Received: 3 February 2022 | Accepted: 14 June 2022

DOI: 10.1111/2041-210X.13954

Methods in Ecology and Evolution BRITISH ECOLOGICAL SOCIETY

RESEARCH ARTICLE

Deep reinforcement learning for conservation decisions

Marcus Lapeyrolerie | **Melissa S. Chapman** | **Kari E. A. Norman** | **Carl Boettiger**

Department of Environmental Science, Policy, and Management, University of California, Berkeley, Berkeley, California, USA

Correspondence
Carl Boettiger
Email: cboettig@berkeley.edu

Funding information
National Science Foundation, Grant/Award Number: DBI-1942280

Handling Editor: Tamara Münkemüller

Abstract

1. Can machine learning help us make better decisions about a changing planet? In this paper, we illustrate and discuss the potential of a promising corner of machine learning known as deep reinforcement learning (RL) to help tackle the most challenging conservation decision problems. We provide a conceptual and technical introduction to deep RL as well as annotated code so that researchers can adopt, evaluate and extend these approaches.
2. RL explicitly focuses on designing an agent who interacts with an environment that is dynamic and uncertain. Deep RL is the subfield of RL that incorporates deep neural networks into the agent. We train deep RL agents to solve sequential decision-making problems in setting fisheries quotas and managing ecological tipping points.
3. We show that a deep RL agent is able to learn a nearly optimal solution for the fisheries management problem. For the tipping point problem, we show that a deep RL agent can outperform a sensible rule-of-thumb strategy.
4. Our results demonstrate that deep RL has the potential to solve challenging decision problems in conservation. While this potential may be compelling, the challenges involved in successfully deploying RL-based management to realistic scenarios are formidable—the required expertise and computational cost may place these applications beyond the reach of all but large, international technology firms. Ecologists must establish a better understanding of how these algorithms work and fail if we are to realize this potential and avoid the pitfalls such a transition would bring. We ultimately set forth a research framework based on well-posed, public challenges so that ecologists and computer scientists can collaborate towards solving hard decision-making problems in conservation.

KEYWORDS
artificial intelligence, conservation, machine learning, reinforcement learning, tipping points

1 | INTRODUCTION

Advances in both available data and computing power are opening the door for machine learning (ML) to play a greater role in addressing some of our planet's most pressing environmental problems. But will ML approaches really help us tackle our most pressing environmental problems? From the growing frequency and intensity of wildfire (Moritz et al., 2014), to over-exploited fisheries (Worm et al., 2006) and declining biodiversity (Dirzo et al., 2014), to emergent zoonotic pandemics (Dobson et al., 2020), the diversity and scope of environmental problems are unprecedented. Applications of ML in ecology have to-date illustrated the promise of two methods: *supervised learning* (Joseph, 2020) and *unsupervised learning* (Valletta et al., 2017). However, the fields of ecology and conservation have largely overlooked the third and possibly most promising approach in the ML triad: *reinforcement learning* (RL). Three features distinguish RL from other ML methods in ways that are particularly well suited to addressing issues of global ecological change:

1. RL is explicitly focused on the task of selecting actions in an uncertain and changing environment to maximize some objective.
2. RL does not require massive amounts of representative sampled historical data.
3. RL approaches easily integrate with existing ecological models and simulations, which may be our best guide to understanding and predicting future possibilities.

Despite relevance to decision making under uncertainty that could make RL uniquely well suited for ecological control, RL has only been applied to this field in a few cases (Fonnesbeck, 2008; Silvestro et al., 2022; Treloar et al., 2020; Xu et al., 2021). To date, the problems considered by RL research have largely been drawn from examples in robotic movement and games like Go and Starcraft (OpenAI et al., 2018; Silver et al., 2018; Vinyals et al., 2019). Complex environmental problems share many similarities to these tasks and games: the need to plan many moves ahead given a large number of possible outcomes, to account for uncertainty and to respond with contingency to the unexpected. RL agents typically develop strategies by interacting with simulators, a practice that should not be unsettling to ecologists, since learning from simulators is common across ecology. Rich, processes-based simulations such as the SORTIE model in forest management (Pacala et al., 1996), Ecopath with Ecosim in fisheries management (Steenbeek et al., 2016) or climate change policy models (Nordhaus, 1992) are already used to explore scenarios and inform ecosystem management. Decision-theoretic approaches based on optimal control techniques can only find the best strategy in the simplest of ecological models; the so called "curse of dimensionality" makes problems with a large number of states or actions intractable by conventional methods (Chades et al., 2021; Ferrer-Mestres et al., 2021; Marescot et al., 2013; Wilson et al., 2006). Neural-network-based RL techniques, referred to as *deep RL*, have proven particularly effective in problems involving complex, high-dimensional spaces that have previously proven intractable to classical methods.

While deep RL may have the potential to open up such intractable problems, it also risks making those problems tractable only for stakeholders with access to extensive computational resources and expertise. It is notable that the landmark advances cited above have been solved not by academic teams but by specialized research teams of international technology firms such as Alphabet. Precise estimates of computational resources used in that research are difficult to establish, but previous estimates benchmarked against commercially available cloud computing platforms place the training of a single model at over $35 million (Hernandez & Brown, 2020; Huang, 2018; Silver et al., 2017), and many realistic ecological problems will involve even greater complexity than these landmark examples (OpenAI et al., 2018; Silver et al., 2017, 2018). While the history of improved efficiency in computing technology has shown a remarkable ability to reduce such barriers, it has simultaneously moved the leading edge of those capabilities farther beyond the reach of traditional ecological research. We believe that ecologists must seek to better understand the design, capabilities and limitations of these algorithms while keeping in mind that the application of RL to conservation will surely require the ambitious collaboration, resources and expertise on par with the scale of the immense environmental and ecological problems we face.

In this paper, we draw on examples from fisheries management and ecological tipping points to illustrate how deep RL techniques can successfully discover optimal solutions to previously solved management scenarios and discover highly effective solutions to unsolved problems. We focus on examining the potential and limitations of deep RL through the lens of simple, classical models. Over a century of theory and practice in ecology has demonstrated that simple models can provide meaningful insights, which improve management outcomes (Getz et al., 2018). As Richard Levins successfully established in his classic paper on the principles of model building (Levins, 1966), model complexity must not be mistaken for model realism. Levins espoused simple mechanistic models that satisfy the goals of being both *realistic* and *general*. More complex models such as those used in fisheries to guide the management of specific stocks typically sacrifice *generality* for *precision*. Such simple, realistic and general models are still the bedrock of most theory and practice today (for instance, the notion of maximum sustainable yield, MSY, in fisheries, or R_0 in epidemiology, remain important concepts in management). These models provide an ideal first benchmark for evaluating the performance of emerging methods of deep RL for several reasons: Firstly, for some cases, the optimal solution is already known, providing a clear standard-of-comparison to evaluate RL performance. Prior work sometimes overlooks this essential step, assuming that whatever behaviour an RL agent produces is sufficiently optimal (Mnih et al., 2015). As our evaluations will illustrate, such an assumption can be quickly misleading. Second, these models are already widely studied and will be familiar to many readers: Schaefer (1954) is a staple of fisheries management textbooks and practice, with over 2800 citations, while May (1977) has

become a canonical model of thresholds and tipping points, which still continues to dominate how many ecologists think about these phenomena (Scheffer et al., 2015). Many readers can thus benefit from existing knowledge and intuition about the behaviour and implications of these models in interpreting the performance of deep RL, something that would not be possible with a more complex model. Third, these models include or can easily be extended to contexts for which the optimal management policy is unknown or inaccessible to classical methods. Our implementations of these models have been published to the python-based PyPi code archive and include many such variations that represent open problems for RL. We include extensive appendices with carefully annotated code, which should allow readers to both reproduce and extend this analysis.

This paper does not intend to validate deep RL as a method that should be used to directly inform decision-making on current conservation problems. Rather, we seek to provide ecologists with a greater understanding of both potentials and pitfalls of this emerging approach. We have selected familiar example problems to provide ecologists with a greater background and intuition to understand these techniques and engage in the collaborative development of deep RL-based methods, while also highlighting challenges that ecological problems pose to existing techniques. Validating deep RL for current conservation problems is beyond the scope of any one paper: this will necessitate examining a range of more "precise" models, which will require more computational resources than that available to most researchers and extensive collaboration between large teams of ecologists and computer scientists.

