Observational Robustness and Invariances in Reinforcement Learning via Lexicographic Objectives

Daniel Jarne Ornia*
Delft University of Technology
The Netherlands

Licio Romao
University of Oxford
United Kingdom

Lewis Hammond
University of Oxford
United Kingdom

Manuel Mazo Jr.
Delft University of Technology
The Netherlands

Alessandro Abate
University of Oxford
United Kingdom

ABSTRACT

Policy robustness in Reinforcement Learning (RL) may not be desirable at any price; the alterations caused by robustness requirements from otherwise optimal policies should be explainable and quantifiable. Policy gradient algorithms that have strong convergence guarantees are usually modified to obtain robust policies in ways that do not preserve algorithm guarantees, which defeats the purpose of formal robustness requirements. In this work we study a notion of robustness in partially observable MDPs where state observations are perturbed by a noise-induced stochastic kernel. We characterise the set of policies that are *maximally robust* by analysing how the policies are altered by this kernel. We then establish a connection between such robust policies and certain properties of the noise kernel, as well as with structural properties of the underlying MDPs, constructing sufficient conditions for policy robustness. We use these notions to propose a robustness-inducing scheme, applicable to any policy gradient algorithm, to formally trade off the reward achieved by a policy with its robustness level through *lexicographic optimisation*, which preserves convergence properties of the original algorithm. We test the the proposed approach through numerical experiments on safety-critical RL environments, and show how the proposed method helps achieve high robustness when state errors are introduced in the policy roll-out.

1 INTRODUCTION

Robustness in Reinforcement Learning (RL) can be looked at from different perspectives [25]: (1) distributional shifts in the training data with respect to the deployment stage Heger [13], Nilim and El Ghaoui [27], Satia and Lave Jr [34], Xu and Mannor [47]; (2) uncertainty in the model or observations [9, 31]; (3) adversarial attacks against actions [10, 29]; and (4) sensitivity of neural networks (used as policy or value function approximators) towards input disturbances [15, 19]. Robustness does not naturally emerge in most RL settings, since agents are typically only trained in a single, unchanging environment. Indeed, there is often a trade-off between how robust a policy is and how close it is to the set of optimal policies in its training environment.

To address this trade-off in the context of robustness versus observational noise, we define a regret-based robustness notion characterised by a stochastic map to quantify robustness and investigate what makes a policy *maximally robust*. We then propose using a *relaxed lexicographic trade-off*, in which we allow policies to trade performance (discounted reward sum) to robustness. This is done through the use of lexicographic RL (LRL) [37], which allows policies to deviate from a prioritised objective in favour of obtaining better results for the next objectives priority-wise.

1.1 Previous Work

In robustness against *model uncertainty*, the MDP may have noisy or uncertain reward signals or transition probabilities, as well as possible resulting *distributional shifts* in the training data [1, 11, 13, 30, 32, 47], which connects to ideas on distributionally robust optimisation [45, 46]. One of the first examples is Heger [13], where the author proposes using minimax approaches to learn Q functions that minimise the worst case total discounted cost in a general MDP setting. Derman et al. [8] propose a Bayesian approach to deal with uncertainty in the transitions. Another robustness subproblem is studied in the form of *adversarial attacks or disturbances* by considering adversarial attacks on policies or action selection in RL agents [12, 17, 20, 28, 42, 43]. Recently, Gleave et al. [12] propose the idea that instead of modifying observations, one could attack RL agents by swapping the policy for an adversarial one at given times, and prove the existence of such policies in a zero-sum game framework. For a detailed review on Robust RL see Moos et al. [24].

* Contact Author: d.jarneornia@tudelft.nl.

Our work focuses in the study of robustness versus *observational disturbances*, where agents observe a disturbed state measurement and use it as input for the policy [2, 15, 19, 22, 48, 49]. In particular Mandlekar et al. [22] consider both random and adversarial state perturbations, and introduce physically plausible generation of disturbances in the training of RL agents that make the resulting policy robust towards realistic disturbances. Zhang et al. [49] propose a *state-adversarial* MDP framework, and utilise adversarial regularising terms that can be added to different deep RL algorithms to make the resulting policies more robust to observational disturbances, minimising the distance bound between disturbed and undisturbed policies through convex relaxations of neural networks to obtain robustness guarantees. Zhang et al. [48] study the existence of optimal state-perturbing adversaries, and investigate how using LSTM increases robustness in such setting.

1.2 Main Contributions

Much of the existing work on RL with observational disturbances proposes modifications to RL algorithms that come at the cost of *explainability* and *verifiability*, since they obscure the induced changes in the new policies and often result in a loss of convergence guarantees. Our main contributions and differentiating features with previous work are summarised in the following points.

Structure of Robust Policy Sets.[1] We consider general unknown stochastic disturbances and formulate a quantitative definition of observational robustness that allows us to characterise the sets of robust policies for any MDP in the form of operator-invariant sets. We analyse how the structure of these sets depends on the MDP and noise kernel, and obtain an inclusion relation (*i.e.* the Inclusion Theorem in Section 3) that provides intuition into how we can search for robust policies more effectively.

Verifiable Robustness through LRL. We cast robustness as a lexicographic objective, allowing us to retain policy optimality up to a specified tolerance while maximising robustness and yielding a mechanism to formally control the performance-robustness trade-off. Previous work proposes either learning to deal with perturbations by means of linear combination with regularising loss terms or adversarial learning: both approaches result in lack of guarantees and of explainability. We provide numerical examples on how this logic is applied to existing policy gradient algorithms, compare with existing algorithms in previous work, and verify how the previously mentioned Inclusion Theorem helps to induce more robust policies while retaining algorithm optimality.

[1]We claim novelty on the application of such concepts to the understanding and improvement of robustness in disturbed observation RL. Although we have not found our results in previous work, there are strong connections between Sections 2-3 in this paper and the literature on planning for POMDPs [38, 39] and MDP invariances [26, 36, 44].

1.3 Preliminaries

We use calligraphic letters $\mathcal{A}$ for collections of sets and $\Delta(\mathcal{A})$ as the space of probability measures over $\mathcal{A}$. For two elements of a vector space we use $\langle \cdot, \cdot \rangle$ as the inner product. We use 1_n as a column-vector of size n that has all entries equal to 1. We say that an MDP is *ergodic* if for any policy the resulting Markov Chain (MC) is ergodic. We say that S is a $n \times n$ row-stochastic matrix if $S_{ij} \geq 0$ and each row of S sums to 1.

2 OBSERVATIONALLY ROBUST REINFORCEMENT LEARNING

We restrict the robustness problem considered in this work to the following version of a noise-induced partially observable Markov Decision Process [38].

Definition 1. *An observationally-disturbed MDP (DOMDP) is (a POMDP) defined by the tuple (X, U, P, R, T, γ) where X is a finite set of states, U is a set of actions, $P : U \times X \mapsto \Delta(X)$ is a probability measure of the transitions between states and $R : X \times U \times X \mapsto \mathbb{R}$ is a reward function. The map $T : X \mapsto \Delta(X)$ is a stochastic kernel induced by some noise signal, such that $T(y \mid x)$ is the probability of measuring y while the true state is x. At last $\gamma \in [0, 1]$ is a reward discount factor.*

In a DOMDP agents can measure the full state, but this state will be disturbed by some possibly unknown random signal when computing policy actions. The difficulty of acting in such DOMDP is that the transitions are actually undisturbed and a function of the true state x, but agents have to act based on disturbed states $T(\cdot \mid x)$. Unlike the POMDP setting, however, we can access the true state x during learning of the policies, but we do not have information about the noise kernel T or a way to estimate it. It is only in the policy execution that the noise is introduced. We then need to construct policies that will be as robust as possible against noise without being able to construct noise estimates. This is a setting that reflects many control problems; we can design a controller for ideal noise-less conditions, and we know that at deployment there will likely be noise, data corruption, adversarial perturbations, *etc.*, but we do not have certainty on the disturbance structure. The problem of deriving robust policies in such a situation is referred to as Observationally Robust RL [24].

Remark 1. *Definition 1 is a generalised form of the State-Adversarial MDP used by Zhang et al. [49], since we do not necessarily assume that the noise kernel is adversarial, and we consider it to be defined by a stochastic kernel (instead of a deterministic disturbance), so the adversarial case is a* *particular form of DOMDP where T is a probability measure that assigns probability one to one of the states.*

A (memoryless) policy for the agent is a stochastic kernel $\pi : X \mapsto \Delta(U)$. For simplicity, we overload notation on π, denoting by $\pi(x, u)$ as the probability of taking action u at state x under the stochastic policy π in the MDP, *i.e.*, $\pi(x, u) = \Pr\{u \mid x\}$. The value function of a policy π, $V^\pi : X \mapsto \mathbb{R}$, is given by $V^\pi(x_0) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(x_t, \pi(x_t), x_{t+1})]$. The action-value function of π (Q-function) is given by $Q^\pi(x, u) = \sum_{y \in X} P(x, u, y)(R(x, u, y) + \gamma V^\pi(y))$. It is well known that, under mild conditions [40], the optimal value function can be obtained by means of the Bellman equation $V^*(x) := \max_u \sum_{y \in X} P(x, u, y)(R(x, u, y) + \gamma V^*(y))$, and an optimal policy $\pi^*(x) := \arg\max_\pi V^\pi(x)\ \forall x \in X$ is guaranteed to exist. We then define the objective function as

$$J(\pi) := \mathbb{E}_{x_0 \sim \mu_0}[V^\pi(x_0)]$$

with μ_0 being a distribution of initial states, and we use $J^* := \max_\pi J(\pi)$. If a policy is parameterised by $\theta \in \Theta$ we write π_θ and $J(\theta)$.

