IMPULSE CONTROL ARBITRATION FOR DUAL SYSTEM OF EXPLOITATION AND EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Efficient reinforcement learning (RL) involves a trade-off between "exploitative" actions that maximise expected reward and "explorative" ones that lead to the visitation of "novel" states. To encourage exploration, existing methods proposed methods such as injecting stochasticity into action selection, implicit regularisation, and additive synthetic reward. However, these techniques do not necessarily offer entirely systematic approaches making this trade-off. Here we introduce **SE**lective **R**einforcement **EX**ploration (SEREX), a plug-and-play framework that casts the exploration-exploitation trade-off as a game between an RL agent—Exploiter, which purely exploits task-dependent rewards, and another RL agent—Switcher, which chooses at which states to activate a *pure exploration* policy that is trained to minimise system uncertainty and override Exploiter. Using a form of policies known as *impulse control*, Switcher is able to determine the best set of states to switch to the exploration policy while Exploiter is free to execute its actions everywhere else. We prove that SEREX converges quickly and induces a natural schedule towards pure exploitation. Through extensive empirical studies in both discrete and continuous control benchmarks, we show that with minimal modification, SEREX can be readily combined with existing RL algorithms and yields significant improvement in performance.

1 INTRODUCTION

Reinforcement learning (RL) is a framework that enables autonomous agents to learn complex behaviours through trial and error (Sutton & Barto, 2018). With the combination of neural-network based function approximations, RL has had notable successes in a number of practical domains such as robotics and games (Silver et al., 2016; Reed et al., 2022). During the training phase, instead of acting greedily all the time, the agent need to sacrifice known rewards for uncertain transitions in order to obtain a sufficient coverage of the state space for finding the globally optimal policy (Sutton & Barto, 2018). However, randomly perturbing actions is sample inefficient since it does not take into account information acquired from previous experiences. In practice, this procedure exacerbates the sample complexity of the agent's learning of the optimal policy, despite theoretically grounded asymptotic convergence (Dabney et al., 2020).

In this paper, we tackle the challenge of performing systematic and efficient exploration in RL. We propose a novel two-agent framework that disentangles exploration and exploitation for more efficient independent learning. We propose SElective Reinforcement EXploration (SEREX), which entails an interdependent interaction between an RL agent, Exploiter, whose goal is to maximise the current estimate of future task-dependent rewards (either model-free or model-based), and an additional RL agent, the Switcher, whose goal is to explore so as to reduce the system uncertainty across the state space. Furthermore, at any given state, Switcher has the power to override the Exploiter and assume control of the system (at that state) to apply exploratory actions. Therefore, the Switcher acts to reduce system uncertainty in subregions of the state space in which (high) system uncertainty exists. A key ingredient of the SEREX framework is the use of a form of policy known as *impulse control* (Øksendal & Sulem, 2007; Mguni et al., 2022) used by Switcher. This enables the Switcher to quickly determine the appropriate points to activate its exploration policy to minimise system uncertainty.

By using a two-agent framework for independent learning of the exploitation and exploration policies, the competing individual goals of completing the task set by the environment versus exploration

over the state space are decoupled and each delegated to an independent agent. This means that the Exploiter pursues its task of maximising its objective by purely exploiting without trading-off rewards from the environment for exploration. Moreover, as the Switcher itself is an RL agent, it learns to perform systematic and targeted arbitration between exploitative and exploratory actions, switching to exploration only where such actions produce a reduction in system uncertainty. This leads to stronger asymptotic behaviour moreover, as we formally prove in Sec 4, a schedule of exploration naturally emerges from SEREX without the need for heuristic exploration scheduling (see Prop. 2). SEREX, which is an independent dual system for exploitation and exploration has a nice correspondence with the neural evidence of orthogonal encoding of reward and information values in the orbitfrontal cortex (OFC, (Zhou et al., 2021) see further discussion in Sec. 7). SEREX is a flexible plug-and-play framework that can be easily integrated with existing RL algorithms. We instantiate SEREX on both value-based and actor-critic RL agents and empirically evaluate on both discrete and continuous control benchmarks. Experimental evidence indicates that SEREX lead to improved empirical performance.

2 PRELIMINARIES

In RL problems, an agent interacts with the environment such that it gradually learns to sequentially selects actions to maximise its expected returns. The underlying problem is typically formalised as an MDP $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$ where $\mathcal{S} \subset \mathbb{R}^p$ is the set of states, $\mathcal{A} \subset \mathbb{R}^k$ is the set of actions, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a transition probability function describing the system's dynamics, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function measuring the agent's performance and the factor $\gamma \in [0, 1)$ specifies the degree to which the agent's rewards are discounted over time (Sutton & Barto, 2018). At time $t \in 0, 1, \ldots,$ the system is in state $s_t \in \mathcal{S}$ and the agent must choose an action $a_t \in \mathcal{A}$ which transitions the system to a new state $s_{t+1} \sim P(\cdot|s_t, a_t)$ and produces a reward $R(s_t, a_t)$. A policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is a probability distribution over state-action pairs where $\pi(a|s)$ represents the probability of selecting action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. The goal of an RL agent is to find the optimal policy $\hat{\pi} \in \Pi$ that maximises its expected returns given by the value function: $v^\pi(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)|a_t \sim \pi(\cdot|s_t)]$ where Π is the agent's policy space.

3 SEREX: A DUAL SYSTEM FOR EXPLORATION AND EXPLOITATION

Our framework, SEREX consists of an RL agent, Exploiter and an *impulse control* agent Switcher (Mguni et al., 2022). Switcher has the ability to transfer control of the system to the exploration policy and does so at a set of states it chooses. The agent Switcher is trained to minimise the system uncertainty across the state space, hence determining the best set of states to transfer control to the exploration policy, whilst the Exploiter is free to exploit everywhere else. Unlike standard RL in which the goals of exploration and exploitation are housed within one objective / heuristic, the goal of minimising uncertainty about unexplored states and exploiting known rewards are now decoupled.