2 | MATERIALS AND METHODS

All applications of RL can be divided into two components: an *environment* and an *agent*. The *environment* is typically a computer simulation, though it is possible to use the real world as the RL environment (Ha et al., 2020). The *agent*, which is often a computer program, continuously interacts with the environment. At each time step, the agent *observes* the current *state* of the environment and then performs an *action*.[1] As a result of this action, the environment transitions to a new state and transmits a numerical *reward* signal to the agent (Figure 1). The goal of the agent is to learn how to maximize its expected cumulative reward. The agent learns how to achieve this objective during a period called *training*. In training, the agent *explores* the available actions. Once the agent comes across a highly rewarding sequence of observations and actions, the agent will reinforce this behaviour so that it is more likely for the agent to *exploit* the same high reward trajectory in the future. Throughout this process, the agent's behaviour is codified into what is called a *policy*, which describes what action an agent should take for a given observation.

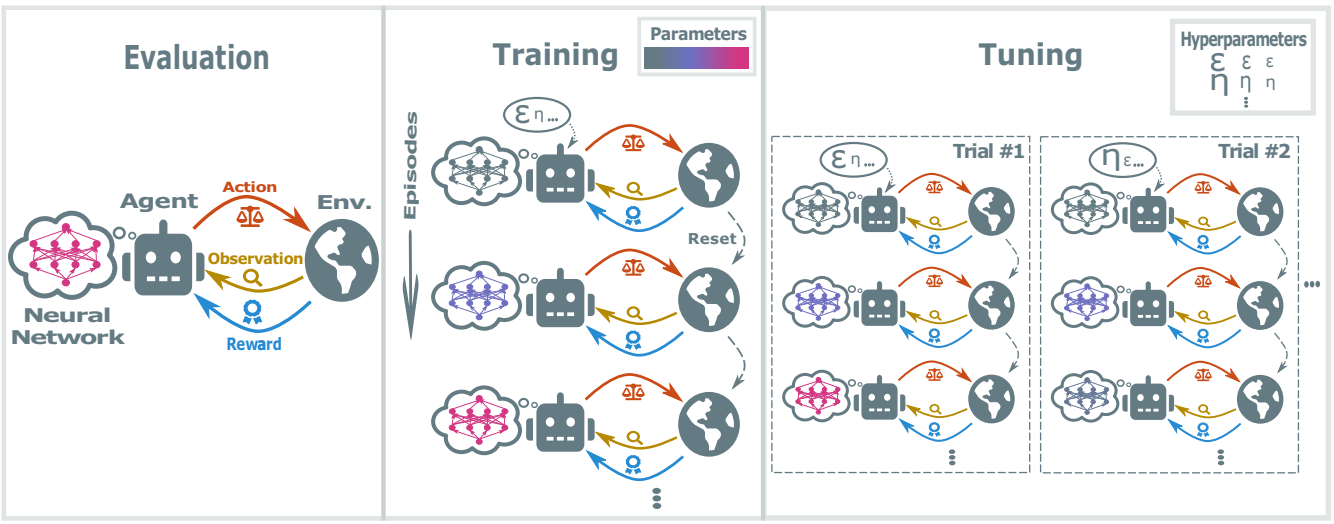

FIGURE 1 Deep reinforcement learning: A deep RL *agent* uses a *neural network* to select an *action* in response to an *observation* of the *environment*, and receives a *reward* from the environment as a result. During *training*, the agent tries to maximize its cumulative reward by interacting with the environment and learning from experience. In the RL loop, the agent performs an action, then the environment returns a reward and an observation of the environment's state. The agent-environment loop continues until the environment reaches a terminal state, after which the environment will reset, causing a new *episode* to begin. Across training episodes, the agent will continually update the *parameters* in its neural network, so that the agent will select better actions. Before training starts, the researcher must input a set of *hyperparameters* to the agent; hyperparameters direct the learning process and thus affect the outcome of training. A researcher finds the best set of hyperparameters during *tuning*. Hyperparameter tuning consists of iterative *trials*, in which the agent is trained with different sets of hyperparameters. At the end of a trial, the agent is evaluated to see which set of hyperparameters results in the highest cumulative reward. An agent is *evaluated* by recording the cumulative reward over one episode, or the mean reward over multiple episodes. Within evaluation, the agent does not update its neural network; instead, the agent uses a trained neural network to select actions

2.1 | RL environments

An environment is a mathematical function, computer program or real world experience that takes an agent's proposed *action* as input and returns an *observation* of the environment's current *state* and an associated *reward* as output. In contrast to classical approaches (Chades et al., 2021; Marescot et al., 2013), there are few restrictions on what comprises a state or action. States and actions may be continuous or discrete, completely or partially observed, and single or multidimensional. The main focus of building an RL environment, however, is on the environment's transition dynamics and reward function. The designer of the environment can make the environment follow any transition and reward function provided that both are functions of the current state and action. The ability to tailor the actions, states, transition dynamics and reward function allows RL environments to model a broad range of decision making problems. For example, we can set the transitions to be deterministic or stochastic. We could map any countable set of actions to a discrete action space. We can also specify the reward function to be *sparse*, whereby a positive reward can only be received after a long sequence of actions, for example, the end point in a maze. In other environments, an agent may have to learn to forgo immediate rewards (or even accept an initial negative reward) in order to maximize the net discounted reward as we illustrate in examples here.

The OpenAI `gym` software framework was created to address the lack of standardization of RL environments and the need for better benchmark environments to advance RL research (Brockman et al., 2016). The `gym` framework defines a standard interface and methods by which a developer can describe an arbitrary environment in a computer program. This interface allows for the application of software agents that can interact and learn in that environment without knowing anything about the environment's internal details. Using the `gym` framework, we turn existing ecological models into valid environmental simulators that can be used with any RL agent. In Appendix C, we give detailed instruction on how an OpenAI `gym` is constructed.

2.2 | Deep RL agents

To optimize the RL objective, agents either take a *model-free* or *model-based* approach. The distinction is that *model-free* algorithms do not attempt to learn or use a predictive model of the environment; yet, *model-based* algorithms employ a predictive model of the environment to achieve the RL objective. A trade-off between these approaches is that when it is possible to quickly learn a model of the environment or the model is already known, model-based algorithms tend to require much less interaction with the environment to learn good-performing policies (Janner et al., 2019; Sutton & Barto, 2018). Yet, frequently, learning a model of the environment is very difficult, and in these cases, model-free algorithms tend to outperform (Janner et al., 2019).

Neural networks become useful in RL when the environment has a large observation-action space,[2] which happens frequently with realistic decision-making problems. Whenever there is a need for an agent to approximate some function, typically a function to represent the policy and/or to model the transition dynamics, neural networks can be used in this capacity due to their property of being general function approximators (Hornik et al., 1989). Although there are other function approximators that can be used in RL, for example Gaussian processes (Grande et al., 2014), neural networks have excelled in this role because of their ability to learn complex, nonlinear and high dimensional functions and their ability to adapt given new information (Arulkumaran et al., 2017). There is a multitude of deep RL algorithms since there are many design choices that can be made in constructing a deep RL agent—see Appendix A for more detail on these engineering decisions. In Table 1, we present some of the more common deep RL algorithms, which serve as good reference points for the current state of deep RL.

Training a deep RL agent involves allowing the agent to interact with the environment for potentially thousands to millions of time steps. During training, the deep RL agent continually updates its neural network parameters so that it will converge to an optimal policy. The amount of time needed for an agent to learn high reward yielding behaviour cannot be predetermined and depends on a host of factors including the complexity of the environment, the complexity of the agent, and more. Yet, overall, it has been well established that deep RL agents tend to be very sample inefficient (Gu et al., 2017), so it is recommended to provide a generous training budget for these agents.

The deep RL agent controls the learning process with parameters called *hyperparameters*. Examples of hyperparameters include the step size used for gradient ascent and the interval to interact with the environment before updating the policy. In contrast, a weight or bias in an agent's neural network is simply called a *parameter*. Parameters are learned by the agent, but the hyperparameters must be specified by the RL practitioner. Since the optimal hyperparameters vary across environments and cannot be predetermined (Henderson et al., 2019), it is necessary to find a good-performing set of hyperparameters in a process called hyperparameter tuning, which uses standard multi-dimensional optimization methods. We further discuss and show the benefits of hyperparameter tuning in Appendix B.