We now formalise a notion of *observational robustness*. Firstly, due to the presence of the stochastic kernel T, the policy we are applying is altered as we are applying a collection of actions in a possibly wrong state. This behaviour can be formally captured by:

$$\Pr\{u \mid x, \pi, T\} = \langle \pi, T \rangle(x, u) := \sum_{y \in X} T(y \mid x)\pi(y, u), \quad (1)$$

where $\langle \pi, T \rangle : X \mapsto \Delta(U)$ is the *disturbed* policy, which averages the current policy given the error induced by the presence of the stochastic kernel. Notice that $\langle \cdot, T \rangle(x) : \Pi \mapsto \Delta(U)$ is an averaging operator yielding the alteration of the policy due to noise. We can then define the *robustness regret*:

$$\rho(\pi, T) := J(\pi) - J(\langle \pi, T \rangle). \quad (2)$$

Definition 2 (Policy Robustness). *We say that a policy π is κ-*robust *against a stochastic kernel T if $\rho(\pi, T) \leq \kappa$. If π is 0-*robust *we say it is maximally robust. We define the sets of κ-robust policies, $\Pi_\kappa := \{\pi \in \Pi : \rho(\pi, T) \leq \kappa\}$, with Π_0 being the set of maximally robust policies.*

One can motivate the characterisation and models above from a control perspective, where policies use as input discretised state measurements with possible sensor measurement errors. For further details on this, we refer the reader to Appendix A.

2.1 Problem Formulation

Formally ensuring robustness properties when learning RL policies will, in general, force the resulting policies to deviate from optimality in the undisturbed MDP. With this motivation, we propose solving the problem of increasing robustness of RL policies through a hierarchical lexicographic approach, which naturally incorporates trade-offs during the policy design. The first objective is to minimise the distance $J^* - J(\pi)$ up to some tolerance. Then, from the policies that satisfy this constraint, we want to steer the learning algorithm towards a maximally robust policy according to the metric defined in Definition 2. This can be formulated as the following problem.

Problem 1. *For a DOMDP and a given tolerance level ϵ, derive a policy π^ϵ that satisfies $J^* - J(\pi^\epsilon) \leq \epsilon$ as a prioritised objective and is as robust as possible according to Definition 2.*

We solve this problem by means of LRL, casting robustness as a valid lexicographic objective. Figure 1 represents a qualitative interpretation of the results in this work. The structure of the robust sets will become clear in following sections.

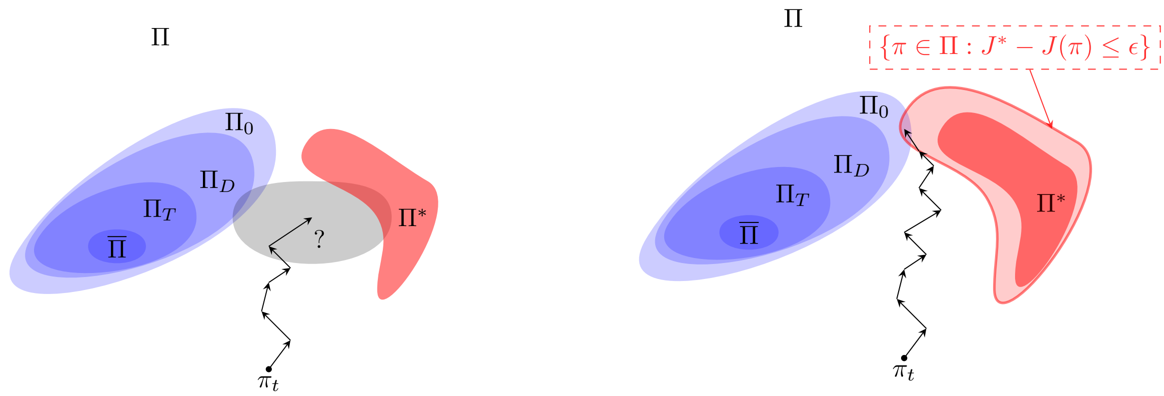
2.2 Lexicographic Reinforcement Learning

Recently, lexicographic optimisation [16, 33] has been applied to the multi-objective RL setting [37]. In an LRL setting with different objective functions $\{K_i\}_{1 \leq i \leq n}$, some objectives may be more important than others, and so we may want to obtain policies that solve the multi-objective problem in a lexicographically prioritised way, *i.e.*, "find the policies that optimise objective i (reasonably well), and from those the ones that optimise objective $i + 1$ (reasonably well), and so on". There exist both value- and policy-based algorithms for LRL, and the approach is broadly applicable to (most) state of the art RL algorithms [37]. We propose using policy-based LRL (PB-LRL) to encode the idea that, when learning how to solve an RL task, robustness is important but *not at any price*, *i.e.*, we would like to solve the original objective reasonably well, and from those policies efficiently find the most robust one[2].

Consider a parameterised policy π_θ with $\theta \in \Theta$, and two objective functions K_1 and K_2. PB-LRL uses a multi-timescale optimisation scheme to optimise θ faster for higher-priority objectives, iteratively updating the constraints indiuced by these priorities and encoding them via Lagrangian relaxation techniques [4]. Let $\theta' \in \arg\max_\theta K_1(\theta)$. Then, PB-LRL can be used to find parameters:

$$\theta'' = \arg\max_\theta K_2(\theta) \quad \text{such that} \quad K_1(\theta) \geq K_1(\theta') - \epsilon.$$

[2]The advantage of LRL is that we need not know in advance how to define "reasonably well" for each new task. Additionally, robustness through LRL provides us with a hyper-parameter that directly controls the trade-off between *robustness and optimality*: the optimality tolerance ϵ. By selecting values of ϵ we determine how far we allow our resulting policy to be from an optimal policy in favour of it being more robust.



(a) PG algorithms when robustness terms are added to the cost function *indiscriminately*.

(b) In LRPG, the policy is guaranteed (up to the original algorithm used) to converge to an ϵ ball of Π^*, and from those, the most robust ones.

Figure 1: Qualitative representation of the proposed LRPG algorithm, compared to usual robustness-inducing algorithms. The sets in blue are the maximally robust policies to be defined in the coming sections. Through LRPG we guarantee that the policies will only deviate a bounded distance from the original objective, and induce a search for robustness in the resulting valid policy set.

This is done by computing the (estimated) gradient ascent update:

$$\theta \leftarrow \operatorname{proj}_{\Theta}\left[\theta + \nabla_\theta \hat{K}(\theta)\right],\ \lambda \leftarrow \operatorname{proj}_{\mathbb{R}_{\geq 0}}\left[\lambda + \eta_t(\hat{k}_1 - \epsilon_t - K_1(\theta))\right], \tag{3}$$

where $\hat{K}(\theta) := (\beta_t^1 + \lambda\beta_t^2)\cdot K_1(\theta) + \beta_t^2 \cdot K_2(\theta)$, λ is a Langrange multiplier, β^1, β^2, η are learning rates, and $\hat{k}_1$ is an estimate of $K_1(\theta')$. Typically, we set $\epsilon_t \to 0$, though we can use other tolerances too, *e.g.*, $\epsilon_t = 0.9 \cdot \hat{k}_1$. For more detail on the convergence proofs and particularities of PB-LRL we refer the reader to Skalse et al. [37].

3 CHARACTERISATION OF ROBUST POLICIES

An important question to be addressed, before trying to synthesise robust policies through LRL, is what these robust policies look like, and how they are related to DOMDP properties.

3.1 Policies that are Maximally Robust

The robustness notion in Definition 2 is intuitive and it allows us to classify policies. We begin by exploring what are the types of policies that are maximally robust, starting with the set of constant policies and set of fix point of the operator $\langle \cdot, T \rangle$, whose formal descriptions are now provided.

Definition 3. *A policy $\pi : X \mapsto \Delta(U)$ is said to be constant if $\pi(x) = \pi(y)$ for all $x, y \in X$, and the collection of all constant policies is denoted by $\bar{\Pi}$. A policy $\pi : X \mapsto \Delta(U)$ is called a fixed point of the operator $\langle \cdot, T \rangle$ if $\pi(x) = \langle \pi, T \rangle(x)$ for all $x \in X$. The collection of all fixed points will be denoted by Π_T.*

In other words, a constant policy is any policy that yields the same action distribution for any state, and a fixed point policy is any policy whose action distributions are un-altered by the noise kernel. Observe furthermore that Π_T *only depends on the kernel T* and the set[3] X. We now present a proposition that links the two sets of policies in Definition 3 with our notion of robustness.

