can precisely detect the anomalously sensitive parameter that leads to the high **FI** of the entire layer. The kernel dimension of the second dense layer is 512×204 and the bias dimension is 204. Experiments show that the **FI** analysis of the kernel is similar to that of the bias, and here we analyze with the bias out of simplicity. We compute the **FI** values for each dimension of the DENSE2 BIAS parameter, and the results are presented in Figure 9(b). As depicted in Figure 9(b), the 26th, 51st, 77th, 102nd, 128th, 153rd, 179th, and 204th parameters in DENSE2 BIAS have relatively large **FI**s exceeding 1.3, which indicates that these parameters have stronger effects on the model performance. Interestingly, we display all the 204 parameters of the bias



(a) Perturb state 1337 (b) Perturb state 1355 (c) Perturb state 1717

Figure 4. Comparison of trajectories before and after perturbation for three selected states with high **FI**s.

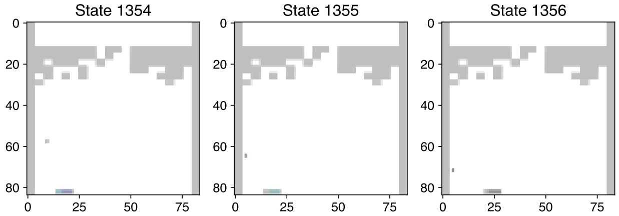


Figure 5. Visualization of state 1354, state 1355 after perturbation, and the new state 1356.

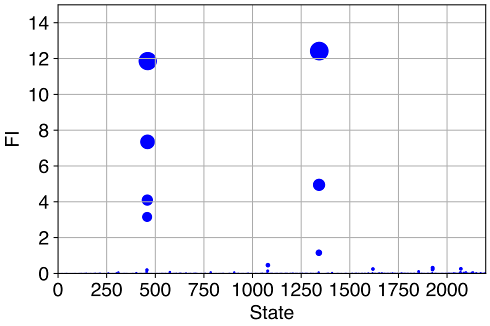


Figure 6. The distribution of **FI**s of states when $\gamma = 1$. The horizontal axis represents the index of the state, and the circle size represents the magnitude of the **FI** value of each state.

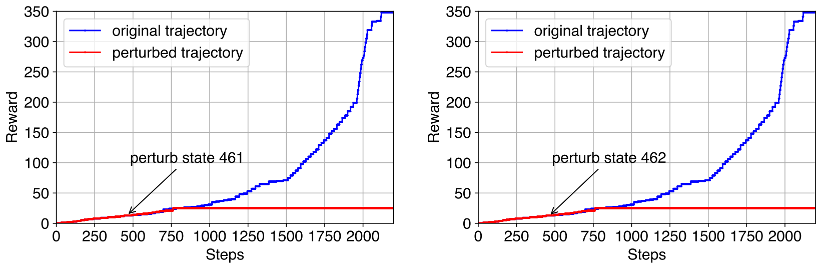
term in Figure 9(a) and find a coincidence that the majority of the 204 parameters are negative, between -0.15 and 0. Only eight parameters are positive, which just happens to be the eight parameters with high **FI** detected by our method. This coincidence suggests that although most parameters in the policy network may be negative, the very few positive parameters can still show great power in affecting the whole system. Moreover, the absolute values of these positive parameters tend to be higher than the negative ones, which may result in their high **FI** values.

To better understand the results of the adversarial learning analysis, we perform slight perturbations to the eight positive parameters with high **FI** by re-scaling them to be 80% or 90% large. As a comparison, we also make some drastic changes to the parameters with the lowest **FI** by changing them to 0. As presented in Figure 10, small changes of the parameters with high **FI** can dramatically change the whole trajectory. The agent loses its ability to select appropriate actions and consequently fails to receive any rewards. However, as Figure 10(c) shows, a 100% magnitude change of the parameter with low **FI** does not have any effect on the model performance. The complete trajectory comparison is in the Appendix.

With the **FI** analysis, we can precisely detect all the ‘weak’ parts of the policy network which can help modify the architecture to achieve better performance or when applying the offline policy to the online environment. This is not the main focus of this paper, but we point out this direction for the future studies.