2.3 | RL objective

The reinforcement learning environment is typically formalized as a discrete-time partially observable Markov decision process (POMDP). A POMDP is a tuple that consists of the following:

- $\mathcal{S}$: a set of states called the state space
- $\mathcal{A}$: a set of actions called the action space
- Ω: a set of observations called the observation space
- $E(o_t \mid s_t)$: an emission distribution, which accounts for an agent's observation being different from the environment's state
- $T(s_{t+1} \mid s_t, a_t)$: a state transition operator which describes the dynamics of the system

TABLE 1 Survey of common deep RL algorithms

Abbreviation	Algorithm name	Model
PlaNet	Deep Planning Network (Hafner et al., 2019)	Model-based
I2A	Imagination-Augmented Agents (Weber et al., 2017)	Model-based
MBPO	Model-based Policy Optimization (Janner et al., 2019)	Model-based
DQN	Deep Q Networks (Mnih et al., 2015)	Model-free
A2C	Advantage Actor Critic (Mnih et al., 2016)	Model-free
A3C	Asynchronous A2C (Babaeizadeh et al., 2016)	Model-free
TRPO	Trust Region Policy Optimization (Schulman, Levine, et al., 2017)	Model-free
PPO	Proximal Policy Optimization (Schulman, Wolski, et al., 2017)	Model-free
DDPG	Deep Deterministic Policy Gradient (Lillicrap et al., 2019)	Model-free
TD3	Twin Delayed DDPG (Fujimoto et al., 2018)	Model-free
SAC	Soft Actor Critic (Haarnoja et al., 2018)	Model-free
IMPALA	Importance Weighted Actor Learner (Espeholt et al., 2018)	Model-free

- $r(s_t, a_t)$: a reward function
- $d_0(s_0)$: an initial state distribution
- $\gamma \in (0, 1]$: a discount factor that describes how much the agent will value rewards to be received in the distant future versus the immediate future (Clark, 2010)

The agent interacts with the environment in an iterative loop, whereby the agent only has access to the observation space, the action space and the discounted reward signal, $\gamma^t r(s_t, a_t)$. As the agent interacts with the environment by selecting actions according to its policy, $\pi(a_t | o_t)$,[3] the agent creates a trajectory, $\tau = (s_0, o_0, a_0, \ldots, s_{H-1}, o_{H-1}, a_{H-1}, s_H)$. From these definitions, we can provide an agent's trajectory distribution for a given policy as,

$$p_\pi(\tau) = d_0(s_0) \prod_{t=0}^{H-1} \pi(a_t | o_t) E(o_t | s_t) T(s_{t+1} | s_t, a_t).$$

The goal of reinforcement learning is for the agent to find an optimal policy distribution, $\pi^*(a_t | o_t)$, that maximizes the expected return, $J(\pi)$:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) \right] = \underset{\pi}{\operatorname{argmax}}\; J(\pi).$$

Although there are RL-based methods for infinite horizon problems, that is, when $H = \infty$, we will only present episodic or finite horizon POMDPs in this study. In Appendix A, we will discuss in greater detail how deep RL algorithms attempt to optimize the RL objective.

3 | RESULTS

We provide two examples that illustrate the application and potential of deep RL to ecological and conservation problems, highlighting both the potential and the inherent challenges. Annotated code for these examples may be found in Appendix B and at https://github.com/boettiger-lab/rl-intro. All algorithms were run on an NVIDIA Quadro RTX 8000 GPU. The training budget for the fishing scenario was 300 K timesteps (3 K runs, taking about 25 min). The training budget for the tipping point example was 3 M timesteps (6 K runs, taking around 3 hr). Software details and hyperparameters are provided in the associated GitHub repo. Hyperparameter tuning typically required 100 s of training runs using both Optuna, a python-based hyperparameter optimization module, and manual adjustments.

3.1 | Sustainable harvest sustainable harvest

The first example focuses on the important but well-studied problem of setting harvest quotas in fisheries management. This provides a natural benchmark for deep RL approaches, since we can compare the RL solution to the mathematical optimum directly. Determining fishing quotas is both a critical ecological issue (Costello et al., 2016; Worm et al., 2006, 2009) and a textbook example that has long informed the management of renewable resources within fisheries and beyond (Clark, 1990).

Given a population growth model that predicts the total biomass of a fish stock in the following year as a function of the current biomass, it is straightforward to determine what biomass corresponds to the maximum growth rate of the stock, or B_{MSY}, the biomass at maximum sustainable yield (MSY; Schaefer, 1954). When the population growth rate is stochastic, the problem is slightly harder to solve, as the harvest quota must constantly adjust to the ups and downs of stochastic growth, but it is still possible to show the optimal strategy merely seeks to maintain

the stock at B_{MSY}, adjusted for any discounting of future yields (Reed, 1979).

For illustrative purposes, we consider the simplest version of the dynamic optimal harvest problem as outlined by Clark (1973) (for the deterministic case) and Reed (1979) (under stochastic recruitment). The manager seeks to optimize the net present value (discounted cumulative catch) of a fishery, observing the stock size each year and setting an appropriate harvest quota in response. In the classical approach, the best model of the fish population dynamics must first be estimated from data, potentially with posterior distributions over parameter estimates reflecting any uncertainty. From this model, the optimal harvest policy—that is, the function which returns the optimal quota for each possible observed stock size—can be determined either by analytic (Reed, 1979) or numerical (Marescot et al., 2013) methods, depending on the complexity of the model. In contrast, a model-free deep RL algorithm makes no assumption about the precise functional form or parameter values underlying the dynamics—it is in principle agnostic to the details of the simulation.

We illustrate the deep RL approach using the model-free algorithm known as Twin Delayed Deep Deterministic Policy Gradient or more simply, TD3 (Fujimoto et al., 2018). A step-by-step walk-through for training agents on this environment is provided in the Appendix. We compare the resulting management, policy and reward under the RL agent to that achieved by the optimal management solution (Figure 2). Despite having no knowledge of the underlying model, the RL agent learns enough to achieve nearly optimal performance.

The cumulative reward (utility) realized across 100 stochastic replicates is indistinguishable from that of the optimal policy (Figure 2). Nevertheless, comparing the mean state over replicate simulations reveals some differences in the RL strategy, wherein the stock is maintained at a slightly higher-than-optimal biomass. Because our state space and action space are sufficiently low-dimensional in this example, we are also able to visualize the policy function directly, and compare to the optimal policy (Figure 2). This confirms that quotas tend to be slightly lower than optimal, most notably at larger stock sizes. These features highlight a common challenge in the design and training of RL algorithms. RL cares only about improving the realized cumulative reward, and may sometimes achieve this in unexpected ways. Because these simulations rarely reach stock sizes at or above carrying capacity, that is, larger stock sizes are under-explored, these larger stock sizes show a greater deviation from the optimal policy than we observe at more frequently visited lower stock sizes. This observation brings up a point that is well worth discussing, which is how to best identify and resolve underexplored scenarios. Usually, RL practitioners identify underexplored scenarios by either doing extensive testing or visualizing the policy, then tweaking the hyperparameters relevant to exploration in hopes of improving the result.

How could an RL agent be applied to empirical data? One solution would be to train an agent on a simulation environment that approximates the fishery of interest then query the policy of the agent to find a quota for the observed stock. To illustrate this, we examine the quota that would be recommended by our newly trained RL agent, above, against historical harvest levels of Argentine hake based on stock assessments from 1986–2014 (RAM Legacy Stock Assessment Database, 2020, see Appendix D). Hake stocks showed a marked decline throughout this period, while harvests decreased only in proportion (Figure 3). In contrast, our RL agent would have recommended significantly lower quotas over most of the same interval, including the closure of the fishery as stocks were sufficiently depleted—a stark contrast to the management policy evidenced in the historical catch. Note that we have no way of knowing for sure if the RL quotas would have led to recovery nor do we know the optimal harvest rates, because we can never know the "true model" of the Argentine

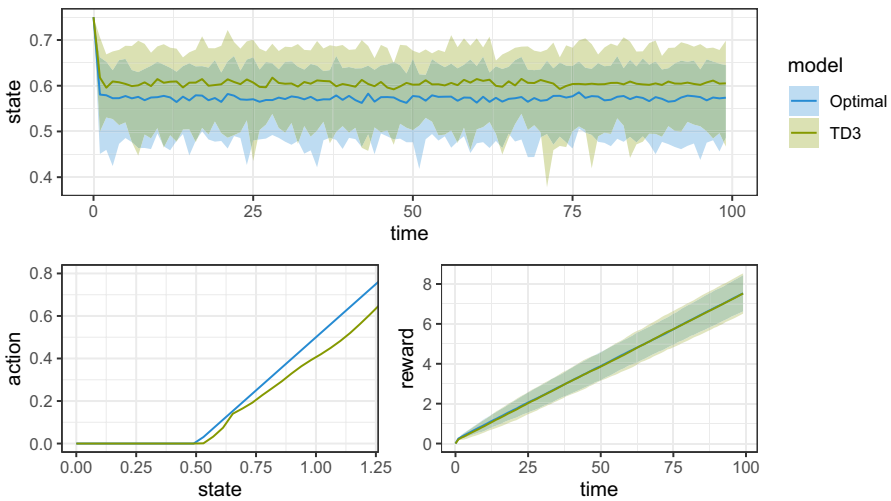

FIGURE 2 Fisheries management using neural network agents trained with RL algorithm TD3 compared to optimal management. Top panel: Mean fish population size over time across 100 replicates. Shaded region shows the 95% confidence interval over simulations. Lower left: The optimal solution is the policy of constant escapement. Below the target escapement of 0.5, no harvest occurs, while any stock above that level is immediately harvested back down. The TD3 agent adopts a policy that ceases any harvest below this level, while allowing a somewhat higher escapement than optimal. Lower right: TD3 achieves a nearly-optimal mean utility.