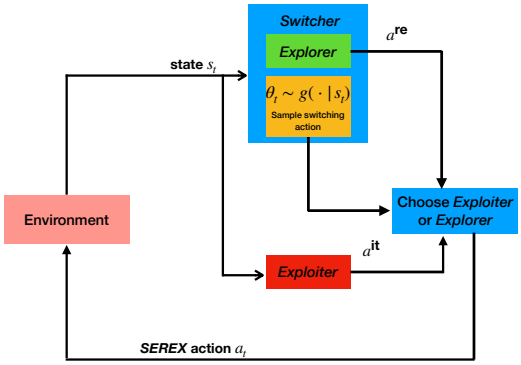


Figure 1: Schematic illustration of SEREX components and how SEREX interacts with the environment for action selection.

Formally, our framework is defined by a tuple $\mathcal{G} = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{A}^{\text{re}}, P, R^{\text{it}}, R^{\text{re}} \rangle$[1] where the new elements are the set of agents $\mathcal{N} = \{\text{Exploiter}, \text{Switcher}\}$, $\mathcal{A}$ and $\mathcal{A}^{\text{re}} \subseteq \mathcal{A}$ are the exploitation (for Exploiter) and exploration (for Switcher) action sets, respectively, and the functions $R^{\text{it}}, R^{\text{re}} : \mathcal{S} \times \mathcal{A} \times \mathcal{A}^{\text{re}} \to \mathbb{R}$ are the one-step rewards . The transition probability $P : \mathcal{S} \times \mathcal{A} \times \mathcal{A}^{\text{re}} \times \mathcal{S} \to [0, 1]$ takes the state and action of both agents as inputs. The Exploiter agent has a Markov policy $\pi^{\text{it}} : \mathcal{S} \to \mathcal{A}$, which is contained in the set $\Pi^{\text{it}} \subseteq \Pi$. The Switcher agent has two components a Markov policy $\pi^{\text{re}} : \mathcal{S} \to \mathcal{A}^{\text{re}}$ from $\Pi^{\text{re}} \subseteq \Pi$, which determines the exploration action based on a measure of uncertainty, and a (categorical) policy

[1]Note that we use $\cdot^{\text{it}}$ and $\cdot^{\text{re}}$ to refer to the quantities associated with the exploiter and explorer, respectively.

$\mathfrak{g} : \mathcal{S} \to \{0, 1\}$, which determines when to activate exploration. At each state the Switcher makes a *binary decision* to decide whether to transfer control of the system to the exploration policy π^{re}. We denote by $\{\tau_k\}_{k\geq 0}$ the timepoints at which the Switcher decides to activate the exploration policy or the *intervention times*. For example, if the Switcher chooses to switch to exploration at state s_6 and again at state s_8, then $\tau_1 = 6$ and $\tau_2 = 8$. The intervention times obey the expression $\tau_k = \inf\{t > \tau_{k-1} | s_t \in \mathcal{S}, \mathfrak{g}(s_t) = 1\}$ and are therefore ***rules** that depend on the state.* Hence, by learning an optimal $\mathfrak{g}$, Switcher learns the best states to activate exploration. As we later explain, these intervention times are determined by a condition on the state which is easy to evaluate (see Prop. 2). Here, we assume all policies $\pi^{\text{it}}, \pi^{\text{re}}, \mathfrak{g}$ are implemented with actor-critic based frameworks.

The Exploiter Objective

The goal of Exploiter is to (greedily) maximise its expected cumulative reward set by the environment. The objective that Exploiter seeks to maximise is:

$$v_1^{\pi^{\text{it}},(\pi^{\text{re}},\mathfrak{g})}(s) = \mathbb{E}\left[\sum_{t\geq 0} \gamma_1^t R^{\text{it}}\left(s_t, a_t^{\text{it}}, a_t^{\text{re}}, g_t\right) \middle| s_0 \equiv s\right], \tag{1}$$

where $a_t^{\text{it}} \sim \pi^{\text{it}}(\cdot|s_t)$ is Exploiter's action and $a_t^{\text{re}} \sim \pi^{\text{re}}$ is an action chosen according to the exploration policy, the reward function is defined by $R^{\text{it}}(s_t, a_t^{\text{it}}, a_t^{\text{re}}, g_t) = R(s_t, a_t^{\text{it}})(1 - \mathbf{1}(g_t)) + R(s, a_t^{\text{re}})\mathbf{1}(g_t)$ and $\mathbf{1}(g)$ is the indicator function which is 1 whenever $g = 1$ and 0 otherwise. Therefore, the reward received by Exploiter is $R(s_t, a_t^{\text{re}})$ when $t = \tau_k$, $k = 1, 2, \ldots$ i.e. whenever the Switcher activates the exploration policy and $R(s_t, a_t^{\text{it}})$ otherwise.

Whenever Switcher decides to transfer control to the exploration policy, the exploration policy overrides the Exploiter and the transition dynamics are affected by only the exploration policy (while Exploiter influences the dynamics at all other times). The transition dynamics are therefore given by $\boldsymbol{P}(s_{t+1}, a_t^{\text{re}}, a_t^{\text{it}}, g_t, s_t) := P(s_{t+1}, a_t^{\text{it}}, s_t)\,(1 - \mathbf{1}(g_t)) + P(s_{t+1}, a_t^{\text{re}}, s_t)\mathbf{1}(g_t)$. Therefore, the transition function is $P(s_{t+1}, a_t^{\text{re}}, s_t)$ when $t = \tau_k$, $k = 1, 2, \ldots$ i.e. whenever the Switch activates the exploration policy and $P(s_{t+1}, a_t^{\text{it}}, s_t)$ otherwise.

