Received: 6 July 2023 | Accepted: 29 January 2024

DOI: 10.1111/2041-210X.14300

Methods in Ecology and Evolution

RESEARCH ARTICLE

Connectivity conservation planning through deep reinforcement learning

Julián Equihua[1] | Michael Beckmann[1] | Ralf Seppelt[1,2,3]

[1]Department of Computational Landscape Ecology, Helmholtz Centre for Environmental Research (UFZ), Leipzig, Germany

[2]Institute of Geoscience and Geography, Martin-Luther-University Halle-Wittenberg, Halle (Saale), Germany

[3]German Centre for Integrative Biodiversity Research (iDiv), Leipzig, Germany

Correspondence
Julián Equihua
Email: julian.equihua@ufz.de

Funding information
Deutscher Akademischer Austauschdienst, Grant/Award Number: 91713889

Handling Editor: Lorna Hernandez-Santin

Abstract

1. The United Nations has declared 2021–2030 the decade on ecosystem restoration with the aim of preventing, stopping and reversing the degradation of the ecosystems of the world, often caused by the fragmentation of natural landscapes. Human activities separate and surround habitats, making them too small to sustain viable animal populations or too far apart to enable foraging and gene flow. Despite the need for strategies to solve fragmentation, it remains unclear how to efficiently reconnect nature. In this paper, we illustrate the potential of deep reinforcement learning (DRL) to tackle the spatial optimisation aspect of connectivity conservation planning.
2. The propensity of spatial optimisation problems to explode in complexity depending on the number of input variables and their states is and will continue to be one of its most serious obstacles. DRL is an emerging class of methods focused on training deep neural networks to solve decision-making tasks and has been used to learn good heuristics for complex optimisation problems. While the potential of DRL to optimise conservation decisions seems huge, only few examples of its application exist.
3. We applied DRL to two real-world raster datasets in a connectivity planning setting, targeting graph-based connectivity indices for optimisation. We show that DRL converges to the known optimums in a small example where the objective is the overall improvement of the Integral Index of Connectivity and the only constraint is the budget. We also show that DRL approximates high-quality solutions on a large example with additional cost and spatial configuration constraints where the more complex Probability of Connectivity Index is targeted. To the best of our knowledge, there is no software that can target this index for optimisation on raster data of this size.
4. DRL can be used to approximate good solutions in complex spatial optimisation problems even when the conservation feature is non-linear like graph-based indices. Furthermore, our methodology decouples the optimisation process and the index calculation, so it can potentially target any other conservation feature implemented in current or future software.

KEYWORDS
connectivity conservation planning, deep reinforcement learning, ecological restoration, machine learning, spatial optimisation, systematic conservation planning

1 | INTRODUCTION

Currently, 40% of all land has been transformed into agriculture (UN, 2022), accounting for 88% of global deforestation (FAO, 2022), devastating or starkly modifying the habitats where many species live. Habitat areas become separated and surrounded, making them too small to sustain viable animal populations or too far apart to move between them to feed and reproduce, eventually capping gene flow (Schlaepfer et al., 2018). The grave harm fragmentation causes on so many levels is clear; the United Nations has declared 2021 to 2030 the decade on ecosystem restoration, which must evidently aspire, among other things, to reconnect nature (Pimm et al., 2021). Despite global urgency, it remains unclear how to execute the measures needed to achieve this goal in an effective way. Analyses that resort to spatial optimisation in order to inform where and how to implement conservation actions are considered systematic conservation planning (SCP). SCP can be traced back to the works of Jamie Kirkpatrick (Kirkpatrick, 1983, 1986) in which a systematic method was first presented to identify priority conservation areas for species. Now SCP comprises a whole body of research mostly aimed at establishing and extending reserves for the direct protection of ecosystems, biological assemblages, species and populations (Margules & Pressey, 2000). Naturally it is possible and important to complement these conservation objectives with other measures, which has given rise to other planning problems, such as ecological restoration planning (Justeau-Allaire et al., 2021) or recently connectivity conservation, defined by Keeley et al. (2019) as 'an emergent approach to counteracting landscape fragmentation and enhancing resilience to climate change at local, national, and global scales'. We refer to the methods that inform on how and where to stop or reduce fragmentation by preserving corridors or reconnecting habitats with stepping stones (Keeley et al., 2019) as connectivity conservation planning (CCP).