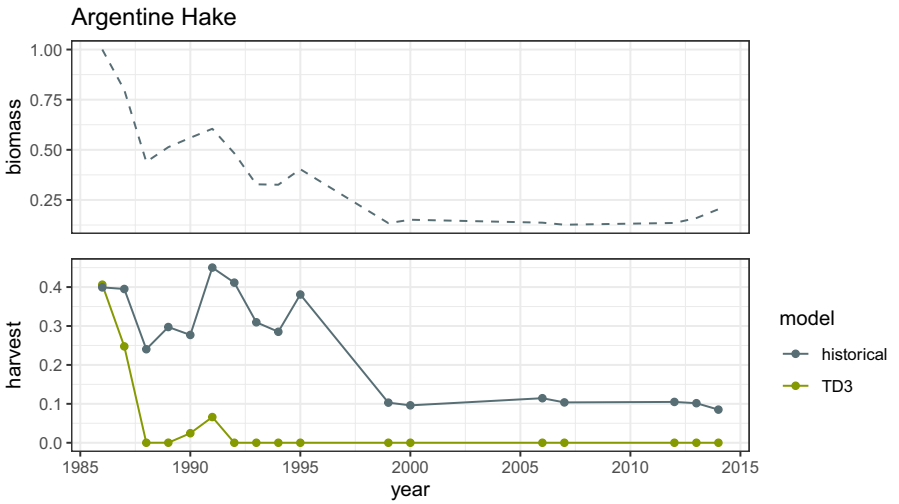

FIGURE 3 Setting fisheries harvest quotas using deep RL. Argentine hake fish stocks show a marked decline between 1986 and 2014 (upper panel). Historical harvests (lower panel) declined only slowly in response to consistently falling stocks, suggesting overfishing. In contrast, RL-based quotas would have been set considerably lower than observed harvests in each year of the data. As decline persists, the RL-based management would have closed the fishery to future harvest until the stock recovered.

hake dynamics. We can confirm that the fishery closures seen in the RL agent's solution are considered optimal under the assumptions of constant escapement theory (Reed, 1979) whenever the stock is below the biomass of maximum sustainable yield (B_{MSY}), and that most fisheries models of this stock (RAM Legacy Stock Assessment Database, 2020) would suggest that the populations observed in the latter two decades of the data are below that threshold.

This approach is not as different from conventional strategies as it may seem. In a conventional approach, ecological models are first estimated from empirical data, (stock assessments in the fisheries case). Quotas can then be set based directly on these model estimates, or by comparing alternative candidate "harvest control rules" (policies) against model-based simulations of stock dynamics. This latter approach, known in fisheries as management strategy evaluation (MSE; Punt et al., 2016) is already closely analogous to the RL process. Instead of researchers evaluating a handful of control rules, the RL agent proposes and evaluates a plethora of possible control rules autonomously.

3.2 | Ecological tipping points

Our second example focuses on a case for which we do not have an existing, provably optimal policy to compare against. We consider the generic problem of an ecosystem facing slowly deteriorating environmental conditions, which move the dynamics closer towards a tipping point (Figure 4). This model of a critical transition has been posited widely in ecological systems, from the simple consumer-resource model of May, 1977 on which our dynamics are based, to microbial dynamics (Dai et al., 2012), lake ecosystem communities (Carpenter et al., 2011) and planetary ecosystems (Barnosky et al., 2012). On top of these ecological dynamics, we introduce an explicit ecosystem service model quantifying the value of a more desirable 'high' state relative to the 'low' state. For simplicity, we assume a proportional benefit b associated with the ecosystem state $X(t)$. Thus, when the ecosystem is near the 'high' equilibrium, $\hat{X}_H$, the corresponding ecosystem benefit, $b\hat{X}_H$, is higher than at the low equilibrium, bx_L, consistent with the intuitive description of ecosystem tipping points (Barnosky et al., 2012).

We also enumerate the possible actions that a manager may take in response to environmental degradation. In the absence of any management response, we assume the environment deteriorates at a fixed rate α, which can be thought of as the incremental increase in global mean temperature or similar anthropogenic forcing term. Management can slow or even reverse this trend by choosing an opposing action A_t. We assume that large actions are proportionally more costly than small actions, consistent with the expectation of diminishing returns: taking the cost associated with an action A_t as equal to cA_t^2. Many alterations of these basic assumptions are also possible: our `gym_conservation` implements a range of different scenarios with user-configurable settings. In each case, the manager observes the current state of the system each year and must then select the policy response that year.

Because this problem involves a parameter whose value changes over time (the slowly deteriorating environment), the resulting ecosystem dynamics are not autonomous. This precludes our ability to solve for the optimal management policy using classical theory such as for Markov decision processes (MDP, Marescot et al., 2013), typically used to solve sequential decision-making problems. However, it is often argued that simple rules can achieve nearly optimal management of ecological conservation objectives in many cases (Joseph et al., 2009; Meir et al., 2004; Wilson et al., 2006). A common conservation strategy employs a fixed response level rather than a dynamic policy which is toggled up or down each year: for example, declaring certain regions as protected areas in perpetuity. An intuitive strategy faced with an ecosystem tipping point would be 'perfect conservation', in which the management response is perfectly calibrated to counter-balance any further decline. While the precise rate of such decline may not be known in practice (and will not be known to RL algorithms before-hand either), it is easy to implement such a policy in simulation for comparative purposes. We compare this rule-of-thumb to a policy found by training an agent using the TD3 algorithm.

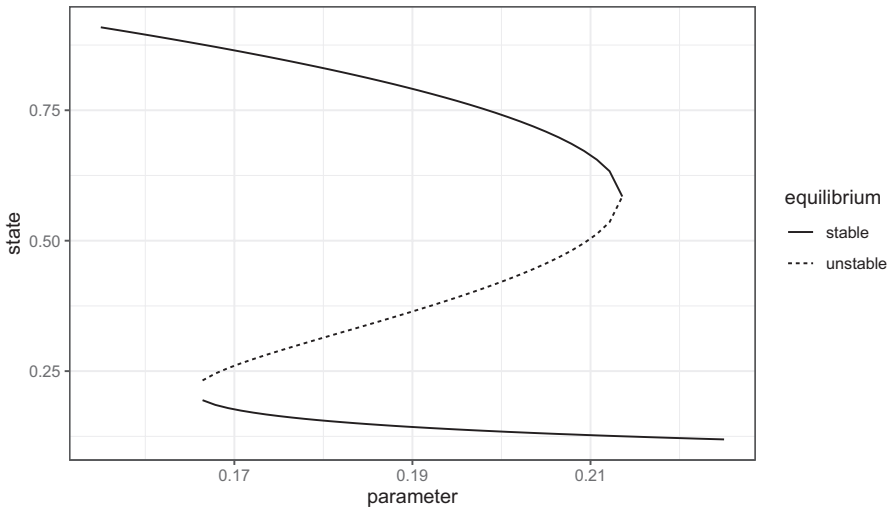

FIGURE 4 Bifurcation diagram for tipping point scenario. The ecosystem begins in the desirable 'high' state under an environmental parameter (e.g. global mean temperature, arbitrary units) of 0.19. In the absence of conservation action, the environment worsens (e.g. rising mean temperature) as the parameter increases. This results in only a slow degradation of the stable state, until the parameter crosses the tipping point threshold at about 0.215, where the upper stable branch is annihilated in a fold bifurcation and the system rapidly transitions to lower stable branch, around state of 0.1. Recovery to the upper branch requires a much greater conservation investment, reducing the parameter all the way to 0.165 where the reverse bifurcation will carry it back to the upper stable branch