6. Related Work

One key challenge to RL is the **distribution shift** due to the difference between the learned policy and the behavior policy (Lagoudakis & Parr, 2003; Lange et al., 2012; Schulman et al., 2015; Sun et al., 2018; Janner et al., 2019). A lot of efforts have been made to reduce the distribution shift, either by limiting policy deviations or by estimating (cognitive) uncertainty as a measure of the distribution shift. Different tools have been developed to address the distribution shift issue and facilitate generalization that can be used in offline RL algorithms, including those from causal inference (Schölkopf, 2022), uncertainty estimation (Gal & Ghahramani, 2016; Kendall & Gal, 2017), density estimation and generative modeling (Kingma et al., 2014), distributional robustness (Sinha et al., 2017; Sagawa et al., 2019), and invariance (Arjovsky et al., 2019). Despite some similarities, these works only pay attention to the distribution shift between the online and offline data while we care about the whole RL system, including the **perturbation** imposed



(a) Perturb state 461

(b) Perturb state 462

Figure 7. Comparison of trajectories before and after perturbation for three selected states with high $\mathbf{FI}$s when $\gamma = 1$.

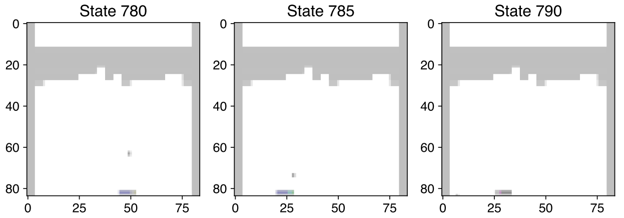


Figure 8. States 780, 785 and 790 in the re-generated trajectory after perturbing state 461 in the original trajectory when γ=1.

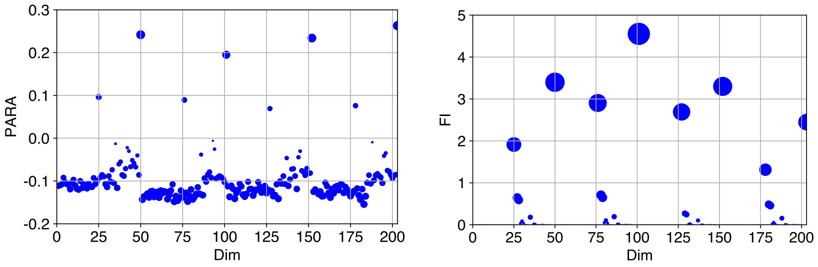
on the input observations, the transition dynamics, and the policy networks.

Although the topic of adversarial learning analysis in RL has never been discussed before, many existing works try to improve the model robustness against the adversarial perturbation attacks in deep learning, see (Everett et al., 2021; Korkmaz, 2020; Lütjens et al., 2020; Tekgul et al., 2022). Adversarial training or retraining adds adversaries to the training dataset (Kurakin et al., 2016; Madry et al., 2017) to increase the robustness of the trained model during testing. Other works increase robustness by distilling networks (Papernot et al., 2016), comparing the output of model ensembles (Tramèr et al., 2017), or comparing the input with a binary filtered transformation of the input (Xu et al., 2017).

Unlike these prior works, we aim to directly quantify the extent of the effect of subtle perturbations on RL performance. We propose an influence measure ($\mathbf{FI}$) for DRL following the development of local influence analysis of (Zhu et al., 2007; 2011). $\mathbf{FI}$ captures the local influence of the objective function around $\boldsymbol{\omega}_0$ (usually 0), and represents the potential sensitivity of a component in RL. $\mathbf{FI}$ is an intrinsic property of the component that remains unchanged as the perturbation changes. To the best of our knowledge, there is currently no similar metric with this intrinsic property that portrays sensitivity in the RL domain. Compared with traditional Euclidean-space based measures, such as Cook's local influence measure (Cook, 1986), our influence measure for DRL captures the intrinsic variation of the objective function (Zhu et al., 2011). Meanwhile, our influence measure provides invariance under diffeomorphisms, which is crucial for evaluating simultaneous effects or comparing the individual effects of different external and/or internal perturbations with respect to their differences in scaling (Shu & Zhu, 2019). Besides the usual analysis of state perturbations (Zhang et al., 2020), our influence measure can also quantify the adversarial strength of policy structure, which is rarely studied.