An SCP optimisation problem formulation starts by selecting a region and designing a conservation feature to target. Then, the actual conservation objective is formulated; this defines the way that decisions on parcels or the so-called planning units located in the region affect the conservation feature. A given set of decisions on the planning units is called a solution to its SCP problem. The mathematical nature of these decisions defines how the problem itself can be approached. In its simplest form, each planning unit is either selected or not, and thus solutions are expressed as sums of binary variables also called integer programs (Sierra-Altamiranda et al., 2020). Such problems belong to the domain of integer linear programming (ILP). The decisions can also be continuous; for example, proportions pf planning units may be selected, making the problem of the domain of linear programming. There is a wide variety of SCP goals that can be tackled; we refer to Billionnet (2013) and Hanson et al. (2023) for a good overview of many problems, their mathematical formulations and solutions using ILP. In most of these examples, planning units can only either be selected or not. Even if each decision variable comprises only two states, the number of possible solutions grows exponentially despite the number of units increasing linearly.[1] ILP problems tend to be NP-hard (Sierra-Altamiranda et al., 2020). This is a specific example of an incredibly widespread issue that comes in many flavours: the combinatorial explosion problem (Schuster, 2000) and state-space explosion (Clarke et al., 2012), which are manifestations of the curse of dimensionality (Alagador & Cerdeira, 2022). For optimisation, it boils down to the polynomial or even exponential growth of problem complexity, depending on the number of input variables and their states. In SCP, this is only exacerbated by the rapid increase in the quantity and resolution of spatial data coupled by advances in ecological knowledge that result in more elaborate conservation features. The propensity of optimisation problems in SCP to explode in complexity is and will continue to be one of its most serious obstacles.

In this paper, we explore the potential of an emerging field of artificial intelligence, deep reinforcement learning (DRL), for SCP. We illustrate this by using DRL to approximate good solutions to spatial optimisation problems in a CCP setting. DRL has attracted enormous attention because of its potential to excel in complex sequential decision-making tasks. DRL was first put in the public spotlight for models that exceed human performance in board and video games (Silver et al., 2016; Vinyals et al., 2019); it has gone on to enable the discovery of faster matrix multiplication algorithms (Fawzi et al., 2022), the design of optimal tax policies (Zheng et al., 2022) and constitutes part of ChatGPT's training process (Christiano et al., 2017; OpenAI, 2023). The central idea of DRL is to incrementally train a deep neural network to achieve a goal by exploring sequences of actions within a virtual environment, and the potential to tackle conservation problems is only beginning to be explored (Lapeyrolerie et al., 2022; Silvestro et al., 2022; Turchetta et al., 2022). It should be considered as an additional toolkit for SCP for a number of reasons:

1. DRL has proven to overcome the curse of dimensionality in many cases (Arulkumaran et al., 2017). It has been specifically used to learn good heuristics for traditional NP-hard combinatorial optimisation problems like the travelling salesman (Bello et al., 2017; Li et al., 2022).
2. DRL it is suitable for solving problems with little prior knowledge and high uncertainty (Francois-Lavet et al., 2018).
3. Since DRL environments themselves are highly customisable software, they can access data of any kind and also interact with any other model or method (Lapeyrolerie et al., 2022).
4. DRL models problems as games. This can encourage the definition of intuitive objectives, focusing on what to achieve instead of the formal problem formulation (Degrave et al., 2022).

[1] 2^N where $N = 1, 2, 3, \ldots$ is the number of planning units.

The increasing importance habitat connectivity plays in conservation efforts is clear. It underlays the UN decade on ecosystem restoration and the Convention on Biological Diversity (CBD) (Keeley et al., 2019), namely of the recent Kunming-Montreal Global Biodiversity Framework (section H) targets 1, 2, 3, 12 and 13 (Biosafety Unit, 2023). However, the most recent connectivity metrics have seldom been used to formulate spatially explicit optimisation problems for CCP. There are many proposed methods to assess the degree of connectedness of a landscape. A popular class of indices considers habitat patches as nodes and the degree of accessibility between them as edges of a graph. The Integral Index of Connectivity (IIC; Pascual-Hortal & Saura, 2006) and the Probability of Connectivity Index (PCI; Saura & Pascual-Hortal, 2007) are the graph-based indices used the most to analyse structural and functional connectivity (Hashemi & Darabi, 2022; Keeley et al., 2021). They have been proven to be good predictors of, among other things, species occupancy and occurrence patterns (Awade et al., 2012; Pereira et al., 2011); designed to be sensitive to even subtle landscape changes, both of which provide a basis for quantifying the importance of landscape elements (Bodin & Saura, 2010); and therefore used for scenario analyses for CCP (Engelhard et al., 2017, Martinez Pardo et al., 2023). However, it can be especially daunting to target them in spatial optimisation problems, as they are nonlinear and computationally costly (Justeau-Allaire et al., 2021). To the best of our knowledge, there are only four previous works in which either the IIC or PCI are targeted for optimisation. In Rubio et al. (2015), a brute force approach was proposed, testing all possible combinations of potential patch removals and their effect on both IIC and PC. Xue et al. (2017) proposed a classical mixed integer program formulation to optimise PCI on a general graph; they chose a set of edges to protect in order to best maintain connectivity. In Hamonic et al. (2023) the problem of optimising the PCI of a landscape was considered under a restricted budget. This was modelled as a discrete optimisation problem directly on graph representations. Their formulation starts on a landscape assumed to be degraded and aims to improve the size/quality of existing patches or create stepping stones, here modelled as new links, from a set of predetermined feasible options. Justeau-Allaire et al. (2021) considered a forest/non-forest raster map: non-forest cells were taken as planning units and thus restorable. Constrained programming (CP) was used to find a set of these cells whose restoration resulted in a maximal improvement of the IIC while also satisfying some landscape configuration constraints. Built upon this, they have also developed the restoptr R package, a flexible framework for ecological restoration planning that leverages the Choco-solver CP library for optimisation (Justeau-Allaire et al., 2023).