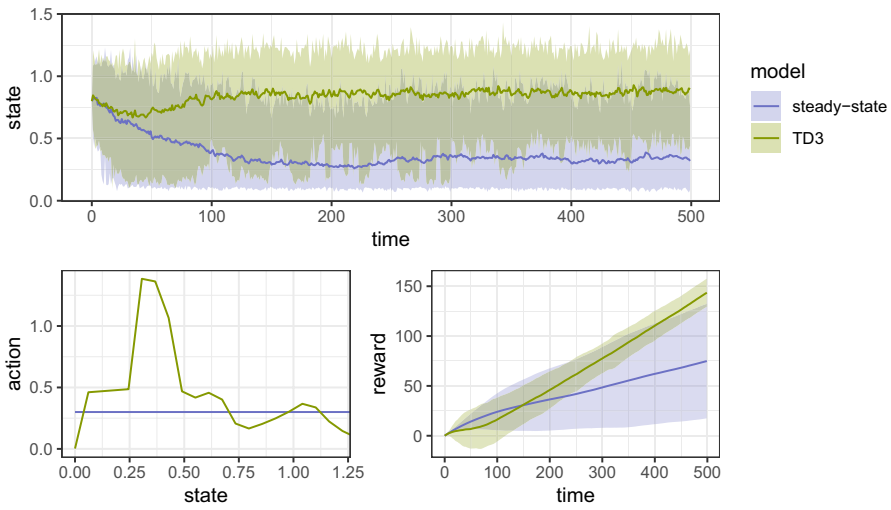

FIGURE 5 Ecosystem dynamics under management using the steady-state rule-of-thumb strategy compared to management using a neural network trained using the TD3 RL algorithm. Top panel: Mean and 95% confidence interval of ecosystem state over 100 replicate simulations. As more replicates cross the tipping point threshold under steady-state strategy, the mean slowly decreases, while the TD3 agent preserves most replicates safely above the tipping point. Lower left: The policy function learned using TD3 relative to the policy function under the steady state. Lower right: Mean rewards under TD3 management eventually exceed those expected under the steady-state strategy as a large initial investment in conservation eventually pays off.

The TD3-trained agent proves far more successful in preventing chance transitions across the tipping point, consistently achieving a higher cumulative ecosystem service value across replicates than the steady-state strategy.

Examining the replicate management trajectories and corresponding rewards (Figure 5) reveal that the RL agent incurs significantly higher costs in the initial phases of the simulation, dipping well below the mean steady-state reward initially before exceeding it in the long run. This initial investment then begins to pay off—by about the 200th time step, the RL agent has surpassed the performance of the steady-state strategy. The policy plot provides more intuition for the RL agent's strategy: at very high state values, the RL agent opts for no conservation action—so far from the tipping point, no response is required. Near the tipping point, the RL agent steeply ramps up the conservation effort, and retains this effort even as the system falls below the critical threshold, where a sufficiently

aggressive response can tip the system back into recovery. For a system at or very close to the zero-state, the RL agent gives up, opting for no action. Recall that the quadratic scaling of cost makes the rapid response of the TD3 agent much more costly to achieve the same net environmental improvement divided into smaller increments over a longer timeline. However, our RL agent has discovered that the extra investment for a rapid response is well justified as the risk of crossing a tipping point increases.

A close examination of the trajectories of individual simulations which cross the tipping point under either management strategy (see Appendix B) further highlights the difference between these two approaches. Under the steady-state strategy, the system remains poised too close to the tipping point: stochastic noise eventually drives most replicates across the threshold, where the steady-state strategy is too weak to bring them back once they collapse. As replicate after replicate stochastically crashes, the mean state and mean reward bend increasingly downwards. In contrast, the RL agent edges the system slightly farther away from the tipping point, decreasing but not eliminating the odds of a chance transition. In the few replicates that experience a critical transition anyway, the RL agent usually responds with sufficient commitment to ensure their recovery (Appendix B). Only 3 out of 100 replicates degrade far enough for the RL agent to give up the high cost of trying to rescue them. The RL agent's use of a more dynamic strategy outperforms the steady-state strategy. Numerous kinks visible in the RL policy function also suggest that this solution is not yet optimal. Such quirks are likely to be common features of RL solutions—long as they have minimal impact on realized rewards. Further tuning of hyper-parameters, increased training, alterations or alternatives to the training algorithm would likely be able to further improve upon this performance.

3.3 | Additional environments

Ecology holds many open problems for deep RL. To extend the examples presented here to reflect greater biological complexity or more realistic decision scenarios, the transition, emission and/or reward functions of the environment can be modified. We provide an initial library of example environments at https://boettiger-lab.github.io/conservation-gym. Some environments in this library include a wildfire `gym` that poses the problem of wildfire suppression with a cellular automata model, an epidemic `gym` that examines timing of interventions to curb disease spread, as well as more complex variations of the fishing and conservation environments presented above.

4 | DISCUSSION

Ecological challenges facing the planet today are complex, and their outcomes are both uncertain and consequential. Even our best models and best research will never provide a crystal ball to the future, only better elucidate possible scenarios. Consequently, that research must also confront the challenge of making robust, resilient decisions in a changing world. The science of ecological management and quantitative decision-making has a long history (e.g. Schaefer, 1954; Walters & Hilborn, 1978) and remains an active area of research (Fischer et al., 2009; Polasky et al., 2011; Wilson et al., 2006). However, the limitations of classical methods such as optimal control frequently constrain applications to relatively simplified models (Wilson et al., 2006), ignoring elements such as spatial or temporal heterogeneity and autocorrelation, stochasticity, imperfect observations, age or state structure, and other sources of complexity that are both pervasive and influential on ecological dynamics (Hastings & Gross, 2012). Complexity comes not only from the ecological processes but also the available actions. Deep RL agents have proven remarkably effective in handling such complexity, particularly when leveraging immense computing resources increasingly available through advances in hardware and software (Matthews, 2018).

This paper does not set the precedent as the first application of RL to ecology. There have been a number of studies applying RL to behavioural ecology, typically with multiagent environments (Frankenhuis et al., 2019; Perolat et al., 2017; Wang et al., 2020). Yet, it is important to distinguish the aim of these behavioural studies from the aim of applying RL to conservation management. In previous behavioural ecology studies, RL algorithms as a substitute for animal learning mechanisms (Perolat et al., 2017; Wang et al., 2020). When applying deep RL to conservation management, we do not make the assumption that an RL algorithm learns analogously to how an animal learns. We instead propose that RL be used as a tool to search for solutions to decision-making problems.

The examples presented here only scrape the surface of possible RL applications to conservation problems. The examples we have focused on are intentionally quite simple, though it is worth remembering that these very same simple models have a long history of relevance and application in both research and policy contexts. Despite their simplicity, the optimal strategy is not always obvious beforehand, however intuitive it may appear in retrospect. In the case of the ecosystem tipping point scenario, the optimal strategy is unknown, and the approximate solution found by our RL implementation could almost certainly be improved upon. In these simple examples in which the simulation implements a single model, training is analogous to classical methods that take the model as given (Marescot et al., 2013). But classical approaches can be difficult to generalize when the underlying model is unknown. In contrast, the process of training an RL algorithm on a more complex problem is no different than training on a simple one: we only need access to a simulation which can generate plausible future states in response to possible actions. This flexibility of RL could allow us to attain better decision-making insight for our most realistic ecological models like those used for the management of forests and wildfire (Moritz et al., 2014; Pacala et al., 1996), disease (Dobson et al., 2020), marine ecosystems (Steenbeek et al., 2016), or global climate change (Nordhaus, 1992).

The rapidly expanding class of model-free RL algorithms is particularly appealing given the ubiquitous presence of model uncertainty in ecological dynamics. Rarely do we know the underlying functional forms for ecological processes. Methods which must first assume something about the structure or functional form of a process before estimating the corresponding parameter can only ever be as good as those structural assumptions. Frequently, available ecological data are insufficient to distinguish between possible alternative models (Knape & de Valpine, 2012), or the correct model may be nonidentifiable with any amount of data. Model-free RL approaches offer a powerful solution for this thorny issue. Model-free algorithms have proven successful at learning effective policies even when the underlying model is difficult or impossible to learn (Pong et al., 2020), as long as simulations of possible mechanisms are available.