Proposition 1. *Consider a DOMDP as in Definition 1, the robustness notion given in Definition 2 and the concepts in Definition 3, then we have that*

$$\bar{\Pi} \subseteq \Pi_T \subseteq \Pi_0.$$

The importance of Proposition 1 is that it allows us to produce (approximately) maximally robust policies by computing the distance of a policy to either the set of constant policies or to the fix point of the operator $\langle \cdot, T \rangle$, and this is at the core of the construction in Section 4. However, before this, let us introduce another set that is sandwiched between Π_0 and Π_T. Let us assume we have a policy iteration algorithm that employs an action-value function Q^π and policy π. The advantage function for π is defined as

[3] There is a (natural) bijection between the set of constant policies and the space $\Delta(U)$. The set of fixed points of the operator $\langle \cdot, T \rangle$ also has an algebraic characterisation in terms of the null space of the operator $\mathrm{Id}(\cdot) - \langle \cdot, T \rangle$. We are not exploiting the later characterisation in this paper.

$A^\pi(x,u) := Q^\pi(x,u) - V^\pi(x)$ and can be used as a maximisation objective to learn optimal policies (as in, *e.g.*, A2C [41], A3C [23]). We can similarly define the *noise disadvantage* (a form of negative advantage) of policy π as:

$$D^\pi(x,T) := V^\pi(x) - \mathbb{E}_{u\sim\langle\pi,T\rangle(x)}[Q^\pi(x,u)], \qquad (4)$$

which measures the difference of applying at state x an action according to the policy π with that of playing an action according to $\langle\pi,T\rangle$ and then continuing playing an action according to π. Our intuition says that if it happens to be the case that $D^\pi(x,T)=0$ for all states in the DOMDP, then such a policy is maximally robust. And this is indeed the case, as shown in the next proposition.

Proposition 2. *Consider a DOMDP as in Definition 1 and the robustness notion as in Definition 2, then if a policy π is such that $D^\pi(x,T)=0$ for all $x\in X$, then π is maximally robust, i.e., let*

$$\Pi_D := \{\pi\in\Pi : D^\pi(x,T)=0\,\forall\, x\in X\}.$$

then we have that $\Pi_D\subseteq\Pi_0$.

So far we have shown that both the set of fixed points $\overline{\Pi}$ and the set of policies for which the disadvantage function is equal to zero Π_D are contained in the set of maximally robust policies. More interesting is the fact that the inclusion established in Proposition 1 and the one in Proposition 2 can be linked in a natural way. We call this connection, which is the main result of this section, the Inclusion Theorem.

Theorem 1 (Inclusion Theorem). *For a DOMDP with noise kernel T, consider the sets $\overline{\Pi},\Pi_T,\Pi_D$ and Π_0. Then, the following inclusion relation holds:*

$$\overline{\Pi}\subseteq\Pi_T\subseteq\Pi_D\subseteq\Pi_0.$$

Additionally, the sets $\overline{\Pi},\Pi_T$ are convex *for all MDPs and kernels T, but Π_D,Π_0 may not be.*

For most DOMDPs the inclusions above are strict. However, this is not always the case, and we expand on this in the next section.

3.2 Considerations on the Robust Policy Sets

Let us reflect on the inclusion relations of Theorem 1. First, any constant policy is a fixed point of the noise kernel T, and any fixed point produces a noise disadvantage of zero, but there are other policies that satisfy $D^\pi(x,T)=0$ without being a fixed point of $\langle\cdot,T\rangle$. For example, consider a particular case of an MDP where two actions u_1,u_2 are optimal for all states. Then we can find policies π' that satisfy $D^{\pi'}(x,T)=0$ and yet are modified by the noise operator, $\langle\pi',T\rangle\neq\pi'$, *e.g.*, a policy that yields u_1 for half of the states, and u_2 for the rest. Then, any transformation with a uniform noise kernel T will yield a different policy, and retain zero disadvantage.

Similarly for the third inclusion, we can find policies that are maximally robust without satisfying $D^\pi(x,T)=0\,\forall\, x\in X$. In particular, recall $J(\pi)=E_{x_0\sim\mu_0}[V^\pi(x_0)]$. There could exist $x'\in X$ that is *never visited* in all the possible MDP trajectories, for which the policy has $D^\pi(x',T)\neq 0$, but $D^\pi(x,T)=0$ for any point x visited in the trajectories.

Corollary 1. *For any* ergodic *MDP (with reward function R) and noise kernel T, there exist reward functions $\overline{R}$ and $\underline{R}$ such that the resulting DOMDP satisfies:*

(i) $\Pi_D=\Pi_0=\Pi$ *(any policy is max. robust), if $R=\overline{R}$,*
(ii) $\Pi_T=\Pi_D=\Pi_0$ *(only fixed point policies are maximally robust), if $R=\underline{R}$.*

We can now summarise the insights from Theorem 1 and Corollary 1 in the following conclusions. An example is presented in Appendix A.

(1) The set $\overline{\Pi}$ is maximally robust, convex and *independent of the DOMDP.*
(2) The set Π_T is maximally robust, convex, includes $\overline{\Pi}$, and its properties *only depend* on T.
(3) The set Π_D includes Π_T and is maximally robust, but its properties *depend on the DOMDP.*

4 ROBUSTNESS THROUGH LEXICOGRAPHIC OBJECTIVES

We have now characterised robustness in a DOMDP and explored the relation between the sets of policies that are robust according to the definition proposed. We have seen in the Inclusion Theorem that several classes of policies are maximally robust, and our goal now is to connect these results with lexicographic optimisation. To be able to apply LRL results to our robustness problem we need to first cast robustness as a valid objective to be maximised, and then show that a stochastic gradient descent approach would indeed find a global maximum of the objective, therefore yielding a maximally robust policy. Then, this robustness objective can be combined with a primary reward-maximising objective $K_1(\theta)=\mathbb{E}_{x_0\sim\mu_0}[V^{\pi_\theta}(x_0)]$ and any algorithm with certified convergence to produce a solution to Problem 1.

Assumption 1 (Learning Rates). *We assume all learning rates $\alpha_t(x,u)\in[0,1]$ satisfy the conditions $\sum_{t=1}^\infty\alpha_t(x,u)=\infty$ and $\sum_{t=1}^\infty\alpha_t(x,u)^2<\infty$.*

Assumption 2. *For any DOMDP and policy π, the resulting Markov Chain is irreducible and aperiodic.*

Assumption 2 ensures that for any DOMDP and policy π, there exists a stationary probability distribution of states, and for every policy and state this probability is larger than zero.

We propose now a valid lexicographic objective for which a minimising solution yields a maximally robust policy.

One of the messages of the Inclusion Theorem is the fact that fixed points and constant policies are maximally robust, the latter being completely oblivious to the specific choice of stochastic kernel, a feature that is relevant to the design of robust policies. Inspired by this fact, we let $\tilde{T}$ be any stochastic kernel whose set of fixed points coincides with the collection of constant policies and consider the optimisation problem

$$\underset{\theta}{\text{minimise}}\, K(\theta) = \sum_{x\in X} p^{\pi_\theta}(x)\frac{1}{2}\|\pi_\theta(x) - \langle \pi_\theta, \tilde{T}\rangle(x)\|_2^2, \quad (5)$$

where we recall that π_θ is a given parameterisation of the set of policies. Notice that the optimisation problem 5 projects the current policy onto the set of fixed points of the operator $\langle\cdot,\tilde{T}\rangle$, and due to Assumption 2, which requires $p^{\pi_\theta}(x) > 0$ for all $x \in X$, the optimal solution is equal to zero if and only if there exists a value of the parameter θ for which the corresponding π_θ is a fixed point of $\langle\cdot,\tilde{T}\rangle$. In practice, the objectives are computed for a batch of trajectory sampled states $X_s \subset X$, and averaged over $\frac{1}{|X_s|}$; we denote these approximations with a hat. By applying standard stochastic approximation arguments, we can prove that convergence is guaranteed for a SGD iteration using $\nabla_\theta \hat{K}_{\tilde{T}}(\theta)(x) = (\pi_\theta(x) - \pi_\theta(y))\nabla_\theta \pi_\theta(x)$, $y \sim \tilde{T}(\cdot \mid x)$ to the optimal solution of problem 5. For details and proof, see Appendix B. A reasonable choice for the stochastic kernel $\tilde{T}$ discussed in the above paragraph is the uniform kernel, following the *Principle of Maximum Entropy* (when no information about T is available, we consider the maximum entropy distribution). In specific problems, other priors, adversarial noise, *etc.*, may be more appropriate.