As an alternative approach, here we will show how to specify custom spatial environments to optimise the IIC and PCI using current DRL software and standards. Based on our findings we suggest future research perspectives to explore increasingly complex datasets and goals using natural modifications of the methodology proposed here.

2 | MATERIALS AND METHODS

2.1 | The IIC and the PCI

The rapid development of sensors amounts to an increasing number of essential biodiversity variables that can be measured from space (Skidmore et al., 2015, 2021). Naturally, land cover and thus fragmentation are among them. The nature of data produced by earth observation means that most products that become available will have a gridded format. Therefore, we will focus on raster inputs when seeking optimal landscape configurations that yield a maximal improvement in IIC and PCI. Although the formulas of the IIC and PCI are not particularly complex, it should be noted that they require several computationally intensive steps to calculate. From an input raster, they first require a segmentation process in which adjacent habitat pixels are clumped into single patches and then all the pairwise distances between them are calculated. After that, they require successive application of a shortest-path algorithm to calculate the least-cost path between each of the pairs of patches; a path consists of a sequence of steps in which no patch is visited more than once (Pascual-Hortal & Saura, 2006). Any change in the landscape raster of interest results in the need to update its number of patches and/or their sizes. As well as potentially many connection paths in which they participate.

The IIC ranges from 0 to 1, where 1 means that a landscape is fully occupied by habitat. It is built upon a binary connection model. This means that two distinct habitat patches are only directly reachable from one another if they are closer than a predefined distance threshold. They could still be reachable by a path that crosses other patches. The IIC is given by

$$\mathrm{IIC} = \frac{1}{A_L^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{a_i a_j}{1 + \mathrm{nl}_{ij}}, \tag{1}$$

where a_i and a_j are the areas of the habitat patches i and j, A_L is the total landscape area, and nl_{ij} is the number of links in the shortest path between i and j.

The PCI is a natural extension of the IIC and is defined as the probability that two animals randomly placed within a landscape fall into interconnected habitat areas. As its name suggests, it assumes a probabilistic connection model. It uses probability to express the feasibility of movement between habitat nodes. For example, calculated using a decreasing exponential function of the inter-patch distance, $p_{ij} = e^{-k \times d_{ij}}$, where d_{ij} is the distance between the patches i and j, and k is chosen so that the function matches a desired probability distance value. The PCI is given by

$$\mathrm{PCI} = \frac{1}{A_L^2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j p_{ij}^*, \tag{2}$$

where a_i and a_j are the areas of the habitat patches i and j, A_L is the total landscape area, and p_{ij}^* is then defined as the maximum product probability of all possible paths between patches i and j. The product probability is the multiplication of all p_{ij} belonging to each step on the corresponding path.