Successfully applying RL to complex ecological problems is no easy task. Even on relatively uncomplicated environments, training an RL agent can be more challenging than expected due to an entanglement of reasons, see Table 2, like hyperparameter instability and poor exploration that can be very difficult to resolve (Berger-Tal et al., 2014; Henderson et al., 2019). It is also worth acknowledging that deep RL algorithms, particularly model-free algorithms, have poor sample efficiency, which could limit deep RL from being effective on environments that are slow to run (Haarnoja et al., 2018). Thus, as Sections 5.1 and 5.2 illustrate, it is important to begin with simple problems, including those for which an optimal strategy is already known. Such examples provide important benchmarks to calibrate the performance, tuning and training requirements of RL. Once RL agents have mastered the basics, the examples can be easily extended into more complex problems by changing the environment. Yet, even in the case that an agent performs well on a realistic problem, there will be a range of open questions in using deep RL to inform decision-making. Since deep neural networks lack transparency (Castelvecchi, 2016), can we be confident that the agent will generalize its past experience to new situations—especially when we cannot readily visualize the policy? To gain such confidence, it will be necessary to do extensive testing on previously unseen contexts (Kazak et al., 2019), but even then, it can be difficult to verify that the agent will perform as expected. Given that there have been many examples of reward misspecification leading to undesirable behaviour (Hadfield-Menell et al., 2020), what if we have selected an objective that unexpectedly causes damaging behaviour? Reward misspecification is not unique to RL and has long been a central problem in ecological management and decision-making (Conroy & Peterson, 2013; Gregory et al., 2012), but it is important to make clear that RL does not resolve this issue. A greater role of algorithms in conservation decision-making also raises questions about ethics and power, particularly when those algorithms are opaque or proprietary (Chapman et al., 2021; Scoville et al., 2021).

Yet, a more immediate barrier to the use of deep RL in conservation is deep RL's hardware requirements. Depending on the complexity of the RL environment and agent, the equipment necessary to train an agent can vary widely. The examples shown above were selected so they can be replicated on a personal computer, but more realistic problems will likely require specialized computational resources. For instance, one of the most notable achievements in RL, Alphastar, required 33 TPUs, processors that are specialized for deep learning, for more than 40 days (Vinyals et al., 2019). Fully detailed conservation decision-making problems will likely require comparable specialized algorithms and hardware that ecologists do not generally have access to. For deep RL to be an effective tool for conservation, there will need to be large investments of time and money, and extensive collaboration across computer science and ecology.

Deep RL is still a very young field, where despite several landmark successes, potential far outstrips practice. Recent advances in the field have proven the potential of the approach to solve complex problems (Mnih et al., 2015; Silver et al., 2016, 2017, 2018), but typically leveraging large teams with decades of experience in ML and millions of dollars worth of computing power (Silver et al., 2017). Successes have so far been concentrated in applications to games and robotics, not scientific and policy domains, though this is beginning to change (Popova et al., 2018; Zhou et al., 2017). Iterative improvements to well-posed public challenges have proven immensely effective in the computer science community in tackling difficult

TABLE 2 Practical issues with deep RL

Issue	Description
Generalization	Agents struggle to adapt to tasks not seen in training (Kirk et al., 2022).
Reproducibility	It can be very challenging to replicate results due to a host of reasons like differences in computational hardware (Henderson et al., 2019)
Lack of transparency	Deep RL users cannot interpret why agents select actions (Castelvecchi, 2016)
Hyperparameter instability	Agent performance can vary significantly over slight alterations in hyperparameters, like initialization seed (Henderson et al., 2019)
Reward misspecification	Agents commonly learn undesirable behaviour that still maximizes the RL objective (Hadfield-Menell et al., 2020)
High capital demands	Landmark successes like AlphaGo and AlphaStar have required very large teams of researchers and large amounts of computational power (Silver et al., 2017; Vinyals et al., 2019)
Sample inefficiency	Current algorithms require large amounts of interaction with the environment to achieve reward maximization (Haarnoja et al., 2018)

problems, which allow many teams with diverse expertise not only to compete but to learn from each other (Deng et al., 2009; Villarroel et al., 2013). By working to develop similarly well-posed challenges as clear benchmarks, ecology and environmental science researchers may be able to replicate that collaborative, iterative success in cracking hard problems.

AUTHOR CONTRIBUTIONS

Marcus Lapeyrolerie and Carl Boettiger developed the code and wrote the manuscript with support from Kari E. A. Norman and Melissa S. Chapman.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. DBI-1942280. We thank Benjamin Blonder, Zachary Sunberg and Claire Tomlin for their thoughtful comments.

CONFLICT OF INTEREST

The authors declare that there are no conflicts of interest.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1111/2041-210X.13954.

DATA AVAILABILITY STATEMENT

We have created a GitHub repository https://github.com/boettiger-lab/rl-intro that contains the code used to produce the figures herein. This repository has been archived on Zenodo at https://doi.org/10.5281/zenodo.6886892 (Lapeyrolerie et al., 2022).

ORCID

Marcus Lapeyrolerie https://orcid.org/0000-0003-2588-7843
Carl Boettiger https://orcid.org/0000-0002-1642-628X

ENDNOTES

[1] The terms observation and state are used nearly interchangeably in describing RL, so it is worth clarifying the distinction. An observation is the depiction of the environment that is given to the agent at each time step, but the state is the true underlying description of the environment. When the term observation is used, this usually means that the observation does not provide an accurate portrayal of the environment's state. Yet, in cases when the observation and state are in agreement, the term observation is typically not used at all.

[2] Conventionally, an observation-action space is considered to be large when it is non-tabular, that is, it cannot be represented in a computationally tractable table.

[3] The policy can also be conditioned on a history of observations, $(o_0, ..., o_t)$.

REFERENCES

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). A brief survey of deep reinforcement learning. *IEEE Signal Processing Magazine*, *34*(6), 26–38. https://doi.org/10.1109/MSP.2017.2743240

Babaeizadeh, M., Frosio, I., Tyree, S., Clemons, J., & Kautz, J. (2016). Reinforcement learning through asynchronous advantage actor-critic on a GPU (version 3). *arXiv*. https://doi.org/10.48550/ARXIV.1611.06256

Barnosky, A. D., Hadly, E. A., Bascompte, J., Berlow, E. L., Brown, J. H., Fortelius, M., Getz, W. M., Harte, J., Hastings, A., Marquet, P. A., Martinez, N. D., Mooers, A., Roopnarine, P., Vermeij, G., Williams, J. W., Gillespie, R., Kitzes, J., Marshall, C., Matzke, N., … Smith, A. B. (2012). Approaching a state shift in Earth's biosphere. *Nature*, *486*(7401), 52–58. https://doi.org/10.1038/nature11018

Berger-Tal, O., Nathan, J., Meron, E., & Saltz, D. (2014). The exploration-exploitation dilemma: A multidisciplinary framework. *PLoS ONE*, *9*(4), e95693. https://doi.org/10.1371/journal.pone.0095693

Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. *arXiv:1606.01540 [Cs]*, June. http://arxiv.org/abs/1606.01540

Carpenter, S. R., Cole, J. J., Pace, M. L., Batt, R., Brock, W. A., Cline, T., Coloso, J., Hodgson, J. R., Kitchell, J. F., Seekell, D. A., Smith, L., & Weidel, B. (2011). Early warnings of regime shifts: A whole-ecosystem experiment. *Science*, *332*(6033), 1079–1082. https://doi.org/10.1126/science.1203672

Castelvecchi, D. (2016). Can we open the black box of AI? *Nature News*, *538*(7623), 20–23. https://doi.org/10.1038/538020a

Chades, I., Pascal, L. V., Nicol, S., Fletcher, C. S., & Mestres, J. F. (2021). A primer on partially observable Markov decision processes (POMDPs). *Methods in Ecology and Evolution*, *12*, 2058–2072. https://doi.org/10.1111/2041-210X.13692

Chapman, M. S., Oestreich, W. K., Frawley, T. H., Boettiger, C., Diver, S., Santos, B. S., Scoville, C., Armstrong, K., Blondin, H., Chand, K., Haulsee, D. E., Knight, C. J., & Crowder, L. B. (2021). Promoting equity in the use of algorithms for high-seas conservation. *One Earth*, *4*(6), 790–794. https://doi.org/10.1016/j.oneear.2021.05.011

Clark, C. W. (1973). Profit maximization and the extinction of animal species. *Journal of Political Economy*, *81*(4), 950–961. https://doi.org/10.1086/260090

Clark, C. W. (1990). *Mathematical bioeconomics: The optimal management of renewable resources* (2nd ed.). Wiley-Interscience.

Clark, C. W. (2010). *Mathematical bioeconomics: The mathematics of conservation* (3rd ed.). Wiley Pure and applied mathematics.