Remark 2. *The gradient approximation $\nabla_\theta \hat{K}_{\tilde{T}}(\theta)(x)$ is not the true gradient of $K_{\tilde{T}}$. However, this approximation is sufficient to ensure convergence of the policy π_θ to a fixed point of the operator $\langle\cdot,\tilde{T}\rangle$, provided we have a fully parametrised policy. Such an approximation is also easy to compute from sampled points $x \in X$ both on- and off-policy. Other types of policy parametrizations may also yield a fixed point of $\langle\cdot,\tilde{T}\rangle$ if it is such that we can make the policy state independent, i.e., if there is a parameter θ for which $\pi_\theta(x) = \pi_\theta(y)$ for all $x, y \in X$.*

Algorithm 1 LRPG

```
input MDP, T̃, ϵ
intialise θ, critic (if using), λ, {β¹, β², η}
set t = 0, x_t ~ μ_0
while t < max_iterations do
    perform u_t ~ π_θ(x_t)
    observe r_t, x_{t+1}
    if K̂_1(θ) not converged then k̂_1 ← K̂_1(θ)
    update critic (if using)
    update θ and λ using equation 3
output θ
```

Now, the convergence of PB-LRL algorithms is guaranteed as long as the original policy gradient algorithm (such as PPO [21] or A2C [5, 18]) for each single objective converges [37]. We can then combine Lemma 1 with these results to guarantee that Lexicographically Robust Policy Gradient (LRPG), Algorithm 1, converges to a policy that maximise robustness while remaining (approximately) optimal with respect to R.

Theorem 2. *Consider a DOMDP as in Definition 1 and let π_θ be a parametrised policy. Take $K_1(\theta) = \mathbb{E}_{x_0\sim\mu_0}[V^{\pi_\theta}(x_0)]$ to be computed through a chosen algorithm (*e.g.*, A2C, PPO) that optimises $K_1(\theta)$, and let $K_2(\theta) = -K_{\tilde{T}}(\theta)$. Given an $\epsilon > 0$, if the iteration $\theta \leftarrow \theta + \nabla_\theta \hat{K}_1$ is guaranteed to converge to a parameter set θ^* that maximises K_1, and hence J (locally or globally), then LRPG converges* a.s. *under PB-LRL conditions to parameters θ^ϵ that satisfy:*

$$\theta^\epsilon \in \arg\min_{\theta\in\Theta'} K_{\tilde{T}}(\theta) \quad \textit{such that} \quad K_1^* \geq K_1(\theta^\epsilon) - \epsilon, \quad (6)$$

where $\Theta' = \Theta$ if θ^* is globally optimal and a compact local neighbourhood of θ^* otherwise.

We reflect again on Figure 1. The main idea behind LRPG is that by formally expanding the set of acceptable policies with respect to K_1, we may find robust policies more effectively while guaranteeing a minimum performance in terms of expected rewards.

5 EXPERIMENTS

We verify the theoretical results of LRPG in a series of experiments on discrete state/action safety-related environments [7]. *Minigrid-LavaGap*, *Minigrid-LavaCrossing* are safe exploration tasks where the agent needs to navigate an environment with cliff-like regions and receives a reward of 1 when it finds a target. *Minigrid-DynamicObstacles* is a dynamic obstacle-avoidance environment where the agent is penalised for hitting an obstacle, and gets a positive reward when finding a target. *Minigrid-LavaGap* is small enough to be fully observable, and the other two environments are partially observable. In all cases observations consist of a 7×7 field of view in front of the agent, with 3 channels encoding the color and state of objects in the environment. We use A2C [40] and PPO [35] for our implementations of LRPG which we denote by LR-PPO and LR-A2C, respectively. In all cases, the lexicographic tolerance was set to $\epsilon = 0.99\hat{k}_1$ to deviate as little as possible from the primary objective. In all cases, we use objective K_T^u computed using $\tilde{T}^u = \mathrm{unif}(X)$ bounded as $\|\xi\|_\infty \leq 2$ (1.5 for LavaCrossing), and K_T^g using a normal distribution noise kernel $\tilde{T}^g$ with $\mathcal{N}(0, 0.5)$. We test the resulting policies against a noiseless environment ($\emptyset$), a kernel $T_1 = \tilde{T}^u$ and a kernel $T_2 = \tilde{T}^g$. The main point of these combinations is to also test the policies when the true noise T is similar to $\tilde{T}$. We can derive two scenarios for policy learning depending on the algorithm we want to implement and the information at hand.

General Robustness Results. Firstly, we investigate the robustness of four algorithms where we do not have a Q function. If we do not have an estimator for the critic Q^π, Proposition 1 suggests that minimising the distance between π and $\langle \pi, T \rangle$ can serve as a proxy to minimise the robustness regret. We consider the algorithms:

(1) Vanilla PPO (noiseless).
(2) LR-PPO with a uniform noise kernel (K_T^u).
(3) LR-PPO with a Gaussian noise kernel (K_T^g).
(4) SA-PPO from Zhang et al. [49].

In these experiments, we use PPO with a neural policies and value functions; the architectures and hyper-parameters used in each case can be found in Appendix C. The results are summarised in the left-hand side of Table 1. Each entry is the median of 10 independent training processes, with reward values measured as the mean of 100 independent trajectories.

Robustness through Disadvantage Objectives. If we have an estimator for the critic Q^π we can obtain robustness without inducing regularity in the policy using D^π, yielding a larger policy subspace to steer towards, and hopefully achieving policies closer to optimal. With the goal of diving deeper into the results of Theorem 1, we consider the objective:

$$K_D(\theta) := \sum_{x \in X} p^{\pi_\theta}(x) \frac{1}{2} \|D^{\pi_\theta}(x, T)\|_2^2.$$

We aim to test the hypothesis introduced through this work: by setting $K_2 = K_D$ and thus aiming to minimise the disadvantage D, we may obtain policies that yield better robustness with similar expected rewards. Observe that $\pi_D \in \Pi_D \implies K_D(\pi_D) = 0$. To test this, we compare the following algorithms on the same environments:

(1) Vanilla A2C (noiseless).
(2) LR-A2C with K_T^u.
(3) LR-A2C with K_T^g.
(4) LR-A2C with $K_2 = K_D$.

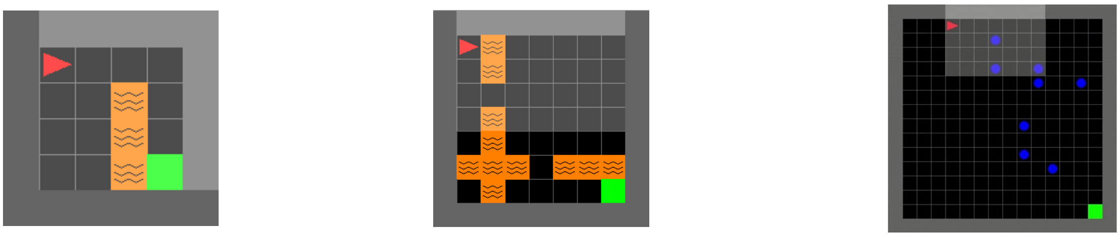
We use A2C in this case since the structure of the original cost functions are simpler than PPO, and hence easier to compare between the scenarios above, and we modified A2C to retain a Q function as a critic. With each objective function resulting in gradient descent steps that pull the policy towards different maximally robust sets ($K_T \to \Pi_T$ and $K_D \to \Pi_D$ respectively), we would expect to obtain increasing robustness for K_D. The results are presented in the right-hand side of Table 1.

6 DISCUSSION

Experiments. We applied LRPG on PPO and A2C algorithms, for a set of discrete action, discrete state grid environments. These environments are particularly sensitive to robustness problems; the rewards are sparse, and applying a sub-optimal action at any step of the trajectory often leads to terminal states with zero (or negative) reward. LRPG successfully induces lower robustness regrets in the tested scenarios, and the use of K_D as an objective (even though we did not prove the convergence of a gradient based method with such objective) yields a better compromise between robustness and rewards. When compared to recent observational robustness methods, LRPG obtains similar robustness results while preserving the original guarantees of the chosen algorithm (it even outperforms in some cases, although this is probably highly problem dependent, so we do not claim an improvement for every DOMDP).

Further Considerations on LRPG. The advantages of LRPG with respect to, for example, using POMDP results to construct optimal policies for the disturbed state case (taking the observation map to be the noise kernel) lie mainly in the situations where we are not able to construct approximations of such noise kernels. For example, if the system presents some kind of non-stationarity (in terms of noise), it may be un-feasible to try to estimate the observation map. Through LRPG, however, we can always steer the policy towards a set that is *always* maximally robust, and this may lead to improvements even in the non-stationary setting. Exploiting this consequence can be an interesting research direction.

Robustness, Complexity and Invariances. Sections 2 and 3 discuss at large the structure, shape and dependence of the maximally robust policy sets. These insights help derive optimisation objectives to use in LRPG, but there is more to be said about how policy robustness is affected by the underlying MDP properties. We hint at this in the proof of Corollary 1. More regular (*less complex* in entropy terms, or more *symmetric*) reward functions (*e.g.*, reward functions with smaller variance across the actions $R(x, \cdot, y)$) seem to induce larger robust policy sets. In other words, for a fixed policy, a *more complex* reward function yields larger robustness regrets as soon as any noise is introduced in the system.