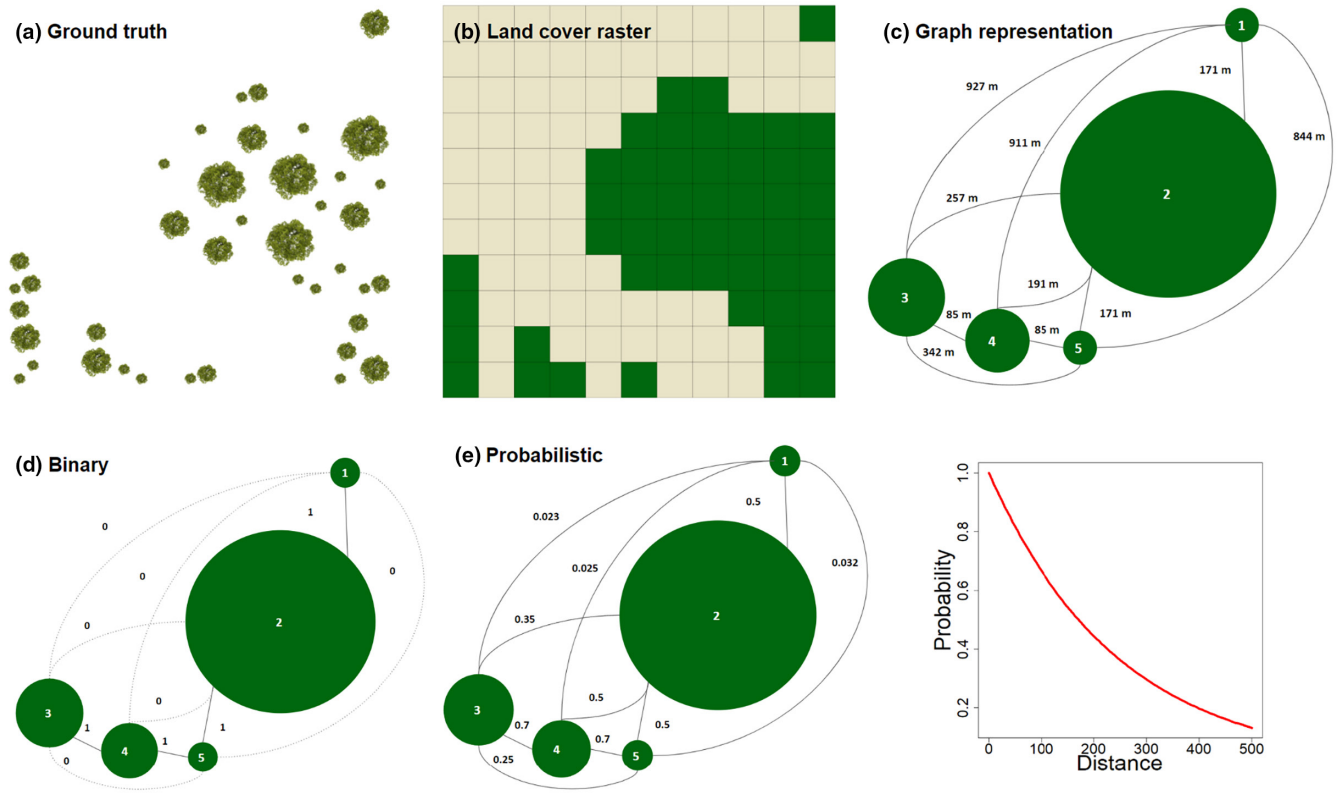

FIGURE 1 (a) The ground truth terrain is sampled in some way, for example by earth observation. Satellite imagery is used in supervised classifications to produce land cover cartography. (b) Land cover rasters are segmented in order to delineate distinct habitat patches. (c) From here, a landscape can be modelled as a graph, the extension of the habitat patches become the node sizes and the distance between them the edge weights. (d) With a distance threshold of 171 m the binary connection model, some habitat patches are considered directly connected, :=1, and some not, :=0. (e) In the probabilistic connection model and fixing a 171 m distance to match a probability of 0.5, all patches are reachable from one another albeit sometimes with a small probability.

To illustrate the binary and probabilistic connection models, we will use a dataset found in restoptr.[2] This raster data refers to a mining area located in the north of the main island of New Caledonia, Mount Kaala. It comprises binary values, where zero refers to non-habitat and one to habitat pixels. We took a small subset of 11×11 pixels from the raster. These have an 85 m spatial resolution; we chose a distance threshold of 171 m, slightly larger than two pixels. For the probabilistic connection, we chose a decreasing exponential function in which the previous distance matches a probability of 0.5, Figure 1.

Proponents of the IIC and PCI developed software, Conefor Sensinode 2.2, in the c++ programming language for their efficient calculation. We will take advantage of the customisability of DRL environments in order to illustrate the fact that we can leverage external software for our purposes, for example, to produce the signal we are interested in optimising.