7. Conclusion

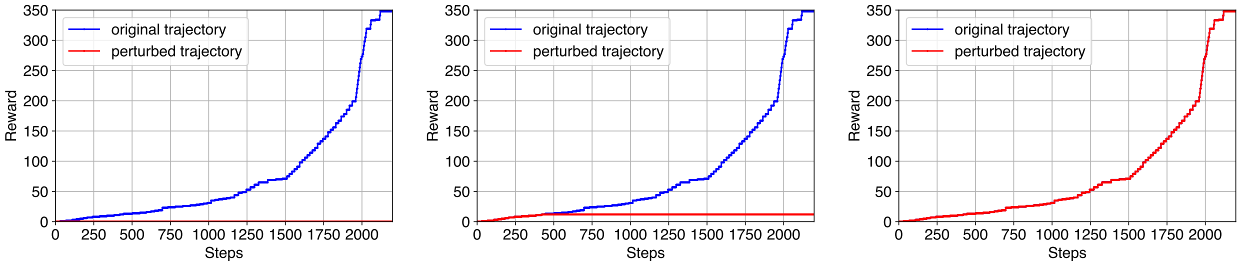
In the real world, many policies are trained in one environment and then applied to another. Although the two environments appear almost the same, the pre-trained policies can still perform poorly in the new environment due to some tiny but non-negligible gaps between the two systems. To better understand the underlying reasons that cause the failure of a trained policy, we introduce a $\mathbf{FI}$-based adversarial learning framework to measure how the change of each local component of the RL system affects the overall performance. $\mathbf{FI}$ is constructed from a perturbation manifold and shows invariance under any reparameterization of the perturbation. We use the $\mathbf{FI}$ to effectively detect the sensitive parts of the system which threaten the model robustness. We also try to impose a small perturbation to the detected components with high $\mathbf{FI}$s to compare the performance of the same policy before and after perturbation. The experimental results on the Atari 2600 platform demonstrate the efficiency of the proposed adversarial learning framework in detecting potentially fragile states and sensitive parameters in the policy network. Our work thus far has primarily concentrated on conducting sensitivity analysis for external input states and internal network structures. Our overarching objective is to expand the $\mathbf{FI}$ analysis framework to encompass all components of the RL system, including continuous action spaces, rewards, and transition models. Additionally, we explore the practical applications of our method. For task



(a) The parameters in DENSE2 BIAS

(b) The **FI**s of the parameters

Figure 9. Visualization of the parameters in the DENSE2 BIAS layer and the corresponding **FI**s.



(a) 102nd parameter -10%

(b) 204th parameter -10%

(c) 152nd parameter -100%

Figure 10. The trajectory comparisons before and after perturbing the parameters with the high **FI** and low **FI**. (a)-(b) Comparison of trajectories for two parameters with high **FI**, (i) Comparison of trajectories for the parameter with the lowest **FI**, where -10% and /100% represent the change of the magnitude of the original parameters.

(i), where vulnerable states are identified, we can leverage these states to generate adversarial examples or enhance the policy through data augmentation. In tasks (ii) and (iii), involving the identification of unstable policy architecture and sensitive policy parameters, the **FI** analysis serves as a valuable guide for selecting or improving the network architecture.

Acknowledgements

Dr. Fan Zhou's work is supported by the National Natural Science Foundation of China (12001356), Shanghai Sailing Program (20YF1412300), "Chenguang Program" supported by Shanghai Education Development Foundation and Shanghai Municipal Education Commission, Open Research Projects of Zhejiang Lab (NO.2022RC0AB06), Shanghai Research Center for Data Science and Decision Technology, Innovative Research Team of Shanghai University of Finance and Economics.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: a system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283, 2016.