(a) ***MiniGrid-LavaGap*** **(b)** ***MiniGrid-LavaCrossing*** **(c)** ***MiniGrid-DynamicObstacles***

Figure 2: Screenshots of the environments used.

	PPO on MiniGrid Environments				*A2C on MiniGrid Environments*			
Noise	Vanilla	LR$_{\text{PPO}}$ (K_T^u)	LR$_{\text{PPO}}$ (K_T^g)	SA-PPO	Vanilla	LR$_{\text{A2C}}$ (K_T^u)	LR$_{\text{A2C}}$ (K_T^g)	LR$_{\text{A2C}}$ (K_D)
LavaGap								
$\emptyset$	**0.95±0.003**	**0.95±0.075**	**0.95±0.101**	0.94±0.068	**0.94±0.004**	**0.94±0.005**	**0.94±0.003**	**0.94±0.006**
T_1	0.80±0.041	**0.95±0.078**	0.93±0.124	0.88±0.064	0.83±0.061	**0.93±0.019**	0.89±0.032	0.91±0.088
T_2	0.92±0.015	**0.95±0.052**	**0.95±0.094**	0.93±0.050	0.89±0.029	**0.94±0.008**	0.93±0.011	0.93±0.021
LavaCrossing								
$\emptyset$	**0.95±0.023**	0.93±0.050	0.93±0.018	0.88±0.091	0.91±0.024	0.91±0.063	0.90±0.017	**0.92±0.034**
T_1	0.50±0.110	**0.92±0.053**	0.89±0.029	0.64±0.109	0.66±0.071	**0.78±0.111**	0.72±0.073	0.76±0.098
T_2	0.84±0.061	**0.92±0.050**	**0.92±0.021**	0.85±0.094	0.78±0.054	0.83±0.105	0.86±0.029	**0.87±0.063**
DynamicObstacles								
$\emptyset$	**0.91±0.002**	**0.91±0.008**	**0.91±0.007**	**0.91±0.131**	**0.91±0.011**	0.88±0.020	0.89±0.009	**0.91±0.013**
T_1	0.23±0.201	**0.77±0.102**	0.61±0.119	0.45±0.188	0.27±0.104	0.43±0.108	0.45±0.162	**0.56±0.270**
T_2	0.50±0.117	**0.75±0.075**	0.70±0.072	0.68±0.490	0.45±0.086	0.53±0.109	0.52±0.161	**0.67±0.203**

Table 1: Reward values gained by LRPG and baselines.

This raises questions on how to use these principles to derive more robust policies in a comprehensive way, but we leave these questions for future work. Addditionally, one could quickly extend these ideas and use LRL to induce other kinds of invariances to policies. For example, one could use LRL to obtain policies that generalise to a subclass of reward functions (connecting to previous thoughts on complexity). One advantage of this approach would be that the resulting policy gradient algorithm would retain the original algorithm guarantees.

ACKNOWLEDGMENTS

Authors D. Jarne and M. Mazo want to acknowledge their work was partly funded by ERC grant SENTIENT #755953.

REFERENCES

[1] Mohammed Amin Abdullah, Hang Ren, Haitham Bou Ammar, Vladimir Milenkovic, Rui Luo, Mingtian Zhang, and Jun Wang. 2019. Wasserstein robust reinforcement learning. *arXiv preprint arXiv:1907.13196* (2019).

[2] Vahid Behzadan and Arslan Munir. 2017. Vulnerability of deep reinforcement learning to policy induction attacks. In *International Conference on Machine Learning and Data Mining in Pattern Recognition*. Springer, 262–275.

[3] Adi Ben-Israel and Bertram Mond. 1986. What is invexity? *The ANZIAM Journal* 28, 1 (1986), 1–9.

[4] Dimitri P Bertsekas. 1997. Nonlinear programming. *Journal of the Operational Research Society* 48, 3 (1997), 334–334.

[5] Shalabh Bhatnagar, Richard S. Sutton, Mohammad Ghavamzadeh, and Mark Lee. 2009. Natural Actor–critic Algorithms. *Automatica* 45, 11 (Nov. 2009), 2471–2482. https://doi.org/10.1016/j.automatica.2009.07.008

[6] V.S. Borkar and K. Soumyanatha. 1997. An analog scheme for fixed point computation. I. Theory. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 44, 4 (1997), 351–355. https://doi.org/10.1109/81.563625

[7] Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. 2018. Minimalistic Gridworld Environment for OpenAI Gym. https://github.com/maximecb/gym-minigrid.

[8] Esther Derman, Daniel Mankowitz, Timothy Mann, and Shie Mannor. 2020. A bayesian approach to robust reinforcement learning. In *Uncertainty in Artificial Intelligence*. PMLR, 648–658.

[9] Michael Everett, Björn Lütjens, and Jonathan P How. 2021. Certifiable robustness to adversarial state uncertainty in deep reinforcement

learning. *IEEE Transactions on Neural Networks and Learning Systems* (2021).
[10] Marc Fischer, Matthew Mirman, Steven Stalder, and Martin Vechev. 2019. Online robustness training for deep reinforcement learning. *arXiv preprint arXiv:1911.00887* (2019).
[11] Justin Fu, Katie Luo, and Sergey Levine. 2018. Learning Robust Rewards with Adverserial Inverse Reinforcement Learning. In *International Conference on Learning Representations*.
[12] Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. 2020. Adversarial Policies: Attacking Deep Reinforcement Learning. In *International Conference on Learning Representations*.
[13] Matthias Heger. 1994. Consideration of risk in reinforcement learning. In *Machine Learning Proceedings 1994*. Elsevier, 105–111.
[14] Roger A Horn and Charles R Johnson. 2012. *Matrix analysis*. Cambridge university press.
[15] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284* (2017).
[16] H Isermann. 1982. Linear lexicographic optimization. *Operations-Research-Spektrum* 4, 4 (1982), 223–228.
[17] Richard Klima, Daan Bloembergen, Michael Kaisers, and Karl Tuyls. 2019. Robust Temporal Difference Learning for Critical Domains. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (Montreal QC, Canada) *(AAMAS '19)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 350–358.
[18] Vijay R. Konda and John N. Tsitsiklis. 2000. Actor-critic Algorithms. In *Advances in Neural Information Processing Systems 12*, S. A. Solla, T. K. Leen, and K. Müller (Eds.). MIT Press, 1008–1014.
[19] Jernej Kos and Dawn Song. 2017. Delving into adversarial attacks on deep policies. *arXiv preprint arXiv:1705.06452* (2017).
[20] Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. 2017. Tactics of adversarial attack on deep reinforcement learning agents. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 3756–3762.
[21] Boyi Liu, Qi Cai, Zhuoran Yang, and Zhaoran Wang. 2019. Neural Trust Region/Proximal Policy Optimization Attains Globally Optimal Policy. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc.
[22] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. 2017. Adversarially robust policy learning: Active construction of physically-plausible perturbations. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 3932–3939.
[23] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. PMLR, 1928–1937.
[24] Janosch Moos, Kay Hansel, Hany Abdulsamad, Svenja Stark, Debora Clever, and Jan Peters. 2022. Robust Reinforcement Learning: A Review of Foundations and Recent Advances. *Machine Learning and Knowledge Extraction* 4, 1 (2022), 276–315.
[25] Jun Morimoto and Kenji Doya. 2005. Robust reinforcement learning. *Neural computation* 17, 2 (2005), 335–359.
[26] Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proc. of the Sixteenth International Conference on Machine Learning, 1999*.
[27] Arnab Nilim and Laurent El Ghaoui. 2005. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research* 53, 5 (2005), 780–798.
[28] Xinlei Pan, Daniel Seita, Yang Gao, and John Canny. 2019. Risk averse robust adversarial reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 8522–8528.
[29] Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. 2017. Robust deep reinforcement learning with adversarial attacks. *arXiv preprint arXiv:1712.03632* (2017).
[30] Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. 2018. Robust Deep Reinforcement Learning with Adversarial Attacks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (Stockholm, Sweden) *(AAMAS '18)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2040–2042.
[31] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. 2017. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2817–2826.
[32] Matteo Pirotta, Marcello Restelli, Alessio Pecorino, and Daniele Calandriello. 2013. Safe policy iteration. In *International Conference on Machine Learning*. PMLR, 307–315.
[33] Mark J Rentmeesters, Wei K Tsai, and Kwei-Jay Lin. 1996. A theory of lexicographic multi-criteria optimization. In *Proceedings of ICECCS'96: 2nd IEEE International Conference on Engineering of Complex Computer Systems (held jointly with 6th CSESAW and 4th IEEE RTAW)*. IEEE, 76–79.
[34] Jay K Satia and Roy E Lave Jr. 1973. Markovian decision processes with uncertain transition probabilities. *Operations Research* 21, 3 (1973), 728–740.
[35] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
[36] Joar Skalse, Matthew Farrugia-Roberts, Stuart Russell, Alessandro Abate, and Adam Gleave. 2022. Invariance in policy optimisation and partial identifiability in reward learning. *arXiv preprint arXiv:2203.07475* (2022).
[37] Joar Skalse, Lewis Hammond, Charlie Griffin, and Alessandro Abate. 2022. Lexicographic Multi-Objective Reinforcement Learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. 3430–3436. https://doi.org/10.24963/ijcai.2022/476
[38] Matthijs TJ Spaan. 2012. Partially observable Markov decision processes. In *Reinforcement Learning*. Springer, 387–414.
[39] Matthijs TJ Spaan and N Vlassis. 2004. A point-based POMDP algorithm for robot planning. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, Vol. 3. IEEE, 2399–2404.
[40] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
[41] Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems* (Denver, CO) *(NIPS'99)*. MIT Press, Cambridge, MA, USA, 1057–1063.
[42] Kai Liang Tan, Yasaman Esfandiari, Xian Yeow Lee, Soumik Sarkar, et al. 2020. Robustifying reinforcement learning agents via action space adversarial training. In *2020 American control conference (ACC)*. IEEE, 3959–3964.
[43] Chen Tessler, Yonathan Efroni, and Shie Mannor. 2019. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*. PMLR, 6215–6224.
[44] Elise van der Pol, Thomas Kipf, Frans A. Oliehoek, and Max Welling. 2020. Plannable Approximations to MDP Homomorphisms: Equivariance under Actions. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems* (Auckland, New

Zealand) *(AAMAS '20)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1431–1439.