2.2 | The reinforcement learning problem

Reinforcement learning (RL) comprises a class of solutions for sequential decision-making tasks. There are many works that can serve as an introduction to RL, Sutton and Barto (2018) is the germinal book on RL in general, Lapeyrolerie et al. (2022) is a great overview of DRL aimed at ecologists. For the reminder of this section, we will mostly follow Francois-Lavet et al. (2018). The main idea of RL is to learn the best way to achieve a goal through trial and error by interacting with an environment. The learner in this iterative process is an artificial agent which starts by receiving a (potentially partial) observation, $w_0 \in W$, of the initial state of an environment, $s_0 \in S$. Subsequently, it will take an action $a_t \in A$, which results in a reward $r_t \in R$, and makes the environment transition to the next state $s_t \in S$, which in turn emits an updated observation to the agent, $w_t \in W$. This loop occurs for every time step $t = 1, 2, 3, \ldots, n$ until a user-defined stopping criterion is met, then the process starts over. The agent is seeking to maximise the sum of the received rewards, and it is in this replaying of episodes that the agent devises a better strategy. This strategy is encoded in a policy, $\pi \in \Pi$, which defines how an agent, given an observation, will decide on an action. One way to model the reinforcement problem is to utilise a finite Markov decision process (MDP), a mathematical framework for modelling decision making in discrete time. It was introduced in the 1950's (Bellman, 1957) and was first used to model dynamic programming problems. We refer to (Marescot et al., 2013) for an introduction to MDPs. An MDP

[2]See the vignette: using historical data to set ecological restoration targets.

provides a natural way to model the three components of an RL problem: environment, agent and reward. Formally, it is a 5-tuple (S, A, T, R, γ), where

- S is the state space,
- A is the action space,
- $T: S \times A \times S \to [0, 1]$ is the transition function (set of conditional transition probabilities between states),
- $R: S \times A \times S \to \mathbb{R}$ is the reward function, where R is a continuous set of possible rewards in a range, and
- $\gamma \in [0, 1]$ is a discount factor, which controls if emphasis is immediate rewards (0) or future rewards,

in which the Markov property holds; this is so that the future of the process only depends on the current observation. If the state of the environment is fully observable, then the states and the observations are always the same.

RL considers the problem of finding a policy $\pi(s, a) \in \Pi$, which maximises the expected return (value function) $V^{\pi}(s): S \to \mathbb{R}$ with

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^{k} r_{t+k} \mid s_t = s, \pi\right], \tag{3}$$

where $r_t = \underset{a \sim (\pi s_t, \cdot)}{\mathbb{E}} R(s_t, a, s_{t+1}), \mathrm{IP}(s_{t+1} \mid s_t, a_t) = T(s_t, a_t, s_{t+1})$ with $a_t \sim \pi(s_t, \cdot)$.

The policy is a function from observations to actions. Since its goal is to distinguish good actions, it must depend in some way on a value function. It is learnt through the reward in multiple ways. For example, seeking immediate rewards is not always the best way to achieve a future goal. It may be more advantageous to learn an estimator of the cumulative reward that can be obtained from each of the possible states the environment can be in and to take action based on these valuations. Methods that learn the value function and base their policies directly on it are referred to as value-based. It is also possible to learn a direct parametrisation of the policy. For example, actor-critics learn both functions but are considered policy-based methods. The policy is the actor, and the value function critiques the policy in order to improve it. Classical RL is limited by the curse of dimensionality because the value and/or policy functions are exhaustively constructed in tabular form. DRL replaces this with deep neural networks, as they have proven to be excellent function approximators (Elbrächter et al., 2021; Hornik et al., 1989), thus making DRL applicable to problems with large state and action spaces.

DRL is arguably still in its early stages. Its software ecosystem is not as mature as others, and it is still in process of standardisation. For DRL applications developed in the python programming language, a notable standard is the Gymnasium toolkit (Gym), which offers a standard API to communicate between models and environments. Of the elements of the general RL problem, Gym encompasses the environment states, the reward, and the actions the agent can perform. For model training, one must draw on one of a multitude of packages with different implementations of learning algorithms. We opted for using Stable-Baselines3 (SB3) as it is, in our opinion, one of the simpler to use packages with which to develop DRL applications. One additional advantage of DRL is that it has naturally built on advances in other areas of machine learning like computer vision in order to learn good representations of environment states. For example, in Mnih et al. (2015), multiple atari games were mastered by feeding crude frames (images) into deep convolutional neural networks (CNNs) in order to approximate optimal value functions, effectively learning how to play 'directly from pixels'. In SB3, the policies by default are coupled with some feature extractor, for example a CNN for images. In Table 1, we present a schematic of the required software components from both Gym and SB3 in order to specify a full custom DRL training environment.

2.3 | A custom environment to target the IIC for spatial optimisation on a small landscape

We developed two examples of custom geospatial Gym environments for solving CCP problems. Both read in land-cover rasters

TABLE 1 The necessary components from Gym and SB3 for a custom deep reinforcement learning (DRL) application.