Amari, S.-i. *Differential-geometrical methods in statistics*, volume 28. Springer Science & Business Media, 2012.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., and Mordatch, I. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*, 2019.

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458. PMLR, 2017.

Cook, R. D. Assessment of local influence. *Journal of the Royal Statistical Society: Series B (Methodological)*, 48 (2):133–155, 1986.

Dabney, W., Rowland, M., Bellemare, M., and Munos, R. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Everett, M., Lütjens, B., and How, J. P. Certifiable robustness to adversarial state uncertainty in deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Fujimoto, S., Conti, E., Ghavamzadeh, M., and Pineau, J. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*, 2019.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Gulcehre, C., Wang, Z., Novikov, A., Paine, T., Gómez, S., Zolna, K., Agarwal, R., Merel, J. S., Mankowitz, D. J., Paduraru, C., et al. Rl unplugged: A suite of benchmarks for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:7248–7259, 2020.

He, S. and Shin, K. G. Spatio-temporal capsule-based reinforcement learning for mobility-on-demand network coordination. In *The World Wide Web Conference*, pp. 2806–2813, 2019.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Jiang, N. and Li, L. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 652–661. PMLR, 2016.

Kendall, A. and Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014.

Kober, J., Bagnell, J. A., and Peters, J. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Korkmaz, E. Nesterov momentum adversarial perturbations in the deep reinforcement learning domain. In *International Conference on Machine Learning, ICML*, 2020.

Kumar, A., Singh, A., Tian, S., Finn, C., and Levine, S. A workflow for offline model-free robotic reinforcement learning. *arXiv preprint arXiv:2109.10813*, 2021.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

Lagoudakis, M. G. and Parr, R. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4: 1107–1149, 2003.

Lample, G. and Chaplot, D. S. Playing fps games with deep reinforcement learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Lange, S., Gabel, T., and Riedmiller, M. Batch reinforcement learning. In *Reinforcement learning*, pp. 45–73. Springer, 2012.

Li, Y. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

Liang, E., Wen, K., Lam, W. H., Sumalee, A., and Zhong, R. An integrated reinforcement learning and centralized programming approach for online taxi dispatching. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Lütjens, B., Everett, M., and How, J. P. Certified adversarial robustness for deep reinforcement learning. In *Conference on Robot Learning*, pp. 1328–1337. PMLR, 2020.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Nguyen, H. and La, H. Review of deep reinforcement learning for robot manipulation. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pp. 590–595. IEEE, 2019.

Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pp. 582–597. IEEE, 2016.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Precup, D. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, pp. 80, 2000.

Qin, Z., Tang, X., Jiao, Y., Zhang, F., Xu, Z., Zhu, H., and Ye, J. Ride-hailing order dispatching at didi via reinforcement learning. *INFORMS Journal on Applied Analytics*, 50(5):272–286, 2020.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Schölkopf, B. Causality for machine learning. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pp. 765–804. 2022.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

Shu, H. and Zhu, H. Sensitivity analysis of deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4943–4950, 2019.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Sinha, A., Namkoong, H., Volpi, R., and Duchi, J. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Sun, W., Gordon, G. J., Boots, B., and Bagnell, J. Dual policy iteration. *Advances in Neural Information Processing Systems*, 31, 2018.

Tang, X., Qin, Z., Zhang, F., Wang, Z., Xu, Z., Ma, Y., Zhu, H., and Ye, J. A deep value-network based approach for multi-driver order dispatching. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1780–1790, 2019.

Tang, X., Zhang, F., Qin, Z., Wang, Y., Shi, D., Song, B., Tong, Y., Zhu, H., and Ye, J. Value function is all you need: A unified learning framework for ride hailing platforms. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3605–3615, 2021.

Tekgul, B. G., Wang, S., Marchal, S., and Asokan, N. Real-time adversarial perturbations against deep reinforcement learning policies: attacks and defenses. In *European Symposium on Research in Computer Security*, pp. 384–404. Springer, 2022.

Thomas, P., Theocharous, G., and Ghavamzadeh, M. High-confidence off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Xu, W., Evans, D., and Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

Xu, Z., Li, Z., Guan, Q., Zhang, D., Li, Q., Nan, J., Liu, C., Bian, W., and Ye, J. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 905–913, 2018.