Conroy, M. J., & Peterson, J. T. (2013). *Decision making in natural resource management: A structured, adaptive approach: A structured, adaptive approach* (1st ed.). Wiley. https://doi.org/10.1002/9781118506196

Costello, C., Ovando, D., Clavelle, T., Strauss, C. K., Hilborn, R., Melnychuk, M. C., Branch, T. A., Gaines, S. D., Szuwalski, C. S., Cabral, R. B., Rader, D. N., & Leland, A. (2016). Global fishery prospects under contrasting management regimes. *Proceedings of the National Academy of Sciences of the United States of America*, *113*(18), 5125–5129. https://doi.org/10.1073/pnas.1520420113

Dai, L., Vorselen, D., Korolev, K. S., & Gore, J. (2012). Generic indicators for loss of resilience before a tipping point leading to population collapse. *Science (New York, N.Y.)*, *336*(6085), 1175–1177. https://doi.org/10.1126/science.1219805

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). IEEE. https://doi.org/10.1109/CVPR.2009.5206848

Dirzo, R., Young, H. S., Galetti, M., Ceballos, G., Isaac, N. J. B., & Collen, B. (2014). Defaunation in the Anthropocene. *Science*, *345*(6195), 401–406.

Dobson, A. P., Pimm, S. L., Hannah, L., Kaufman, L., Ahumada, J. A., Ando, A. W., Bernstein, A., Busch, J., Daszak, P., Engelmann, J., Kinnaird, M. F., Li, B. V., Loch-Temzelides, T., Lovejoy, T., Nowak, K., Roehrdanz, P. R., & Vale, M. M. (2020). Ecology and economics

for pandemic prevention. *Science*, *369*(6502), 379–381. https://doi.org/10.1126/science.abc3189

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., Legg, S., & Kavukcuoglu, K. (2018). IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures (version 3). *arXiv*. https://doi.org/10.48550/ARXIV.1802.01561

Ferrer-Mestres, J., Dietterich, T. G., Buffet, O., & Chades, I. (2021). K-N-MOMDPs: Towards interpretable solutions for adaptive management. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(17), 14775–14784. https://ojs.aaai.org/index.php/AAAI/article/view/17735

Fischer, J., Peterson, G. D., Gardner, T. A., Gordon, L. J., Fazey, I., Elmqvist, T., Felton, A., Folke, C., & Dovers, S. (2009). Integrating resilience thinking and optimisation for conservation. *Trends in Ecology & Evolution*, *24*(10), 549–554. https://doi.org/10.1016/j.tree.2009.03.020

Fonnesbeck, C. J. (2008). Solving dynamic wildlife resource optimization problems using reinforcement learning. *Natural Resource Modeling*, *18*(1), 1–40. https://doi.org/10.1111/j.1939-7445.2005.tb00147.x

Frankenhuis, W. E., Panchanathan, K., & Barto, A. G. (2019). Enriching behavioral ecology with reinforcement learning methods. *Behavioural Processes*, *161*(April), 94–100. https://doi.org/10.1016/j.beproc.2018.01.008

Fujimoto, S., van Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. *arXiv:1802.09477 [Cs, Stat]*. http://arxiv.org/abs/1802.09477

Getz, W. M., Marshall, C. R., Carlson, C. J., Giuggioli, L., Ryan, S. J., Romañach, S. S., Boettiger, C., Chamberlain, S. D., Larsen, L., D'Odorico, P., & O'Sullivan, D. (2018). Making ecological models adequate. *Ecology Letters*, *21*(2), 153–166. https://doi.org/10.1111/ele.12893

Grande, R., Walsh, T., & How, J. (2014). Sample efficient reinforcement learning with gaussian processes. In E. P. Xing & T. Jebara (Eds.), *Proceedings of the 31st international conference on machine learning* (Vol. *32*, pp. 1332–1340). PMLR: Proceedings of Machine Learning Research 2. http://proceedings.mlr.press/v32/grande14.html

Gregory, R., Failing, L., Harstone, M., Long, G., McDaniels, T., & Ohlson, D. (2012). *Structured decision making: A practical guide to environmental management choices* (1st ed.). Wiley. https://doi.org/10.1002/9781444398557

Gu, S., Lillicrap, T., Ghahramani, Z., Turner, R. E., & Levine, S. (2017). Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv:1611.02247 [Cs]*. http://arxiv.org/abs/1611.02247

Ha, S., Xu, P., Tan, Z., Levine, S., & Tan, J. (2020). Learning to walk in the real world with minimal human effort. *arXiv:2002.08550 [Cs]*. http://arxiv.org/abs/2002.08550

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv:1801.01290 [Cs, Stat]*. http://arxiv.org/abs/1801.01290

Hadfield-Menell, D., Milli, S., Abbeel, P., Russell, S., & Dragan, A. (2020). Inverse Reward Design. *arXiv:1711.02827 [Cs]*. http://arxiv.org/abs/1711.02827

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., & Davidson, J. (2019). Learning latent dynamics for planning from pixels. *arXiv:1811.04551 [Cs, Stat]*. http://arxiv.org/abs/1811.04551

Hastings, A., & Gross, L. J. (Eds.). (2012). *Encyclopedia of theoretical ecology*. University of California Press.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2019). Deep reinforcement learning that matters. *arXiv:1709.06560 [Cs, Stat]*. http://arxiv.org/abs/1709.06560

Hernandez, D., & Brown, T. B. (2020). Measuring the algorithmic efficiency of neural networks. *arXiv:2005.04305 [Cs, Stat]*. http://arxiv.org/abs/2005.04305

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Huang, D. (2018). How Much Did AlphaGo Zero Cost? https://www.yuzeh.com/data/agz-cost.html

Janner, M., Justin, F., Zhang, M., & Levine, S. (2019). When to trust your model: Model-based policy optimization. *arXiv:1906.08253 [Cs, Stat]*. http://arxiv.org/abs/1906.08253

Joseph, L. N., Maloney, R. F., & Possingham, H. P. (2009). Optimal allocation of resources among threatened species: A project prioritization protocol. *Conservation Biology*, *23*(2), 328–338. https://doi.org/10.1111/j.1523-1739.2008.01124.x

Joseph, M. B. (2020). Neural hierarchical models of ecological populations. *Ecology Letters*, *23*(4), 734–747. https://doi.org/10.1111/ele.13462

Kazak, Y., Barrett, C., Katz, G., & Schapira, M. (2019). Verifying Deep-RL-Driven systems. In *Proceedings of the 2019 workshop on network meets AI & ML – NetAI'19* (pp. 83–89). ACM Press. https://doi.org/10.1145/3341216.3342218

Kirk, R., Zhang, A., Grefenstette, E., & Rocktäschel, T. (2022). A survey of generalisation in deep reinforcement learning. *arXiv:2111.09794 [Cs]*. http://arxiv.org/abs/2111.09794

Knape, J., & de Valpine, P. (2012). Are patterns of density dependence in the global population dynamics database driven by uncertainty about population abundance? *Ecology Letters*, *15*(1), 17–23. https://doi.org/10.1111/j.1461-0248.2011.01702.x

Lapeyrolerie, M., Boettiger, C., Norman, K., & Chapman, M. (2022). boettiger-lab/rl-intro: First look submission (version v1.1) [Computer software]. *Zenodo*. https://doi.org/10.5281/ZENODO.6886892

Levins, R. (1966). The strategy of model building in population biology. *American Scientist*, *54*(4), 421–431.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2019). Continuous control with deep reinforcement learning. *arXiv:1509.02971 [Cs, Stat]*. http://arxiv.org/abs/1509.02971

Marescot, L., Chapron, G., Chadès, I., Fackler, P. L., Duchamp, C., Marboutin, E., & Gimenez, O. (2013). Complex decisions made simple: A primer on stochastic dynamic programming. *Methods in Ecology and Evolution*, *4*(9), 872–884. https://doi.org/10.1111/2041-210X.12082

Matthews, D. (2018). Supercharge your data wrangling with a graphics card. *Nature*, *562*(7725), 151–152. https://doi.org/10.1038/d41586-018-06870-8

May, R. M. (1977). Thresholds and breakpoints in ecosystems with a multiplicity of stable states. *Nature*, *269*(5628), 471–477. https://doi.org/10.1038/269471a0

Meir, E., Andelman, S., & Possingham, H. P. (2004). Does conservation planning matter in a dynamic and uncertain world? *Ecology Letters*, *7*(8), 615–622. https://doi.org/10.1111/j.1461-0248.2004.00624.x

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. *arXiv:1602.01783 [Cs]*. http://arxiv.org/abs/1602.01783