Software component	Description	Package	DRL component
The general environment class, gym.Env	A subclass must be created for a custom environment. Requires several methods and two spaces	Gym	Environment
The init() method	To initialise the training environment		
The observation space	Required by init(). One of the gymnasium.spaces		
The close() method	Handles how to close the environment, important when external software is used		
The step() method	How an agent performs an action, updates the environment, delivers the reward and a new observation		Environment and Agent
The reset() method	Restarts the environment for a new episode and returns an initial observation		
The render() method	Visualisation of the learning process and environment states		
The action space	Required by init(). One of the gymnasium.spaces		Agent
Reinforcement learning algorithms	One of many on or off-policy learning algorithms	SB3	
Policy networks	One of three default feature extractors or a user-defined network		

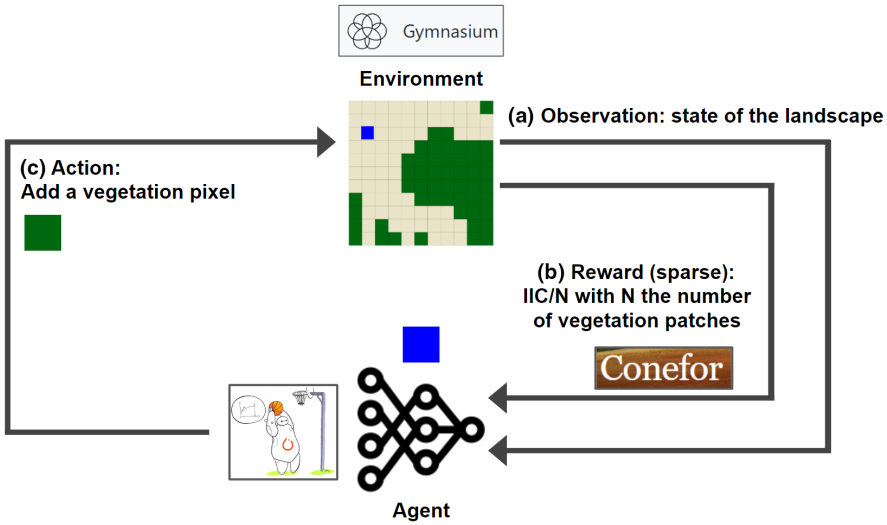

FIGURE 2 The training loop starts with a certain budget of habitat pixels to be placed, and then (a) the (Gym) environment emits an observation, the state of the landscape raster, to the agent. (b) The agent processes the observation (SB3) and decides where on the landscape to move and place a vegetation patch; the budget is reduced by 1. (c) The environment emits a reward to the agent, if budget = 0 the reward is the Integral Index of Connectivity (IIC) value of the current landscape divided by N, the number of vegetation patches. It is 0 in any other case. When the reward requires the IIC, the environment makes a call to the Conefor software for its calculation.

to serve as their state spaces. We first present an environment to maximise the IIC of the simple landscape of size (11×11) that was used to illustrate the binary and probabilistic connection models. It includes only two classes, habitat and nonhabitat, and has a spatial resolution of 85 m. We used the restoptr (Justeau-Allaire et al., 2023) package to verify the optimality of our solutions. We kept the default threshold of the edge-to-edge distance of at most one pixel (Justeau-Allaire et al., 2021) to parameterise the IIC. This means that only habitat pixels that directly touch (rook neighbourhood) will be considered connected and simultaneously part of the same habitat patch. This encourages solutions that directly connect habitat patches, knowledge that we used to design our reward. Using domain knowledge to increase the sample efficiency of DRL or to get it to work altogether is commonly known as reward shaping; designing good rewards can be a feat.

We first formulated the connectivity planning problem as an MDP in order to solve it using DRL. The way the learning procedure unfolds is as follows: At each time step, the agent receives the current state of the landscape raster as an observation (the state is fully observable, $w_t = s_t$). It performs an action that consists of warping to a location on the landscape, virtually restoring a nonhabitat pixel to natural habitat and subtracting from the restoration budget. This continues until a predetermined budget is spent, with which the episode ends. The agent receives a reward of 0 at every step except at the end of the episode in which it receives the IIC value of the final landscape divided by the number of distinct habitat patches.[3] The IIC is calculated by calling Conefor Sensinode 2.2. This type of reward, which is seldom informative, is called a sparse reward. DRL is known to overcome the sparse reward problem to some extent, although since exploration commonly beings at random, it can fail for problems with very large state spaces and very sparse rewards. For example, in the Montezuma Revenge Atari 2600 game, initial random exploration translates into an informative reward about every half-million steps (Salimans & Chen, 2018). In our example, the state space is modest and the time horizon in which an informative reward is emitted is short due to the limited restoration budget. The computational cost of calculating the IIC in each time step greatly outweighs the sparsity of the reward. This reward also alleviates the fact that the order of patch placements is irrelevant for the final IIC value. As a final detail, the episode is terminated if the agent repeats an action with the aim of discouraging such behaviour. In Figure 2, we provide a schematic of the DRL training loop.