Zhang, H., Chen, H., Xiao, C., Li, B., Liu, M., Boning, D., and Hsieh, C.-J. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems*, 33: 21024–21037, 2020.

Zhou, F., Wang, J., and Feng, X. Non-crossing quantile regression for distributional reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 15909–15919, 2020.

Zhou, F., Lu, C., Tang, X., Zhang, F., Qin, Z., Ye, J., and Zhu, H. Multi-objective distributional reinforcement learning for large-scale order dispatching. In *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 1541–1546. IEEE, 2021a.

Zhou, F., Luo, S., Qie, X., Ye, J., and Zhu, H. Graph-based equilibrium metrics for dynamic supply–demand systems with applications to ride-sourcing platforms. *Journal of the American Statistical Association*, 116(536):1688–1699, 2021b.

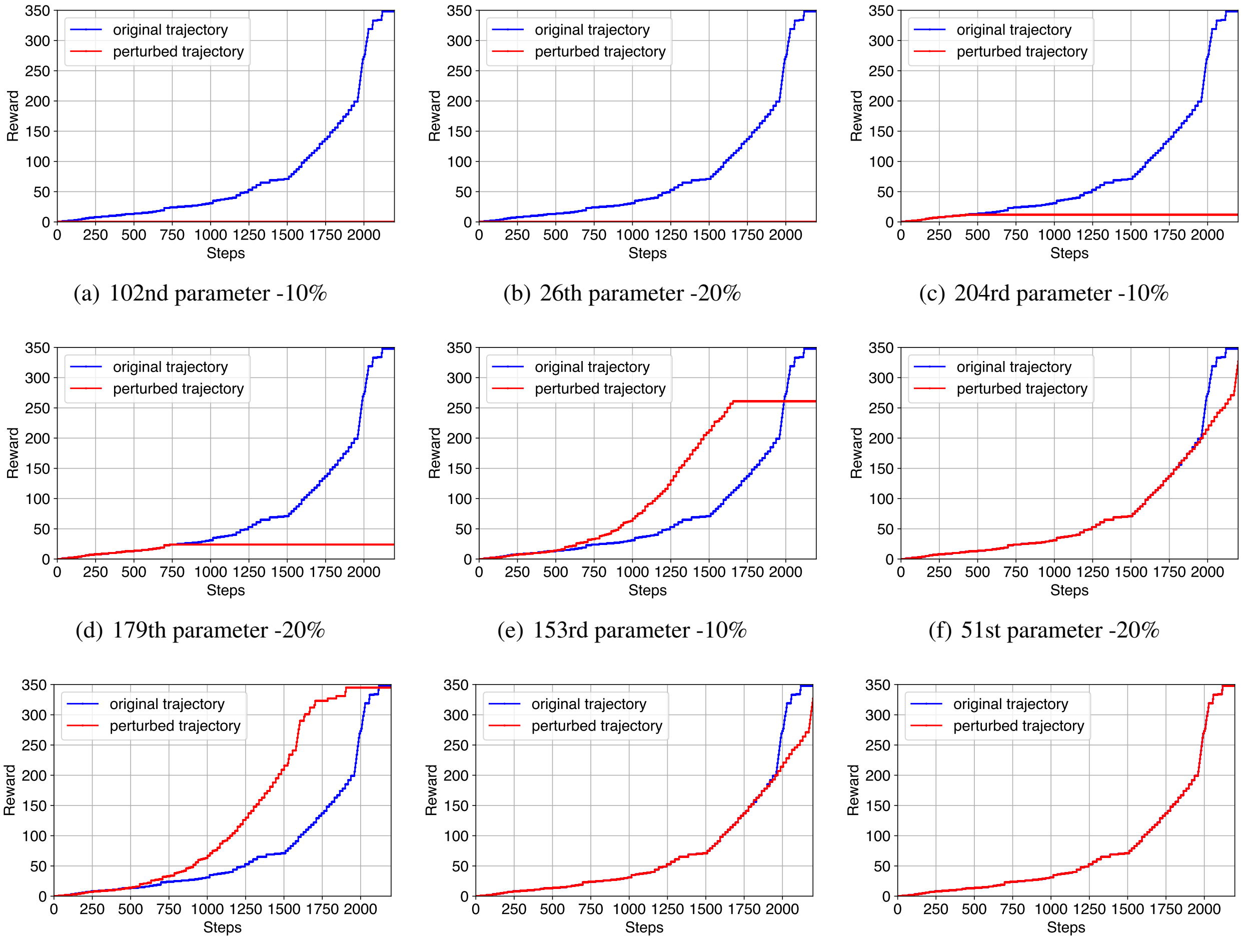
Zhou, F., Zhu, Z., Kuang, Q., and Zhang, L. Non-decreasing quantile function network with efficient exploration for distributional reinforcement learning. In Zhou, Z. (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*, pp. 3455–3461. ijcai.org, 2021c.