[45] Bart PG Van Parys, Daniel Kuhn, Paul J Goulart, and Manfred Morari. 2015. Distributionally robust control of constrained stochastic systems. *IEEE Trans. Automat. Control* 61, 2 (2015), 430–442.

[46] Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. 2014. Distributionally robust convex optimization. *Operations Research* 62, 6 (2014), 1358–1376.

[47] Huan Xu and Shie Mannor. 2006. The robustness-performance tradeoff in Markov decision processes. *Advances in Neural Information Processing Systems* 19 (2006).

[48] Huan Zhang, Hongge Chen, Duane Boning, and Cho-Jui Hsieh. 2021. Robust Reinforcement Learning on State Observations with Learned Optimal Adversary. In *International Conference on Learning Representation (ICLR)*.

[49] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. 2020. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems* 33 (2020), 21024–21037.

[50] Shangtong Zhang. 2018. Modularized Implementation of Deep RL Algorithms in PyTorch. https://github.com/ShangtongZhang/DeepRL.

A EXAMPLES AND MOTIVATION

The motivation for the DOMDP setting can be framed from the perspective of using RL to control dynamical systems where we discretise the state-space. In such cases, we would perhaps still use a simulator to learn optimal policies, since it can be very costly to learn from real systems. This naturally raises the question of how to best represent the noise we may expect in the system. The learned policies are to be deployed and executed in a real system, and sensor measurements will be inevitable disturbed by a (set of) noise signal(s).

Then, observational robustness emerges as an organic problem. Consider an example where we train a robot to navigate a "dangerous" environment, where the cost of taking the wrong action results in catastrophic rewards (e.g. destruction of the robot). One could devise very mild noise signals that could appear in the real scenario, without the need for *e.g.* adversarial noise, that could induce the robot to take the wrong action by mis-judging the state it is in.

Referring back to Definition 1, it does not impose any structure on X. However, if X is a vector space and we have an additive noise perturbing the state space we can provide a clean interpretation of the stochastic kernel as follows. Consider a disturbed state $\tilde{x} = x + \xi$, where ξ is a random variable with distribution $P \in \Delta(X)$. Then, $T(y \mid x) = P\{y - x\}$. This is the case in most dynamical systems applications.

Example 1. Consider the simple MDP in Figure 3. By defining different reward functions, we can obtain the limit scenarios $\Pi_0 = \Pi$ (any policy is maximally robust) or $\Pi_0 = \Pi_T$ (only the fixed point policies are maximally robust). First, consider the reward function $R_1(x_1, \cdot, \cdot) = 10$, $R_1(x_2, \cdot, \cdot) = 0$. This produces a "dummy" MDP where all policies have the same reward sum. Then, $\forall T, \pi$, $V^{\langle \pi, T\rangle} = V^{\pi}$, and therefore we have $\Pi_D = \Pi_0 = \Pi$.

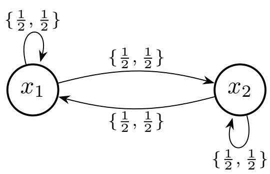


Figure 3: Example MDP. Values in brackets represent $\{P(\cdot, u_1, \cdot), P(\cdot, u_2, \cdot)\}$.

Now, consider the reward function $R_2(x_1, u_1, \cdot) = 10$, $R_2(\cdot, \cdot, \cdot) = 0$ elsewhere. Take a non-constant policy π, *i.e.*, $\pi(x_1) \neq \pi(x_2)$. In the example DOMDP (assuming the initial state is drawn uniformly from $X_0 = \{x_1, x_2\}$) one can show that at any time in the trajectory, there is a stationary probability $\Pr\{x_t = x_1\} = \frac{1}{2}$. Let us abuse notation and write $\pi(x_i) = \begin{pmatrix} \pi(x_i, u_1) & \pi(x_i, u_2) \end{pmatrix}^\top$ and $R(x_i) = \begin{pmatrix} R(x_i, u_1, \cdot) & R(x_i, u_2, \cdot) \end{pmatrix}^\top$.

For the given reward structure we have $R(x_2) = \begin{pmatrix} 0 & 0 \end{pmatrix}^\top$, and therefore:

$$J(\pi) = E_{x_0 \sim \mu_0}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] = \frac{1}{2}\langle R(x_1), \pi(x_1)\rangle \frac{\gamma}{1-\gamma}. \quad (7)$$

Since the transitions of the MDP are independent of the actions, following the same principle as in equation 7: $J\langle \pi, T\rangle = \frac{1}{2}\langle R(x_1), \langle \cdot, T\rangle(\pi)(x_1)\rangle \frac{\gamma}{1-\gamma}$. For any noise map $\langle \cdot, T\rangle \neq \mathrm{Id}$, for the two-state policy it holds that $\pi \notin \Pi_T \implies \langle \pi, T\rangle \neq \pi$. Therefore $\langle \pi, T\rangle(x_1) \neq \pi(x_1)$ and:

$$J(\pi) - J(\langle \pi, T\rangle) = \frac{5\gamma}{1-\gamma} \cdot \big(\pi(x_1, 1) - \langle \pi, T\rangle(x_1, 1)\big) \neq 0,$$

which implies that $\pi \notin \Pi_0$.

B THEORETICAL RESULTS

B.1 Auxiliary Results

Theorem 3 (Stochastic Approximation with Non-Expansive Operator). *Let $\{\xi_t\}$ be a random sequence with $\xi_t \in \mathbb{R}^n$ defined by the iteration:*

$$\xi_{t+1} = \xi_t + \alpha_t(F(\xi_t) - \xi_t + M_{t+1}),$$

where:

(1) The step sizes α_t satisfy Assumption 1.
(2) $F : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a $\|\cdot\|_\infty$ non-expansive map. That is, for any $\xi_1, \xi_2 \in \mathbb{R}^n$, $\|F(\xi_1) - F(\xi_2)\|_\infty \leq \|\xi_1 - \xi_2\|_\infty$.
(3) $\{M_t\}$ is a martingale difference sequence with respect to the increasing family of $\sigma-$fields

$$\mathcal{F}_t := \sigma(\xi_0, M_0, \xi_1, M_1, ..., \xi_t, M_t).$$

Then, the sequence $\xi_t \longrightarrow \xi^*$ almost surely *where ξ^* is a fixed point such that $F(\xi^*) = \xi^*$.*

Proof. See Borkar and Soumyanatha [6]. □

Theorem 4 (PB-LRL Convergence). *Let $\mathcal{M}$ be a multi-objective MDP with objectives K_i, $i \in \{1, ..., m\}$ of the same form. Assume a policy π is twice differentiable in parameters θ, and if using a critic V_i assume it is continuously differentiable on w_i. Suppose that if PB-LRL is run for T steps, there exists some limit point $w_i^*(\theta)$ when θ is held fixed under conditions C on $\mathcal{M}$, π and V_i. If $\lim_{T\to\infty} \mathbb{E}_t[\theta] \in \Theta_1^\epsilon$ for $m = 1$, then for any $m \in \mathbb{N}$ we have $\lim_{T\to\infty} \mathbb{E}_t[\theta] \in \Theta_m^\epsilon$ where ϵ depends on the representational power of the parameterisations of π, V_i.*

Proof. See Skalse et al. [37]. □

Lemma 1. *Let π_θ be a fully-parameterised policy in a DOMDP, and α_t a learning rate satisfying Assumption 1. Consider the following* approximated *gradient for objective $K_{\tilde{T}}(\pi)$ and sampled point $x \in X$:*

$$\nabla_\theta \hat{K}_{\tilde{T}}(\theta)(x) = (\pi_\theta(x) - \pi_\theta(y))\nabla_\theta \pi_\theta(x), \quad y \sim \tilde{T}(\cdot \mid x). \quad (8)$$

Then, the following iteration with $x \in X$ and some initial θ_0,

$$\theta_{t+1} = \theta_t - \alpha_t \nabla_\theta \hat{K}_{\tilde{T}}(\theta_t) \tag{9}$$

yields $\theta \to \tilde{\theta}$ almost surely *where $\tilde{\theta}$ satisfies $K_{\tilde{T}}(\tilde{\theta}) = 0$.*

Proof. See Appendix B.2. □

B.2 Proofs

We now present the proofs for the statements through the work.