The Exploration Policy

The actions selected by the exploration policy π^{re} are chosen so as to maximise the following:

$$\hat{v}_2^{\pi^{\text{it}},(\pi^{\text{re}},\mathfrak{g})}(s) = \mathbb{E}\left[\sum_{t\geq 0} \gamma_2^t R^{\text{re}}\left(s_t, a_t^{\text{it}}, a_t^{\text{re}}\right) \middle| s_0 \equiv s\right], \tag{2}$$

where $R^{\text{re}}(s_t, a_t^{\text{it}}, a_t^{\text{re}}) := L(s, a_t^{\text{re}})\mathbf{1}(g_t) + L(s_t, a_t^{\text{it}})(1 - \mathbf{1}(g_t))$, and L is the measure of uncertainty which we specify in detail shortly and is chosen to satisfy the property that $L \to 0$ as the system uncertainty decreases. Analogous to the reward function for the Exploiter, the function R^{re} is defined so that the received reward is $L(s_t, a_t^{\text{re}})$ when $t = \tau_k$, $k = 0, 1, \ldots$ i.e. whenever the Switcher activates the exploration policy and $L(s_t, a_t^{\text{it}})$ otherwise.

We note that both $R^{\text{it}}\left(s_t, a_t^{\text{it}}, a_t^{\text{re}}, g_t\right)$ and $R^{\text{re}}\left(s_t, a_t^{\text{it}}, a_t^{\text{re}}, g_t\right)$ are evaluated at all timesteps, instead of independently when exploitation and exploration actions are chosen, respectively. Hence the joint agent can maximally utilise the supervisory signals available and learn to choose the optimal combination of exploitative and exploratory actions to maximise both across all timepoints.

In general, SEREX accommodates various measures of uncertainty, possible choices include the model-based ensemble epistemic uncertainty in state prediction (Chua et al., 2018) and action-prediction errors (Pathak et al., 2017). In the current setting, we focus on the model-free instantiation of the proposed framework, hence employing a model-free estimate of the uncertainty across state space. In this case, we assume that the Exploiter uses an ensemble of neural networks as its critic / value function. We quantify the uncertainty over the state space using a non-parametric estimate based on ensemble modelling of the value function of the Exploiter (Osband et al., 2016). In particular, for an ensemble of E critic estimates of $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_E\}$, we have the following measure of uncertainty for any $a \in \mathcal{A}$ and for any $s \in \mathcal{S}$:

$$L(s, a) = \frac{1}{E-1}\sum_e (\mathcal{Q}_e(s, a) - \mu(s, a))^2, \tag{3}$$

where $\mu(s,a) := \frac{1}{E}\sum_e \mathcal{Q}_e(s,a)$ is the empirical mean of the ensemble predictions.

The Switcher Mechanism

The goal of the Switcher is to minimise uncertainty over the entire state (-action) space. To induce the Switcher to selectively choose when to switch to exploration, each switch activation incurs a fixed cost for Switcher. These costs are quantified by the indicator function which is β whenever an exploratory action is performed and 0 otherwise where β is a fixed positive constant. The presence of this cost ensures that the gain for Exploration for performing an exploratory action to arrive at a given set of states is sufficiently high to merit forgoing rewards from exploitative actions. Therefore to maximise its objective, the Switcher must determine the sequence of points $\{\tau_k\}$ at which the benefit of performing an exploratory action overcomes the cost of doing so. Accordingly at time $t \in 0, 1, \ldots$, the Switcher seeks to maximise the following quantity:

$$v_2^{\pi^{\mathrm{it}},(\pi^{\mathrm{re}},\mathfrak{g})}(s) = \mathbb{E}\left[\sum_{t\geq 0}\gamma_3^t\left(R^{\mathrm{re}}\left(s_t, a_t^{\mathrm{it}}, a_t^{\mathrm{re}}\right) - \beta\cdot\mathbf{1}_{\mathcal{A}^{\mathrm{re}}}(a_t^{\mathrm{re}})\right)\right]. \tag{4}$$

Therefore to maximise its objective, the Switcher must determine the best set of states to reduce system uncertainty. Note that since R^{re} depends on the uncertainty measure L, it has the non-stationary property that $R^{\mathrm{re}} \to 0$ as system uncertainty decreases. With low level of uncertainty L, the cost of switching dominates so the Switcher does not intervene leaving Exploiter to take actions that deliver high rewards. This effectively pushes the Switcher out of the game as systemic uncertainty is reduced. This is precisely the behaviour that we seek as more about the environment becomes known. We formally prove this property in Sec. 4 (see Prop. 2). Note that both agents use deterministic policies so that the system naturally evolves towards full exploitation.

Switcher Learns Faster than Exploiter

The role of $\mathfrak{g}$ is to determine if at a given state s, the exploration policy π^{re} should be activated. In this setup, the policy π^{it} first proposes an action which is observed by the policy $\mathfrak{g}$. If activated, the policy π^{re} determines the (exploratory) action to be selected otherwise the action is selected according to the policy π^{it}. The policy $\mathfrak{g}$ involves a binary decision at each state leading to a a decision space of $\mathcal{S}\times\{0,1\}$ — this differs from Exploiter though both agents share the same experiences. Consequently, the learning process for $\mathfrak{g}$ is relative quick (and unlike Exploiter's who must optimise over a decision space which is $|\mathcal{S}|\cdot|\mathcal{A}|$, choosing an action from its action space at every state). This results in the Switcher rapidly learning when to activate π^{re}, enabling it to efficiently guide exploration during training (Mguni et al., 2022).