Zhu, H., Ibrahim, J. G., Lee, S., and Zhang, H. Perturbation selection and influence measures in local influence analysis. *The Annals of Statistics*, 35(6):2565–2588, 2007.

Zhu, H., Ibrahim, J. G., and Tang, N. Bayesian influence analysis: a geometric approach. *Biometrika*, 98(2):307–323, 2011.

A. Additional Adversarial Learning Results

The complete trajectory comparisons before and after perturbing the eight positive parameters with high **FI** by re-scaling them to be 80% or 90% large and changing the parameter with the lowest **FI** to 0 are in Figure 11. The influence of making changes to parameters with high **FI**s is dramatic, but a 100% magnitude change to parameters with low **FI** does not have any effect on the trajectory.



(a) 102nd parameter -10%

(b) 26th parameter -20%

(c) 204rd parameter -10%

(d) 179th parameter -20%

(e) 153rd parameter -10%

(f) 51st parameter -20%

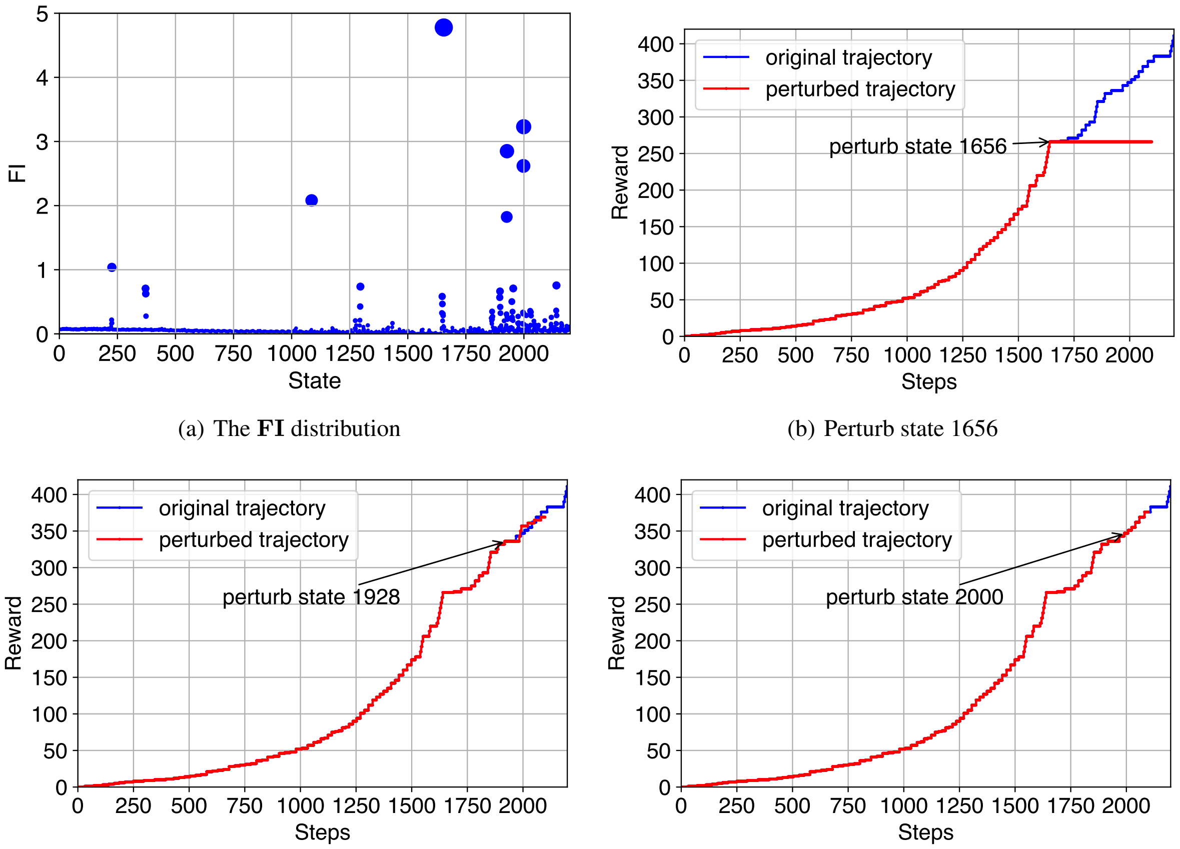
(g) 77th parameter -20%

(h) 128th parameter -20%

(i) 152nd parameter -100%

Figure 11. The trajectory comparisons before and after perturbing the parameters with the high **FI** and the lowest **FI**, respectively, and the original trajectory, (a)-(g) the comparisons of the first eight parameters with high **FI**, (i) the comparison of the parameter with the lowest **FI**, where -10%/20%/100% represents the percentage value of the original parameter subtracted.

We provide some additional results of the state sensitivity analysis by **FI**, including the result of the *breakout* environment when γ is 0.995, in Figure 12, and the corresponding results for the *Alien* and *Asteroids* environments are shown in Figure 13 and Figure 14, respectively, with sensitivity analysis similar to that mentioned above. In addition to the cases with a relatively discrete **FI** distribution, our experiments also obtained some very concentrated **FI** distribution cases, for example, in the *Pong* and *Freeway* environments, **FI** fluctuates around 0.032 and 0.025, respectively, see Figure 15, indicating that there are no particularly vulnerable state points on the trajectory. Perturbing the state of the highest **FI** in the trajectory in the *pong* environment changes the distribution of Q-values from [1.3898636, 1.4035478, 1.3787719, 1.3975887, 1.3763726, 1.4176952] to [1.4398711, 1.4520737, 1.4266012, 1.4448254, 1.4240125, 1.4781928], with minimal change, and does not affect the choice of action. To investigate the reason, the trajectory scores in the *Pong* and *Freeway* environments basically reach the upper limit, indicating that the strategies are well trained and do not have excessive fragility.