We selected the proximal policy optimisation (PPO) learning algorithm to train the agent, a policy-based learning algorithm popular due to its broad applicability and usually requiring little hyperparameter tuning (Schulman et al., 2017). We chose hyperparameters by trial and error and by studying tuned benchmark environments in the RL Baselines3 Zoo. We developed methods to visualise the training process and behaviour of trained agents, which is necessary to tune the performance of DRL models. We trained two separate agents for 60,000 time steps, one with a budget of 6 habitat pixels and the other of 5 habitat pixels. The training process for each takes around 4 min on an AMD Ryzen 55600X 6-Core Processor 3.70 GHz with 32 GB of RAM. For a complete description of DRL Model 1, see Table 2.

Furthermore, we used restoptr (Justeau-Allaire et al., 2023) to optimise the same raster landscape subject to the same budgets. CP is an exact constrained optimisation technique based on

[3]Must be greater than 0 so a non-empty landscape.

TABLE 2 Summary of the characteristics of deep reinforcement learning Models 1 and 2.

Environment	Model 1	Model 2	Observations
Connection model	Binary	Probabilistic	
Distance threshold	85 m	2000 m	
Observations	Landscape raster	RGB image: initial landscape, patch placements, decreasing budget on remaining restorable pixels	
Actions	Agent movement	Agent movement and patch size (two dimensions)	
Reward	(sparse) Integral Index of Connectivity/N, N the number of distinct patches	Probability of Connectivity Index (%)	
Budget	5 and 6 pixels	20,000 units	Episode ends when budget reaches 0
Learning algorithm	Proximal policy optimisation (PPO)	PPO	PPO
Learning algorithm			
Feature extractor	MlpPolicy	CNNPolicy	
Training steps	60,000	300,000	
Learning rate	0.001	0.001	
Gamma	0.99	0.99	Discount factor for rewards
Entropy coefficient	0.01	0	Can encourage exploration (although it has been shown empirically that this is not always the case)
Clip range	0.2	0.2	Roughly the probability that an action cannot change by more than factor 1 + clip range
Random seed	7	666	

automated reasoning, meaning that if enough runtime is available, it guarantees global optimality, which is not the case for DRL. This allowed us to compare the solution to which DRL arrives against the known best.

2.4 | A custom environment to target the PCI for spatial optimisation on a large landscape with additional cost and spatial configuration constraints

We applied the previous methodology to a much larger real-world dataset. First, we obtained the official vector layer of vegetation and land use in Mexico (1:250,000) from the National Institute of Statistics and Geography (INEGI), also known as Series VII. This layer includes 220 classes of land cover and land use that include agriculture, urban, and primary and secondary natural vegetation. Subsequently, these classes were remapped to refer only to seven fundamental classes of interest: natural forests (including mangroves, rainforests and temperate forests), grasslands (including all natural grassy vegetation), bare ground, urban settlements, water bodies, agriculture (including human-induced grasslands) and secondary herbaceous vegetation. This layer was rasterised at a spatial resolution of 250 m. Finally, Mexico's road network from the Digital Map of the World, available as a line vector layer, was rasterised at the same spatial resolution and added to the map as an additional class (roads). We selected a landscape (249 × 249) in size and assigned costs to the restoration of each non-forest class to natural forests. Another difference from the previous environment is that the agent can now choose the size of the patch to "restore" from a selection of square patches originating from the range 2×2 to 8×8. The reward function is no longer sparse; each training step, the agent receives as feedback the percentage change in PCI. Between time steps n and $n + 1$, it is given by

$$\Delta\mathrm{PCI} = 100 \times \frac{\mathrm{PCI}^{n+1} - \mathrm{PCI}^{n}}{\mathrm{PCI}^{n}}. \quad (4)$$

At the end of the episode, it receives the total percentage change in PCI. Naturally, these rewards have a high degree of redundancy, but since the state space is so large, it makes sense to provide information at each time step in order to estimate the value function. The ΔPCI's are influenced by the order of patch placements, allowing the collection of more diverse data. But in the end, the most important thing is the total gain which, as in the previous example, is not influenced by the order of placements.

The learning algorithm also consisted of PPO and its complete description is in Table 2. In this example, the agent receives the observations as RGB images. Their first channel contains the initial landscape raster, where the pixels are the land cover restoration costs. The second channel is a raster where the evolution of the placement of vegetation patches is recorded. The third channel consists of the remaining budget located only on pixels, which are still subject to restoration. These image observations

are processed by a CNN feature extractor for the policy neural network. The training process in this case took a total of 28 h on the same hardware.