Relation to Other Exploratory Mechanisms

Many existing exploration models can be viewed as some degenerate form of SEREX. For instance, the classical ϵ-greedy exploration can be interpreted as SEREX with a random switching mechanism and uniform exploration policy. If we consider the case in which Switcher has the identical objective to Exploiter, consists of task rewards with additive intrinsic rewards, then the model is equivalent to exploration with intrinsic bonus (Schmidhuber, 1991; Pathak et al., 2017; Burda et al., 2018). We hope the framework of a dual system for exploration-exploitation tradeoff proposed in the current paper could inspire more systematic exploration methods currently unthought of.

Training

As we show in Sec. 4, the learning processes for both agents converge to a stable solution. Note that since $\mathcal{A}^{\mathrm{re}} \equiv \mathcal{A}^{\mathrm{it}}$, the Exploiter is trained off-policy using the data generated by both exploration and exploitation policies. The Exploiter, the Explorer and the Switcher maintain their independent replay buffers with respective reward functions. We note as the learning process progresses, the uncertainty inevitably decreases, yielding the reward structure for the exploration policy training non-stationary. To counteract the non-stationarity of the reward structure of Explorer, the discounting factor γ_2 is set to be appropriately lower (comparing to standard values) such that the agent still learns a policy that maximises future returns, but at the same time reduce the negative effects caused by the distributional shift in the reward distribution in Explorer learning.

Algorithm 1: SElective **R**einforcement **EX**ploration (SEREX)

Algorithm 1

Given reward objective function for Switcher, uncertainty objective function $\mathbf{L}(\cdot,\cdot)$, initialise Replay Buffers $\mathcal{B}^{\text{it}}$, $\mathcal{B}^{\text{re}}$, $\mathcal{B}^{\mathfrak{g}}$, Switcher intervention cost β.
for $N_{episodes}$ **do**
  Reset state s_0
  for $t = 0, 1, \ldots$ **do**
    Sample Exploiter action, $a_t^{\text{it}} \sim \pi^{\text{it}}(\cdot|s_t)$; Explorer action, $a_t^{\text{re}} \sim \pi^{\text{re}}(s_t)$; Switcher action, $g_t \sim \mathfrak{g}(s_t)$,
    if $g_t = 0$ **then**
      Apply a_t^{it} so $s_{t+1} \sim P(\cdot|a_t^{\text{it}}, s_t)$,
    else
      Apply a_t^{re} so $s_{t+1} \sim P(\cdot|a_t^{\text{re}}, s_t)$,
    end if
    Receive rewards $r_t^{\text{it}} = R(s_t, a_t^{\text{it}})$, $r_t^{\text{re}} = \boldsymbol{L}(s_t, a_t^{\text{re}})$ and $r_t^{\mathfrak{g}} = \boldsymbol{L}(s_t, a_t^{\text{it}}) - \mathbf{1}(g_t)(\boldsymbol{L}(s_t, a_t^{\text{it}}) - \boldsymbol{L}(s_t, a_t^{\text{re}}) + \beta)$.
    Store $(s_t, a_t^{\text{it}}, s_{t+1}, r_t^{\text{it}})$ in $\mathcal{B}^{\text{it}}$, store $(s_t, a_t^{\text{re}}, s_{t+1}, r_t^{\text{re}})$ in $\mathcal{B}^{\text{re}}$, store $(s_t, g_t, s_{t+1}, r_t^{\mathfrak{g}})$ in $\mathcal{B}^{\mathfrak{g}}$,
  end for
  // Learn the individual policies
  Sample batches of $|B|$ transitions, $B^i = \{(s_t^i, a_t^i, s_{t+1}^i, r_t^i\}_{b=1}^{|B|}$ from $\mathcal{B}^i$ for $i \in \{\text{it}, \text{re}, \mathfrak{g}\}$
  Update π^{it} with B^{it}, update π^{re} with B^{re}, update $\mathfrak{g}$ with $B^{\mathfrak{g}}$.
end for

4 Convergence & Optimality of SEREX

A key aspect of SEREX is the presence of two RL agents that each adapt their play according to the other's behaviour. This produces two concurrent learning processes each designed to fulfill distinct objectives. At a stable point of the learning processes the Switcher minimises uncertainty about less explored states while Exploiter maximises the environment reward. Introducing simultaneous learners can occasion issues that prevent convergence to the stable point (Zinkevich et al., 2006).

We now show $\mathcal{G}$ admits a stable point and that our method converges to it. In particular, we show that the joint system converges in its value functions for each agent. Additionally, we show that SEREX induces a natural schedule in which as the environment is explored, Switcher's interventions (to perform exploration) tend to 0 (all proofs can be found in Appendix 9). We solve these challenges with the following scheme of results:

[A] Given any Exploiter policy, the Switcher's learning process converges.

[B] The switch activations performed by Switcher can be characterised by a 'single obstacle condition' which can be evaluated online. Moreover, the number of switch activations tends to 0 as the system uncertainty decreases.

[C] The system of two joint learners (SEREX) converges, moreover, SEREX converges to an approximate solution using function approximators for the critic.

We begin by stating a key result:

Theorem 1 *SEREX converges to a stable solution in the agents' value functions.*

Theorem 1 is established by proving a series of results; firstly that for a given Exploiter policy, Switcher's learning process converges (to its optimal value function). Secondly, we show that the system of the two learners Exploiter and Switcher jointly converges to their optimal value functions.

Our first result proves the Switcher's optimal value function can be obtained as a limit point of a sequence of Bellman operations. We then prove that its convergence extends to function approximators. To begin, first define a *projection* Π by: $\Pi\Lambda := \underset{\bar{\Lambda} \in \{\Phi r | r \in \mathbb{R}^p\}}{\arg\min} \left\| \bar{\Lambda} - \Lambda \right\|$ for any function Λ.