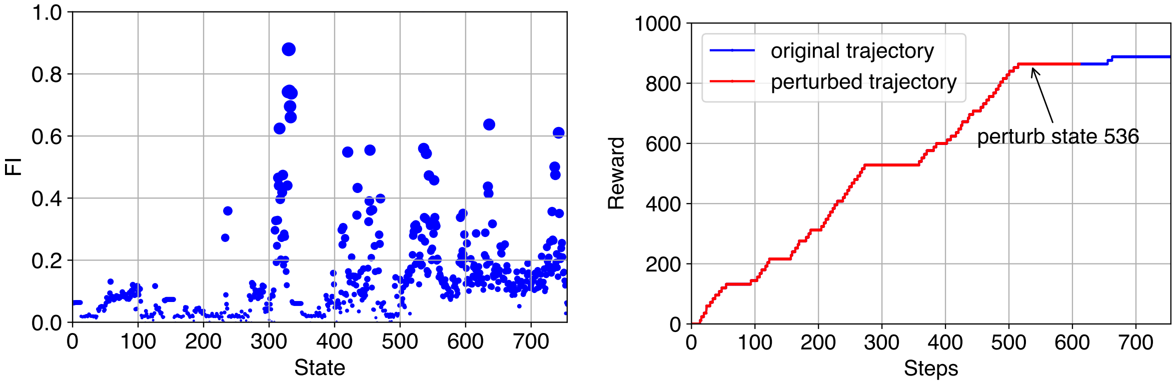
(a) The **FI** distribution

(b) Perturb state 1656

(c) Perturb state 1928

(d) Perturb state 2000

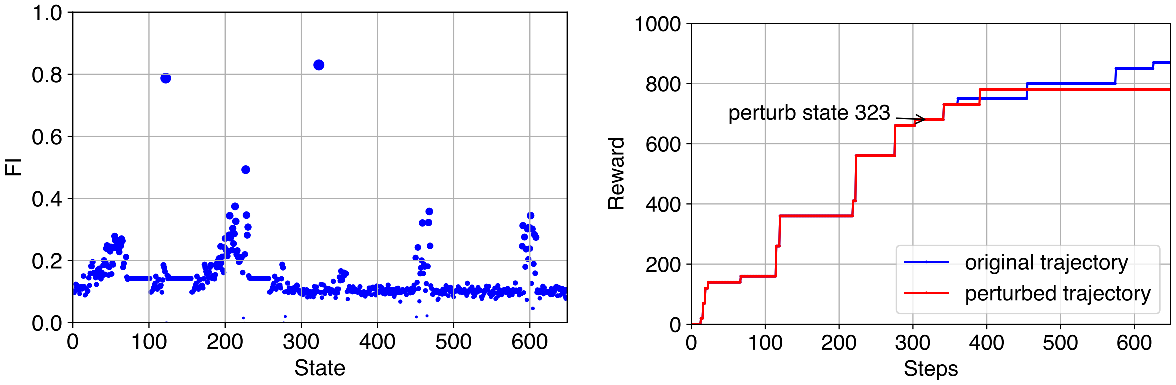
Figure 12. The adversarial learning analysis of *Breakout* with γ=0.995, (a) for the the **FI** distribution along the steps, (b)-(d) for the comparisons of the trajectory after perturbing the parameters with the high **FI**s.



(a) The **FI** distribution

(b) Perturb state 536

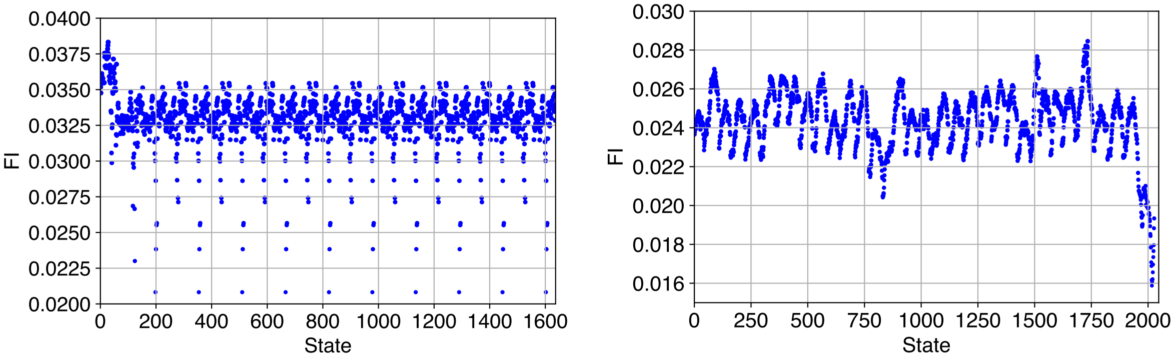
Figure 13. The adversarial learning analysis of *Alien*, (a) for the the **FI** distribution along the steps, (b) for the comparisons of the trajectory after perturbing the parameters with the high **FI**.



(a) The **FI** distribution

(b) Perturb state 323

Figure 14. The adversarial learning analysis of *Asteroids*, (a) for the the **FI** distribution along the steps, (b) for the comparisons of the trajectory after perturbing the parameters with the high **FI**.



(a) The **FI** distribution of *Pong*

(b) The **FI** distribution of *Freeway*

Figure 15. The **FI** distribution of *Pong* and *Freeway*, respectively, (a) for *Pong*, (b) for *Freeway*.