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Moritz, M. A., Batllori, E., Bradstock, R. A., Gill, A. M., Handmer, J., Hessburg, P. F., Leonard, J., McCaffrey, S., Odion, D. C., Schoennagel, T., & Syphard, A. D. (2014). Learning to coexist with wildfire. *Nature*, *515*(7525), 58–66. https://doi.org/10.1038/nature13946

Nordhaus, W. D. (1992). An optimal transition path for controlling greenhouse gases. *Science*, *258*(5086), 1315–1319. https://doi.org/10.1126/science.258.5086.1315

OpenAI, Andrychowicz, M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., Schneider, J., Sidor, S., Tobin, J., Welinder, P., Weng, L., & Zaremba, W. (2018). Learning dexterous in-hand manipulation (version 5). *arXiv*. https://doi.org/10.48550/ARXIV.1808.00177

Pacala, S. W., Canham, C. D., Saponara, J., Silander, J. A., Kobe, R. K., & Ribbens, E. (1996). Forest models defined by field measurements: Estimation, error analysis and dynamics. *Ecological Monographs*, *66*(1), 1–43. https://doi.org/10.2307/2963479

Perolat, J., Leibo, J. Z., Zambaldi, V., Beattie, C., Tuyls, K., & Graepel, T. (2017). A multi-agent reinforcement learning model of common-Pool resource appropriation. *arXiv:1707.06600 [Cs, q-Bio]*. http://arxiv.org/abs/1707.06600

Polasky, S., Carpenter, S. R., Folke, C., & Keeler, B. (2011). Decision-making under great uncertainty: Environmental management in an era of global change. *Trends in Ecology & Evolution*, *26*(8), 398–404. https://doi.org/10.1016/j.tree.2011.04.007

Pong, V., Shixiang, G., Dalal, M., & Levine, S. (2020). Temporal difference models: Model-free deep RL for model-based control. *arXiv:1802.09081 [Cs]*. http://arxiv.org/abs/1802.09081

Popova, M., Isayev, O., & Tropsha, A. (2018). Deep reinforcement learning for de novo drug design. *Science Advances*, *4*(7), eaap7885. https://doi.org/10.1126/sciadv.aap7885

Punt, A. E., Butterworth, D. S., de Moor, C. L., De Oliveira, J. A. A., & Haddon, M. (2016). Management strategy evaluation: Best practices. *Fish and Fisheries*, *17*(2), 303–334. https://doi.org/10.1111/faf.12104

RAM Legacy Stock Assessment Database. (2020). RAM legacy stock assessment database V4.491. https://doi.org/10.5281/zenodo.3676088

Reed, W. J. (1979). Optimal escapement levels in stochastic and deterministic harvesting models. *Journal of Environmental Economics and Management*, *6*(4), 350–363. https://doi.org/10.1016/0095-0696(79)90014-7

Schaefer, M. B. (1954). Some aspects of the dynamics of populations important to the management of the commercial marine fisheries. *Bulletin of the Inter-American Tropical Tuna Commission*, *1*(2), 27–56. https://doi.org/10.1007/BF02464432

Scheffer, M., Carpenter, S. R., Dakos, V., & van Nes, E. (2015). Generic indicators of ecological resilience. *Annual Review of Ecology, Evolution, and Systematics*, *46*(1), 145–167. https://doi.org/10.1146/annurev-ecolsys-112414-054242

Schulman, J., Levine, S., Moritz, P., Jordan, M. I., & Abbeel, P. (2017). Trust region policy optimization. *arXiv:1502.05477 [Cs]*. http://arxiv.org/abs/1502.05477

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms (Version 2). *arXiv*. https://doi.org/10.48550/ARXIV.1707.06347

Scoville, C., Chapman, M., Amironesei, R., & Boettiger, C. (2021). Algorithmic conservation in a changing climate. *Current Opinion in Environmental Sustainability*, *51*(August), 30–35. https://doi.org/10.1016/j.cosust.2021.01.009

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489. https://doi.org/10.1038/nature16961

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., & Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, *362*(6419), 1140–1144. https://doi.org/10.1126/science.aar6404

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., & Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, *550*(7676), 354–359. https://doi.org/10.1038/nature24270

Silvestro, D., Goria, S., Sterner, T., & Antonelli, A. (2022). Improving biodiversity protection through artificial intelligence. *Nature Sustainability*, *5*(5), 415–424. https://doi.org/10.1038/s41893-022-00851-6

Steenbeek, J., Buszowski, J., Christensen, V., Akoglu, E., Aydin, K., Ellis, N., Felinto, D., Guitton, J., Lucey, S., Kearney, K., Mackinson, S., Pan, M., Platts, M., & Walters, C. (2016). Ecopath with Ecosim as a model-building toolbox: Source code capabilities, extensions, and variations. *Ecological Modelling*, *319*(January), 178–189. https://doi.org/10.1016/j.ecolmodel.2015.06.031

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.

Treloar, N. J., Fedorec, A. J. H., Ingalls, B., & Barnes, C. P. (2020). Deep reinforcement learning for the control of microbial co-cultures in bioreactors. *PLoS Computational Biology*, *16*(4), e1007783. https://doi.org/10.1371/journal.pcbi.1007783

Valletta, J. J., Torney, C., Kings, M., Thornton, A., & Madden, J. (2017). Applications of machine learning in animal behaviour studies. *Animal Behaviour*, *124*(February), 203–220. https://doi.org/10.1016/j.anbehav.2016.12.005

Villarroel, J. A., Taylor, J. E., & Tucci, C. L. (2013). Innovation and learning performance implications of free revealing and knowledge brokering in competing communities: Insights from the Netflix prize challenge. *Computational and Mathematical Organization Theory*, *19*(1), 42–77. https://doi.org/10.1007/s10588-012-9137-7

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., … Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, *575*(7782), 350–354. https://doi.org/10.1038/s41586-019-1724-z

Walters, C. J., & Hilborn, R. (1978). Ecological optimization and adaptive management. *Annual Review of Ecology and Systematics*, *9*(1), 157–188. https://doi.org/10.1146/annurev.es.09.110178.001105

Wang, X., Cheng, J., & Wang, L. (2020). A reinforcement learning-based predator-prey model. *Ecological Complexity*, *42*(March), 100815. https://doi.org/10.1016/j.ecocom.2020.100815

Weber, T., Racanière, S., Reichert, D. P., Buesing, L., Guez, A., Rezende, D. J., Badia, A. P., Vinyals, O., Heess, N., Li, Y., Pascanu, R., Battaglia, P., Hassabis, D., Silver, D., & Wierstra, D. (2017). Imagination-augmented agents for deep reinforcement learning (version 2). *arXiv*. https://doi.org/10.48550/ARXIV.1707.06203

Wilson, K. A., McBride, M. F., Bode, M., & Possingham, H. P. (2006). Prioritizing global conservation efforts. *Nature*, *440*(7082), 337–340. https://doi.org/10.1038/nature04366

Worm, B., Barbier, E. B., Beaumont, N., Duffy, J. E., Folke, C., Halpern, B. S., Jackson, J. B. C., Lotze, H. K., Micheli, F., Palumbi, S. R., Sala, E., Selkoe, K. A., Stachowicz, J. J., & Watson, R. (2006). Impacts of biodiversity loss on ocean ecosystem services. *Science*, *314*(5800), 787–790. https://doi.org/10.1126/science.1132294

Worm, B., Hilborn, R., Baum, J. K., Branch, T. A., Collie, J. S., Costello, C., Fogarty, M. J., Fulton, E. A., Hutchings, J. A., Jennings, S., Jensen, O. P., Lotze, H. K., Mace, P. M., McClanahan, T. R., Minto, C., Palumbi, S. R., Parma, A. M., Ricard, D., Rosenberg, A. A., … Zeller, D. (2009). Rebuilding global fisheries. *Science (New York, N.Y.)*, *325*(5940), 578–585. https://doi.org/10.1126/science.1173146

Xu, L., Perrault, A., Fang, F., Chen, H., & Tambe, M. (2021). Robust reinforcement learning under minimax regret for green security. *arXiv*. http://arxiv.org/abs/2106.08413

Zhou, Z., Li, X., & Zare, R. N. (2017). Optimizing chemical reactions with deep reinforcement learning. *ACS Central Science*, *3*(12), 1337–1344. https://doi.org/10.1021/acscentsci.7b00492

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Lapeyrolerie, M., Chapman, M. S., Norman, K. E. A., & Boettiger, C. (2022). Deep reinforcement learning for conservation decisions. *Methods in Ecology and Evolution*, *13*, 2649–2662. https://doi.org/10.1111/2041-210X.13954