Proposition 1 *For a given Exploiter policy $\pi \in \Pi$, the Switcher's learning process converges, moreover given a set of linearly independent basis functions $\Phi = \{\phi_1, \ldots, \phi_p\}$ where $\phi_{1 \leq k \leq p} \in L_2$, the Switcher's value function converges to a limit point $r^\star \in \mathbb{R}^p$ which is the unique solution to $\Pi\mathfrak{F}(\Phi r^\star) = \Phi r^\star$ where $\mathfrak{F}$ is defined by: $\mathfrak{F}\Lambda := R + \gamma P \max\{\mathcal{M}\Lambda, \Lambda\}$ where $\mathcal{M}$ is the Switcher's intervention operator (c.f. (13)). Moreover, $r^\star$ satisfies: $\|\Phi r^\star - Q_2^\star\| \leq (1-\gamma^2)^{-1/2} \|\Pi Q_2^\star - Q_2^\star\|$.*

Prop. 1 establishes the convergence of the Switcher's learning process with the use of a function approximator. The second statement bounds the proximity of the convergence point by the smallest approximation error that can be achieved given the choice of basis functions.

Having constructed a procedure to find the optimal Exploiter policy, our next result characterises the Switcher policy $\mathfrak{g}$ and the times that Switcher must perform an intervention.

Proposition 2 *i) For any $s \in \mathcal{S}$, the Switcher intervention times are given by the following:*

$$\tau_k = \inf\left\{\tau > \tau_{k-1} | \mathcal{M}^{\pi^{\text{re}}} v_2^{\pi^{\text{it}}, \pi^{\text{re}}} = v_2^{\pi^{\text{it}}, \Pi^{\text{re}}}\right\} \tag{5}$$

ii) Denote by $\mu_l(\mathfrak{g})$ the number of switch activations performed by the Switcher when $\max_{(s,a) \in \mathcal{S} \times \mathcal{A}} L(s,a) = l$ under the Switcher policy $\mathfrak{g}$, then $\lim_{l \to 0} \mu_l(\mathfrak{g}) = 0$.

Part i) of Prop. 2 characterises the distribution $\mathfrak{g}$. Moreover, given the function V, the times $\{\tau_k\}$ can be determined by evaluating if $\mathcal{M}V = V$ holds. Part ii) of Prop. 2 establishes that the number of switches performed by Switcher tends to 0 as the system uncertainty is reduced through Switcher's exploration. This induces a natural exploration schedule based on the current system uncertainty.

RELATION TO MARKOV GAMES.

Our framework involves a system of two agents each with their individual objectives. Settings of this kind are formalised by Markov games (MG), a framework for studying self-interested agents that simultaneously act over time (Littman, 1994). In the standard MG setup, the actions of *both* agents influence both each agent's rewards and the system dynamics. Therefore, each agent $i \in \{1, 2\}$ has its own reward function $R_i : \mathcal{S} \times (\times_{i=1}^2 \mathcal{A}_i) \to \mathbb{R}$ and action set $\mathcal{A}_i$ and its goal is to maximise its *own* expected returns. The system dynamics, now influenced by both agents, are described by a transition probability $P : \mathcal{S} \times (\times_{i=1}^2 \mathcal{A}_i) \times \mathcal{S} \to [0, 1]$. Unlike classical MGs, in our MG, Switcher does not intervene at each state but is allowed to assume control of the system at certain states which it decides using impulse controls. Our setup is related to stochastic differential games with impulse control (Mguni, 2018). However, our Markov Game differs markedly since it is nonzero-sum, an agent *assumes control* and is a discrete-time treatment.

5 RELATED WORK

Exploration-Exploitation Tradeoff is a fundamental question in RL research, i.e. trading off finding higher reward states and exploiting known rewards. Existing approaches include directly injecting pure noise or certain parametric stochasticity into action choices during learning (Sutton & Barto, 2018; Lillicrap et al., 2015); using stochastic controllers regularised by the maximum entropy principle (Haarnoja et al., 2018); augmenting task rewards with synthetic exploration bonus / intrinsic reward (Stadie et al., 2015; Pathak et al., 2017; Sekar et al., 2020). Despite the simplicity, no existing methods explicitly learn an exploration policy for performing targeted explorative actions that maximise the expected uncertainty over the future states. Moreover, most existing methods utilise one policy for capturing both the task-dependent optimal behaviour and the explorative behaviour for efficient covering of the state space, yielding suboptimal learning in both aspects, whereas SEREX is able to disentangle the learning of the two policies with independently trained RL agents, leading to improved training for both the optimal policy and the exploration policy. Among more systematic approaches , is exploration according to 'Optimism in the Face of Uncertainty' (OFU). Some popular algorithms under the OFU framework include the Upper Confidence Bound (UCB) algorithm Auer (2002) that achieves theoretically justified regret bounds and active inference algorithms that relate exploration with free energy maximisation under the variational inference principle (Schwartenbeck et al., 2013). However such systematic approaches predominantly exist in the multit-armed bandits literature, whereas SEREX addresses reinforcement learning in Markov decision processes.

Reward free exploration, also known as the task-agnostic or reward-agnostic setting is a closely related method (Zhang et al., 2020; Jin et al., 2020). In this setting, the agent goes through a two-stage process. In the exploration phase the agent interacts with the environment without the guidance of any reward information, and in the planning phase the reward information is revealed and the agent computes a policy based on the transition information collected in the exploration phase and the reward information revealed in the planning phase. We also separate the tasks of exploration from exploitation using two processes, however we note that here the two processes are performed concurrently and actions are chosen based on either process interchangeably, hence achieving a more self-contingent tradeoff between exploration and exploitation.