Proposition 1. If a policy $\pi \in \Pi$ is a fixed point of the operator $\langle \cdot, T\rangle$, then it holds that $\langle \pi, T\rangle = \pi$. Therefore, one can compute the robustness of the policy π to obtain $\rho(\pi, T) = J(\pi) - J(\langle \pi, T\rangle) = J(\pi) - J(\pi) = 0 \implies \pi \in \Pi_0$. Therefore, $\Pi_T \subseteq \Pi_0$.

For a discrete state and action spaces, the space of stochastic kernels $\mathcal{K} : X \mapsto \Delta(X)$ is equivalent to the space of row-stochastic $|X| \times |X|$ matrices, therefore one can write $T(y \mid x) \equiv T_{xy}$ as the xy−th entry of the matrix T. Then, the representation of a constant policy as an $X \times U$ matrix can be written as $\overline{\pi} = 1_{|X|} v^\top$, where $1_{|X|}$ where $v \in \Delta(U)$ is any probability distribution over the action space. Observe that, applying the operator $\langle \pi, T\rangle$ to a constant policy yields:

$$\langle \overline{\pi}, T\rangle = T 1_{|X|} v^\top. \tag{10}$$

Matrix T is row-stochastic, and by the Perron-Frobenius Theorem [14] it has at least one eigenvalue $\mathrm{eig}(T) = 1$, and this admits a (strictly positive) eigenvector $T1_{|X|} = 1_{|X|}$. Therefore, substituting this in equation 10:

$$\langle \overline{\pi}, T\rangle = T 1_{|X|} v^\top = 1_{|X|} v^\top = \overline{\pi} \implies \overline{\Pi} \subseteq \Pi_T.$$

□

Proposition 2. Recall the definition in equation 1 and that the noise disadvantage function of a policy π is given by equation 4. We want to show that $D^\pi(x, T) = 0 \implies \rho(\pi, T) = 0$. Taking $D^\pi(x, T) = 0$ one has a policy that produces an disadvantage of zero when noise kernel T is applied. Then,

$$D^\pi(x, T) = 0 \implies \mathbb{E}_{u\sim\langle \pi, T\rangle(x)}[Q^\pi(x, u)] = V^\pi(x) \ \forall x \in X. \tag{11}$$

Now define the value of the disturbed policy

$$V^{\langle \pi, T\rangle}(x_0) := \mathbb{E}_{\substack{u_k\sim\langle \pi, T\rangle(x_k),\\ x_{k+1}\sim P(\cdot|x_k, u_k)}} \left[\sum_{k=0}^{\infty} \gamma^k r(x_k, u_k)\right],$$

and take:

$$V^{\langle \pi, T\rangle}(x) = \mathbb{E}_{\substack{u\sim\langle \pi, T\rangle(x),\\ y\sim P(\cdot|x,u)}} \left[r(x, u, y) + \gamma V^{\langle \pi, T\rangle}(y)\right].$$

We will now show that $V^\pi(x) = V^{\langle \pi, T\rangle}(x)$, for all $x \in X$. Observe, from equation 11 using $V^\pi(x) = \mathbb{E}_{u\sim\langle \pi, T\rangle(x)}[Q^\pi(x, u)]$, we have $\forall x \in X$:

$$\begin{aligned}
&V^\pi(x) - V^{\langle \pi, T\rangle}(x) = \\
=&\mathbb{E}_{u\sim\langle \pi, T\rangle(x)}[Q^\pi(x, u)] - \mathbb{E}_{\substack{u\sim\langle \pi, T\rangle(x)\\ y\sim P(\cdot|x,u)}} \left[r(x, u, y) + \gamma V^{\langle \pi, T\rangle}(y)\right] \\
=&\mathbb{E}_{\substack{u\sim\langle \pi, T\rangle(x)\\ y\sim P(\cdot|x,u)}} \left[r(x, u, y) + \gamma V^\pi(y) - r(x, u, y) - \gamma V^{\langle \pi, T\rangle}(y)\right] \\
=&\gamma \mathbb{E}_{y\sim P(\cdot|x,u)} \left[V^\pi(y) - V^{\langle \pi, T\rangle}(y)\right].
\end{aligned} \tag{12}$$

Now, taking the sup norm at both sides of equation 12 we get

$$\|V^\pi(x) - V^{\langle \pi, T\rangle}(x)\|_\infty = \gamma \left\|\mathbb{E}_{y\sim P(\cdot|x,u)} \left[V^\pi(y) - V^{\langle \pi, T\rangle}(y)\right]\right\|_\infty. \tag{13}$$

Observe that for the right hand side of equation 13, we have $\left\|\mathbb{E}_{y\sim P(\cdot|x,u)} \left[V^\pi(y) - V^{\langle \pi, T\rangle}(y)\right]\right\|_\infty \leq \|V^\pi(x) - V^{\langle \pi, T\rangle}(x)\|_\infty$. Therefore, since $\gamma < 1$,

$$\begin{aligned}
&\|V^\pi(x) - V^{\langle \pi, T\rangle}(x)\|_\infty \leq \gamma \|V^\pi(x) - V^{\langle \pi, T\rangle}(x)\|_\infty \implies \\
\implies &\|V^\pi(x) - V^{\langle \pi, T\rangle}(x)\|_\infty = 0.
\end{aligned}$$

Finally, $\|V^\pi(x) - V^{\langle \pi, T\rangle}(x)\|_\infty = 0 \implies V^\pi(x) - V^{\langle \pi, T\rangle}(x) = 0 \ \forall x \in X$, and $V^\pi(x) - V^{\langle \pi, T\rangle}(x) = 0 \,\forall\, x \in X \implies J(\pi) = J(\langle \pi, T\rangle) \implies \rho(\pi, T) = 0.$ □

Inclusion Theorem 1. Combining Proposition 1 and Proposition 2, we simply need to show that $\Pi_T \subset \Pi_D$. Take π to be a fixed point of $\langle \pi, T\rangle$. Then $\langle \pi, T\rangle = \pi$, and from the definition in equation 4:

$$\begin{aligned}
D^\pi(x, T) =& V^\pi(x) - \mathbb{E}_{u\sim\langle \pi, T\rangle(x,\cdot)}[Q^\pi(x, u)] \\
=& V^\pi(x) - \mathbb{E}_{u\sim\pi(x,\cdot)}[Q^\pi(x, u)] \\
=& V^\pi(x) - V^\pi(x) \\
=& 0.
\end{aligned}$$

Therefore, $\pi \in \Pi_D$, which completes the inclusions.

To show convexity of $\overline{\Pi}, \Pi_T$, first for a constant policy $\overline{\pi} \in \overline{\Pi}$, recall that we can write $\overline{\pi} = 1v^\top$, where $v \in \Delta(U)$ is any probability distribution over the action space. Now take $\overline{\pi}_1, \overline{\pi}_2 \in \overline{\Pi}$. For any $\alpha \in [0, 1]$, $\alpha\overline{\pi}_1 + (1 - \alpha)\overline{\pi}_2 = \alpha 1 v_1^\top + (1 - \alpha) 1 v_2^\top = 1(\alpha v_1 + (1 - \alpha) v_2)^\top \in \overline{\Pi}$.

At last, for the set Π_T, assume there exist two different policies π_1, π_2 both fixed points of $\langle \cdot, T\rangle$. Then, for any $\alpha \in [0, 1]$, $\langle (\alpha\pi_1 + (1 - \alpha)\pi_2), T\rangle = \alpha T\pi_1 + (1 - \alpha) T\pi_2 = \alpha\pi_1 + (1 - \alpha)\pi_2$. Therefore, any affine combination of fixed points is also a fixed point. □

Corollary 1. For statement (i), let $\overline{R}(\cdot, \cdot, \cdot) = c$ for some constant $c \in \mathbb{R}$. Then, $J(\pi) = \mathbb{E}_{x_0\sim\mu_0}[\sum_t \gamma^t \overline{r}_t \mid \pi] = \frac{c\gamma}{1-\gamma}$, which does not depend on the policy π. For any noise kernel T and policy π, $J(\pi) - J\langle \pi, T\rangle = 0 \implies \pi \in \Pi_0$.