Uncertainty quantification in exploration is an active field of research in RL. It is common practice to use the disagreement of the predictions over an ensemble of neural networks as the epistemic uncertainty to guide exploration Osband et al. (2016); Janner et al. (2019); Sekar et al. (2020); Lee et al. (2021). Connections between ensemble disagreement and information theory have been drawn such that choosing actions that maximises the expected ensemble disagreement would maximally increase the information gain, improving the efficiency of exploration Sekar et al. (2020); O'Donoghue (2021). Other popular alternatives involve the prediction error of a discriminative dynamics model Schmidhuber et al. (1997); Pathak et al. (2017) and the predictive uncertainty given a generative dynamics model Ratzlaff et al. (2020); Jiang & Lu (2020). We wish to note that existing works utilise the uncertainty as the additive intrinsic reward that facilitate implicit exploration towards less explored states, and the single augmented reward fails to learn optimal exploitation or exploration policies.

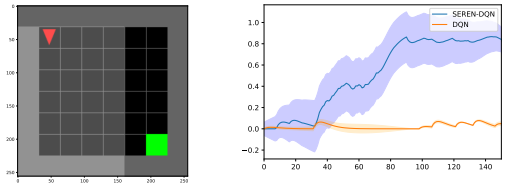
6 Experiments

We performed a series of experiments to demonstrate that SEREX's multi-player framework is able to improve the tradeoff between exploration and exploitation leading to marked improvement of the underling RL methods (all experimental details can be found in Appendix 11).

Specifically, we wish to address the question that if SEREX learns to improve performance of an underlying base RL learner by more efficiently locating higher reward states in MDPs with a) discrete b) continuous action spaces with different base learners (e.g., value-based and actor-critic). Moreover, we wish to empirically investigate if the non-stationary exploration reward structure negatively impact the overall learning (hence justifying our choice of lower discounting value for the Explorer).

6.1 MiniGrid Experiments

We firstly demonstrate SEREX in combination with a standard DQN (Mnih et al., 2013). It is well known that DQN usually performs poorly in sparse-reward settings (Osband et al., 2016; Pathak et al., 2017). To this end, we choose the MiniGrid environments (Chevalier-Boisvert et al., 2018), where all transitions to non-goal states leads to zero reward. As we observe in Figure 2(b), SEREX-DQN quickly learns to consistently navigate towards the goal state within 100 training episodes, whereas the standard DQN with ϵ-greedy has failed to acquire a sensible policy over the 150 episodes. Hence we conclude that SEREX can be readily plugged into DQNs to deal with sparse-reward and/or goal-directed tasks.

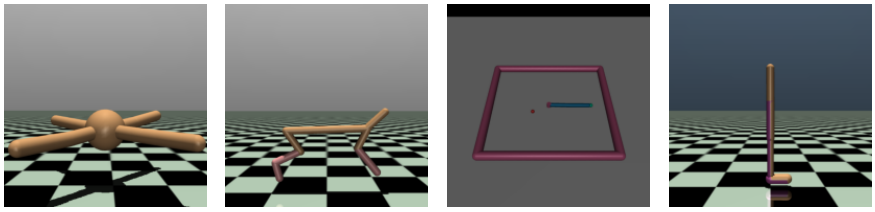


(a) (b)

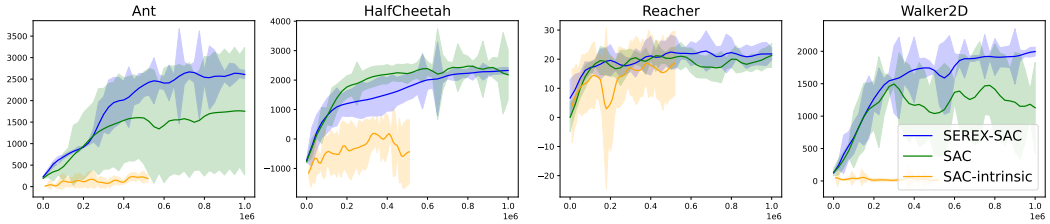
Figure 2: DQN and SEREX-DQN in the MiniGrid-World Chevalier-Boisvert et al. (2018). (a) Graphical illustration of the "8x8" minigrid-world environment, only transitions into goal states (green block) lead to non-zero rewards; (b) SEREX-DQN quickly learns the optimal policy to the goal state while the standard DQN has not learned good policy over 150 episodes of training.

6.2 MuJoCo Experiments

We evaluate SEREX on the continuous control benchmarks from the MuJoCo suite (Figure 3(a); Todorov et al. (2012)) to show that SEREX yields a more efficient explorative strategy that enables more sample-efficient and stronger learning of the optimal (exploitation) policy. We choose to implement SEREX on the Soft Actor-Critic (SAC; Haarnoja et al. (2018)). SAC is an off-policy policy gradient algorithm



(a)



(b)

Figure 3: Evaluation of SEREX with the baseline SAC algorithm on the MuJoCo tasks (Todorov et al., 2012). (a) Graphical demonstration of the selected MuJoCo environments; (b) Average evaluation returns (over 5 random seeds) of SEREX-SAC and SAC over 1×10^6 training steps.

where the policy is trained under the maximum entropy principle, and it achieves state-of-the-art performance across a number of continuous control benchmarks.

From Figure 3(b), we observe that in 3 out of the 4 selected tasks, SEREX-SAC outperforms the baseline SAC agent in terms of both sample efficiency and the asymptotic performance, over the first 1×10^6 training steps. SEREX-SAC performs slightly worse than the baseline SAC on the HalfCheetah task in terms of sample efficiency of learning, but reaches similar asymptotic performance. Moreover, we note that across all 4 tasks, SEREX-SAC leads to more robust training, as indicated by the standard deviation of the evaluation scores over 5 random seeds throughout training. The gain may be attributed to the effective exploration by the Switcher, especially during the early phase of training, which facilitates the diversity of the off-policy replay buffer, hence enabling the identifying of better solutions.

In order to disentangle the contribution to the performance improvement with respect to the uncertainty-based reward signal for guiding exploration and the learned impulse switching control mechanism for the arbitration between exploration and exploitation, we implement SAC-intrinsic, which utilise the ensemble epistemic uncertainty (Eq. 3) as the additive intrinsic reward for incentivising exploration towards less explored states. In Figure 3(b) we observe that merely including the additive uncertainty-based intrinsic reward leads to worse performance across all tasks, and SEREX-SAC outperforms SAC-intrinsic on all 4 selected mujoco tasks. Hence we have empirically justified that the impulse swtiching control arbitration enables the learning of more targeted exploration policies comparing to the naive additive combination of the extrinsic reward and the uncertainty-based exploration bonus.

6.2.1 Ablation Studies on the Explorer discounting factor

As discussed in Sec. 3, the discounting factor for the Explorer needs to be set small to mitigate the negative effects of the non-stationarity reward structured in the training of Explorer. However, naively setting the discounting factor too small would not yield good performance either, where the resulting agent takes exploratory actions only dependent on the epistemic uncertainty of the current state instead of a value estimate that guides targeted exploration towards areas of high uncertainties. Here we empirically justify our hypothesis by performing an ablation study on the effect of the value of the discounting factor for the learning of the Explorer. By examining the asymptotic performance given 1×10^6 training stes (Table 1), we see that setting the discounting factor too large or too small both induce worse performance, whereas intermediate values of γ_2 (0.6) yields the best performance. Noticeably, we also observe that setting γ_2 too large or too small would yield the training less robust with respect to the random seed, leading to increased noise in the evaluation performance.

	Ant	HalfCheetah	Reacher	Walker2D
SEREX-SAC ($\gamma_2 = 0.6$)	$\mathbf{2607.6 \pm 99.0}$	$\mathbf{2327.8 \pm 102.5}$	$\mathbf{21.8 \pm 2.0}$	$\mathbf{1996.1 \pm 69.9}$
SEREX-SAC ($\gamma_2 = 0.1$)	1853.7 ± 1362.7	2205.4 ± 846.2	20.2 ± 4.0	1169.7 ± 762.9
SEREX-SAC ($\gamma_2 = 0.98$)	2477.9 ± 128.9	1944.9 ± 587.0	18.6 ± 0.8	1720.7 ± 177.6

Table 1: Ablation studies on the effects of Explorer's discounting factor (γ_2).

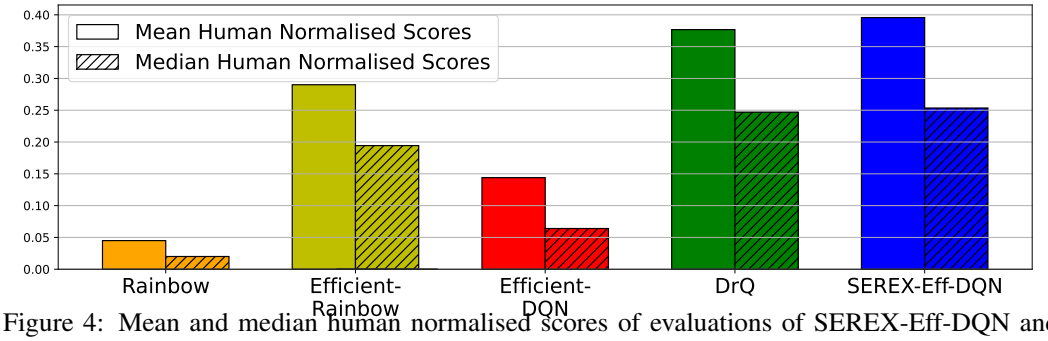


Figure 4: Mean and median human normalised scores of evaluations of SEREX-Eff-DQN and selected baselines (see Appendix 11) on Atari 100K benchmarks.

6.3 Atari Experiments

We further evaluate SEREX on a set of sample-constrained discrete control tasks, the Atari 100K benchmark (Kaiser et al., 2019). Here we instantiate SEREX based on DQN (Mnih et al., 2013), with additional components, including double Q-learning (Van Hasselt et al., 2016), dueling network for value estimation (Wang et al., 2016), and multi-step TD-target (Mnih et al., 2016). We additionally utilise the image augmentation techniques for training the DQN (Kostrikov et al., 2020) (we only use the "intensity" augmentation instead of all augmentation types as in Kostrikov et al. (2020)). We apply the resulting model, SEREX-Eff-DQN on all games in the Atari 100K benchmark, and we evaluate the performance given 100K training steps. We follow the evaluation procedures outlined in Kaiser et al. (2019). In Figure 4, we show that SEREX-Eff-DQN outperforms all selected baselines (see Appendix 11) in terms of both the median and mean human normalized returns for all SEREX-Eff-DQN and selected baselines, hence again indicating the improvement brought by the SEREX framework. Full experimental setup and results (for all games) can be found in Appendix 11 and 12.