For statement (ii) assume $\exists \pi \in \Pi_0 : \pi \notin \Pi_T$. Then, $\exists x^* \in X$ and $u^* \in U$ such that $\pi(x^*, u^*) \neq \langle \pi, T \rangle (x^*, u^*)$. Let:

$$\underline{R}(x, u, x') := \begin{cases} c & \text{if } x = x^* \text{ and } u = u^* \\ 0 & \text{otherwise} \end{cases}.$$

Then, $\mathbb{E}[R(x, \pi(x), x'] < \mathbb{E}[R(x, \langle \pi, T \rangle (x), x']$ and since the MDP is ergodic x is visited infinitely often and

$$J(\pi) - J(\langle \pi, T \rangle) > 0 \implies \pi \notin \Pi_0,$$

which contradicts the assumption. Therefore, $\Pi_0 \setminus \Pi_T = \emptyset \implies \Pi_0 = \Pi_T$. □

Lemma 1. We make use of standard results on stochastic approximation with non-expansive operators (specifically, Theorem 3 in the appendix) [6]. First, observe that for a fully parameterised policy, one can assume to have a tabular representation such that $\pi_\theta(x, u) = \theta_{xu}$, and $\nabla_\theta \pi_\theta(x) \equiv \mathrm{Id}$. We can then write the stochastic gradient descent problem in terms of the policy. Let $y \sim \tilde{T}(\cdot \mid x)$. Then:

$$\pi_{t+1}(x) = \pi_t(x) - \alpha_t \big(\pi_t(x) - \pi_t(y)\big) =$$
$$= \pi_t(x) - \alpha_t \Big(\pi_t(x) - \langle \pi_t, \tilde{T} \rangle (x) - \big(\pi_t(y) - \langle \pi_t, \tilde{T} \rangle (x)\big)\Big).$$

We now need to verify that the necessary conditions for applying Theorem 3 hold. First, α_t satisfies Assumption 1. Second, making use of the property $\|\tilde{T}\|_\infty = 1$ for any row-stochastic matrix $\tilde{T}$, for any two policies $\pi_1, \pi_2 \in \Pi$:

$$\|\langle \pi_1, \tilde{T} \rangle - \langle \pi_2, \tilde{T} \rangle\|_\infty = \|\tilde{T}\pi_1 - \tilde{T}\pi_2\|_\infty = \|\tilde{T}(\pi_1 - \pi_2)\|_\infty \leq$$
$$\leq \|\tilde{T}\|_\infty \|\pi_1 - \pi_2\|_\infty = \|\pi_1 - \pi_2\|_\infty.$$

Therefore, the operator $\langle \cdot, \tilde{T} \rangle$ is non-expansive with respect to the sup-norm. For the final condition, we have

$$\mathbb{E}_{y \sim \tilde{T}(\cdot|x)} \left[\pi_t(y) - \langle \pi_t, \tilde{T} \rangle (x) \mid \pi_t, \tilde{T}\right] =$$
$$= \sum_{y \in X} \tilde{T}(y \mid x) \pi_t(y) - \langle \pi_t, \tilde{T} \rangle (x) = 0.$$

Therefore, the difference $\pi_t(y) - \langle \pi_t, \tilde{T} \rangle (x)$ is a martingale difference for all x. One can then apply Theorem 3 with $\xi_t(x) \equiv \pi_t(x)$, $F(\cdot) \equiv \langle \cdot, \tilde{T} \rangle$ and $M_{t+1} \equiv \pi_t(y) - \langle \pi_t, \tilde{T} \rangle (x)$ to conclude that $\pi_t(x) \to \tilde{\pi}(x)$ almost surely. Finally from assumption 2, for any policy all states $x \in X$ are visited infinitely often, therefore $\pi_t(x) \to \tilde{\pi}(x) \forall x \in X \implies \pi_t \to \tilde{\pi}$ and $\tilde{\pi}$ satisfies $\langle \tilde{\pi}, \tilde{T} \rangle = \tilde{\pi}$, and $K_{\tilde{T}}(\tilde{\pi}) = 0$. □

Theorem 2. We apply the results from [37] in Theorem 4. Essentially, Skalse et al. [37] prove that for a policy gradient algorithm to lexicographically optimise a policy for multiple objectives, it is a sufficient condition that the stochastic gradient descent algorithm finds optimal parameters for each of the objectives independently. From Lemma 1 we know that a policy gradient algorithm using the gradient estimate in equation 8 converges to a maximally robust policy, *i.e.* a set of parameters $\theta' = \arg\max_\theta K_{\tilde{T}}$. Additionally, by assumption, the chosen algorithm for K_1 converges to an optimal point θ^*. While the two objective functions are not of the same form – as in [37] – the fact they are both invex [3] either locally or globally depending on the form of K_1, implies that $\hat{K}$ is also invex and hence that the stationary point θ^ϵ computed by LRPG satisfies equation 6. □

C EXPERIMENTS METHODOLOGY

We use in the experiments well-tested implementations of A2C and PPO adapted from [50] to include the computation of the lexicographic parameters in equation 3. Since all the environments use a pixel representation of the observation, we use a shared representation for the value function and policy, where the first component is a convolutional network, implemented as in Zhang [50]. The hyperparameters of the neural representations are presented in Table 2.

Layer	Output	Func.
Conv1	16	ReLu
Conv2	32	ReLu
Conv3	64	ReLu
Fc4	256	ReLu

Table 2: Shared Observation Layers

The actor and critic layers, for both algorithms, are a fully connected layer with 256 features as input and the corresponding output. We used in all cases an Adam optimiser. We optimised the parameters for each (vanilla) algorithm through a quick parameter search, and apply the same parameters for the Lexicographically Robust versions.

	LavaGap	LavaCrossing	DynObs
Steps	10^6	10^6	8×10^5
γ	0.99	0.999	0.99
α	0.001	0.001	0.001
ϵ(Adam)	10^{-8}	10^{-8}	10^{-8}
Grad. Clip	0.5	0.5	0.5
Gae	0.95	0.95	0.95
Rollout	256	512	256

Table 3: A2C Parameters

	LavaGap	LavaCrossing	DynObs
Parallel Envs	8	8	8
Steps	10^6	10^6	8×10^5
γ	0.99	0.99	0.99
α	0.001	0.001	0.001
ϵ(Adam)	10^{-8}	10^{-8}	10^{-8}
Grad. Clip	0.5	0.5	0.5
Ratio Clip	0.2	0.2	0.2
Gae	0.95	0.95	0.95
Rollout	256	512	256
Epochs	10	10	10
Entr. Weight	0	0	0

Table 4: PPO Parameters

For the implementation of the LRPG versions of the algorithms, in all cases we allow the algorithm to iterate for $1/3$ of the total steps before starting to compute the robustness objectives. In other words, we use $\hat{K}(\theta) = K_1(\theta)$ until $t = \frac{1}{3}$ max_steps, and from this point we resume the lexicographic robustness computation as described in Algorithm 1. This is due to the structure of the environments simulated. The rewards (and in particular the positive rewards) are very sparse in the environments considered. Therefore, when computing the policy gradient steps, the loss for the primary objective is practically zero until the environment is successfully solved at least once. If we implement the combined lexicographic loss from the first time step, many times the algorithm would converge to a (constant) policy without exploring for enough steps, leading to convergence towards a maximally robust policy that does not solve the environment.

Noise Kernels. We consider two types of noise; a normal distributed noise $\tilde{T}^g$ and a uniform distributed noise $\tilde{T}^u$. For the environments LavaGap and DynamicObstacles, the kernel $\tilde{T}^u$ produces a disturbed state $\tilde{x} = x + \xi$ where $\|\xi\|_\infty \leq 2$, and for LavaCrossing $\|\xi\|_\infty \leq 1.5$. The normal distributed noise is in all cases $\mathcal{N}(0, 0.5)$. The maximum norm of the noise is quite large, but this is due to the structure of the observations in these environments. The pixel values are encoded as integers $0-9$, where each integer represents a different feature in the environment (empty space, doors, lava, obstacle, goal...). Therefore, any noise $\|\xi\|_\infty \leq 0.5$ would most likely not be enough to *confuse* the agent. On the other hand, too large noise signals are unrealistic and produce pathological environments. All the policies are then tested against two "true" noise kernels, $T_1 = \tilde{T}^u$ and $T_2 = \tilde{T}^g$. The main reason for this is to test both the scenarios where we assume a *wrong* noise kernel, and the case where we are training the agents with the correct kernel.

LRPG Parameters. The LRL parameters are initialised in all cases as $\beta^1 = 2$, $\beta^2 = 1$, $\lambda = 0$ and $\eta = 0.001$. The LRL tolerance is set to $\epsilon_t = 0.99\hat{k}_1$ to ensure we never deviate too much from the original objective, since the environments have very sparse rewards. We use a first order approximation to compute the LRL weights from the original LMORL implementation.

Comparison with SA-PPO. One of the baselines included is the State-Adversarial PPO algorithm proposed in Zhang et al. [49]. We altered our implementation of PPO to incorporate the adversarial optimisation for the disturbances as described by Zhang et al. [49]. The implementation includes an extra parameter that multiplies the regularisation objective, k_{ppo}. Since we were not able to find indications on the best parameter for discrete action environments, we implemented $k_{ppo} \in \{0.1, 1, 2\}$ and picked the best result for each entry in Table 1. Larger values seemed to de-stabilise the learning in some cases. The rest of the parameters are kept as in the vanilla PPO implementation.