7 Discussion

We introduced SEREX, a plug-and-play framework that seeks to learn the optimal arbitration between exploitative and exploratory behaviours using an impulse control mechanism. SEREX can be readily combined with existing value-based and actor-critic algorithms, here we demonstrate the instantiations of SEREX given DQN and SAC, but more combinations can be considered for future works. We formulate the problem of the arbitration between the exploration and the exploitation policies under a Markov game framework, where the Exploiter seeks to only maximise the cumulative return and the Explorer seeks to minimise the epistemic uncertainty of the Exploiter's value estimate over the state space. We provide theoretical justification for the convergence of SEREX to the optimal achievable value estimates with linear function approximator. We demonstrate the utility of SEREX through extensive experimental studies on continuous control benchmarks. When implemented with state-of-the-art policy gradient algorithms (SAC), we show that the SEREX-augmented agents consistently yield improvement in terms of sample efficiency and asymptotic performance with respect to the baseline agents. We also showed that SEREX can be combined with value-based algorithms such as DQN, and yield improvement on the Atari 100K benchmarks over competitive baseline algorithms.

Behaviourally, animals tend to sacrifice short-term rewards to obtain information gain in uncertain environments (Bromberg-Martin & Hikosaka, 2009; Gottlieb et al., 2013). Blanchard et al. (2015) demonstrated that the OFC neurons have firing correlated with both information value and primary value signals. Instead of integrating these variables to code subjective value, they found that OFC neurons tend to encode the two signals in an orthogonal manner. Hence despite being behaviourally similar, the dual system of independent value representation and learning in SEREX may provides a more biologically plausible framework for exploration-exploitation tradeoff than intrinsic exploration based on the combination of extrinsic primary reward and intrinsic estimate of information value.

REFERENCES

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Albert Benveniste, Michel Métivier, and Pierre Priouret. *Adaptive algorithms and stochastic approximations*, volume 22. Springer Science & Business Media, 2012.

Tommy C Blanchard, Benjamin Y Hayden, and Ethan S Bromberg-Martin. Orbitofrontal cortex uses distinct codes for different choice attributes in decisions motivated by curiosity. *Neuron*, 85(3): 602–614, 2015.

Vivek S Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5): 291–294, 1997.

Ethan S Bromberg-Martin and Okihide Hikosaka. Midbrain dopamine neurons signal preference for advance information about upcoming rewards. *Neuron*, 63(1):119–126, 2009.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in Neural Information Processing Systems*, 31, 2018.

Will Dabney, Georg Ostrovski, and André Barreto. Temporally-extended {\epsilon}-greedy exploration. *arXiv preprint arXiv:2006.01782*, 2020.

Jacqueline Gottlieb, Pierre-Yves Oudeyer, Manuel Lopes, and Adrien Baranes. Information-seeking, curiosity, and attention: computational and neural mechanisms. *Trends in cognitive sciences*, 17 (11):585–593, 2013.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Tommi Jaakkola, Michael I Jordan, and Satinder P Singh. Convergence of stochastic iterative dynamic programming algorithms. In *Advances in neural information processing systems*, pp. 703–710, 1994.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, pp. 12519–12530, 2019.

Jiechuan Jiang and Zongqing Lu. Generative exploration and exploitation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4337–4344, 2020.

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pp. 4870–4879. PMLR, 2020.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 6131–6141. PMLR, 2021.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

David Mguni. A viscosity approach to stochastic differential games of control and stopping involving impulsive control. *arXiv preprint arXiv:1803.11432*, 2018.

David Mguni. Cutting your losses: Learning fault-tolerant control and optimal stopping under adverse risk. *arXiv preprint arXiv:1902.05045*, 2019.

David Mguni, Aivar Sootla, Juliusz Ziomek, Oliver Slumbers, Zipeng Dai, Kun Shao, and Jun Wang. Timing is everything: Learning to act selectively with costly actions and budgetary constraints. *arXiv preprint arXiv:2205.15953*, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

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

Brendan O'Donoghue. Variational bayesian reinforcement learning with regret bounds. *Advances in Neural Information Processing Systems*, 34, 2021.

Bernt Karsten Øksendal and Agnes Sulem. *Applied stochastic control of jump diffusions*, volume 498. Springer, 2007.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29:4026–4034, 2016.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, pp. 2778–2787, 2017.

Neale Ratzlaff, Qinxun Bai, Li Fuxin, and Wei Xu. Implicit generative modeling for efficient exploration. In *International Conference on Machine Learning*, pp. 7985–7995. PMLR, 2020.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Jürgen Schmidhuber. Curious model-building control systems. In *Proc. international joint conference on neural networks*, pp. 1458–1463, 1991.

Jürgen Schmidhuber, Jieyu Zhao, and Marco Wiering. Shifting inductive bias with success-story algorithm, adaptive levin search, and incremental self-improvement. *Machine Learning*, 28(1): 105–130, 1997.

Philipp Schwartenbeck, Thomas FitzGerald, Ray Dolan, and Karl Friston. Exploration, novelty, surprise, and free energy minimization. *Frontiers in psychology*, 4:710, 2013.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pp. 8583–8592. PMLR, 2020.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

John N Tsitsiklis and Benjamin Van Roy. Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives. *IEEE Transactions on Automatic Control*, 44(10):1840–1851, 1999.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Hado P Van Hasselt, Matteo Hessel, and John Aslanides. When to use parametric models in reinforcement learning? *Advances in Neural Information Processing Systems*, 32, 2019.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pp. 1995–2003. PMLR, 2016.

Xuezhou Zhang, Adish Singla, et al. Task-agnostic exploration in reinforcement learning. *arXiv preprint arXiv:2006.09497*, 2020.

Jingfeng Zhou, Chunying Jia, Marlian Montesinos-Cartagena, Matthew PH Gardner, Wenhui Zong, and Geoffrey Schoenbaum. Evolving schema representations in orbitofrontal ensembles during learning. *Nature*, 590(7847):606–611, 2021.

Martin Zinkevich, Amy Greenwald, and Michael Littman. Cyclic equilibria in Markov games. *Advances in Neural Information Processing Systems*, 18:1641, 2006.