3 | RESULTS

We presented two examples of how to tackle CCP problems using DRL. The first was on an arguably small and simple raster landscape in which the IIC was targeted and in which the reward for the DRL agent was sparse (only emitted at the end of a training episode). The second, targeting the PCI on a larger raster landscape, with different costs associated with restoring each land use class, additional spatial configuration constraints, and a cumulative reward (sequentially emitted after each habitat patch placement). After training, the best models were deployed on the landscape rasters to obtain final solution landscapes. For the first example, we know the optimum. Using restoptr, we can be sure of the level of IIC achieved by a global optimum solution to each problem. Even slight modifications of spatial optimisation problems can yield unexpected changes in the complexity of the problem. For example, the simple example with a budget of six habitat pixels is optimised by restoptr in around 10 min. But the case with a budget of five only takes 1 min. DRL takes around 4 min in both cases, beating restoptr in the first, even with the well-known sample inefficiency of DRL. Although this is probably very specific to this problem and to the fact that we shaped the reward function in order to aggressively seek less and larger habitat patches which could make the comparison unfair. It is also clear that for the case in which the budget is five a multitude of global optimums exist. Once the patches in the lower part of the map are connected, which only requires four habitat pixels, the remaining one can be placed anywhere connected to the large resulting habitat patch to achieve the maximum IIC level. In the case of a budget of 6, only two global optium solutions exist, Figure 3a. Our DRL models converge to only one of the globally optimal solutions. One advantage of restoptr is that it can find all of them, given enough runtime.

The objective of Model 2 was to optimise a large landscape in order to achieve a maximal improvement in its PCI value. Unfortunately, there is no way to assess how close our final solution is to a global optimum as there simply is not any readily available software that can target the PCI for optimisation on a raster of this size and nature. In the absence of an exact solver with which to benchmark, one possibility is to compare a solution with a large amount of random ones or with those produced by a multi-armed bandit model when appropriate (Haj-Ali et al., 2019). It can be seen that this DRL model performs much better than what was achieved by mere chance in the same number of episodes, with still room for improvement after 28 training hours, Figure 3b. The final solution achieves an increase of 66% in the overall forest connectivity measured by the PCI. It does so by restoring 0.2% of the restorable extension and spending 20,000 units, which represents 0.1% of what would be spent restoring the entire landscape.

In the DRL environments we developed, some constraints are analogous to what is encountered, for example, in linear programming. Budget and cost constraints act in the same way. But DRL environments allow for a multitude of ways in which to guide the exploration of solutions. For example, in our first case study, enforcing an episode reset if the agent repeated an action made the agents quickly learn they should not do that. This sort of punishment can be softer, for example multiplying the current reward by some quantity smaller than 1. Since solutions that consist of sparse individual pixels may be undesirable, different mathematical constraints have been devised to avoid them (Justeau-Allaire et al., 2021). We proposed actively targeting solutions that consist of connected and clumped-up pixels by allowing the agent to only choose between a minimum and maximum size for restored forest patches. It can be seen that the agent prefers vegetation patches of the maximum allowed size of 8×8 pixels, but, in the special case in which it targets roads to reconnect forests, it sometimes chooses smaller patches to restore.

4 | DISCUSSION

We have shown that DRL can be leveraged to optimise complex connectivity indices over raster landscapes of varying sizes and complexity. Since our methodology decouples the optimisation process and the index calculation, it can potentially target any other conservation feature implemented in current or future software with relative ease.

We stress that exact optimisation methods should be used whenever feasible. Unfortunately, intractable problems will continue to appear in spatial optimisation analyses for SCP. We mentioned four works in which either the IIC or PC are targeted for optimisation. As mentioned in Hamonic et al. (2023), the brute-force approach proposed in Rubio et al. (2015) would naturally eventually find global optimums but was shown to be impractical for landscapes with more than 20 habitat patches, and the mixed integer approach proposed in Xue et al. (2017) did not scale to landscapes with more than a few hundred pixels on a grid dataset. In Hamonic et al. (2023), the PC was targeted by modelling the problem with discrete optimisation directly on graph representations. They propose an ingenious problem simplification, which allows the approximation of good solutions to even large problems. Unfortunately, these solutions are not spatially explicit; they can inform where a reconnection is optimal on a graph but not the actual geographical location or size of the habitat patch to achieve it, for example when working with land cover maps. For this purpose, the most promising of these exact methods is the CP approach proposed in Justeau-Allaire et al. (2021) although, for now, restricted to the indices incorporated by the authors into restoptr. CP guarantees global optimality, which DRL simply cannot. An optimal policy is guaranteed to exist (Sutton & Barto, 2018). From the value function (Equation 3), this optimal policy π^* satisfies

$$V^{\pi^*}(s) \geq V^{\pi}(s), \tag{5}$$

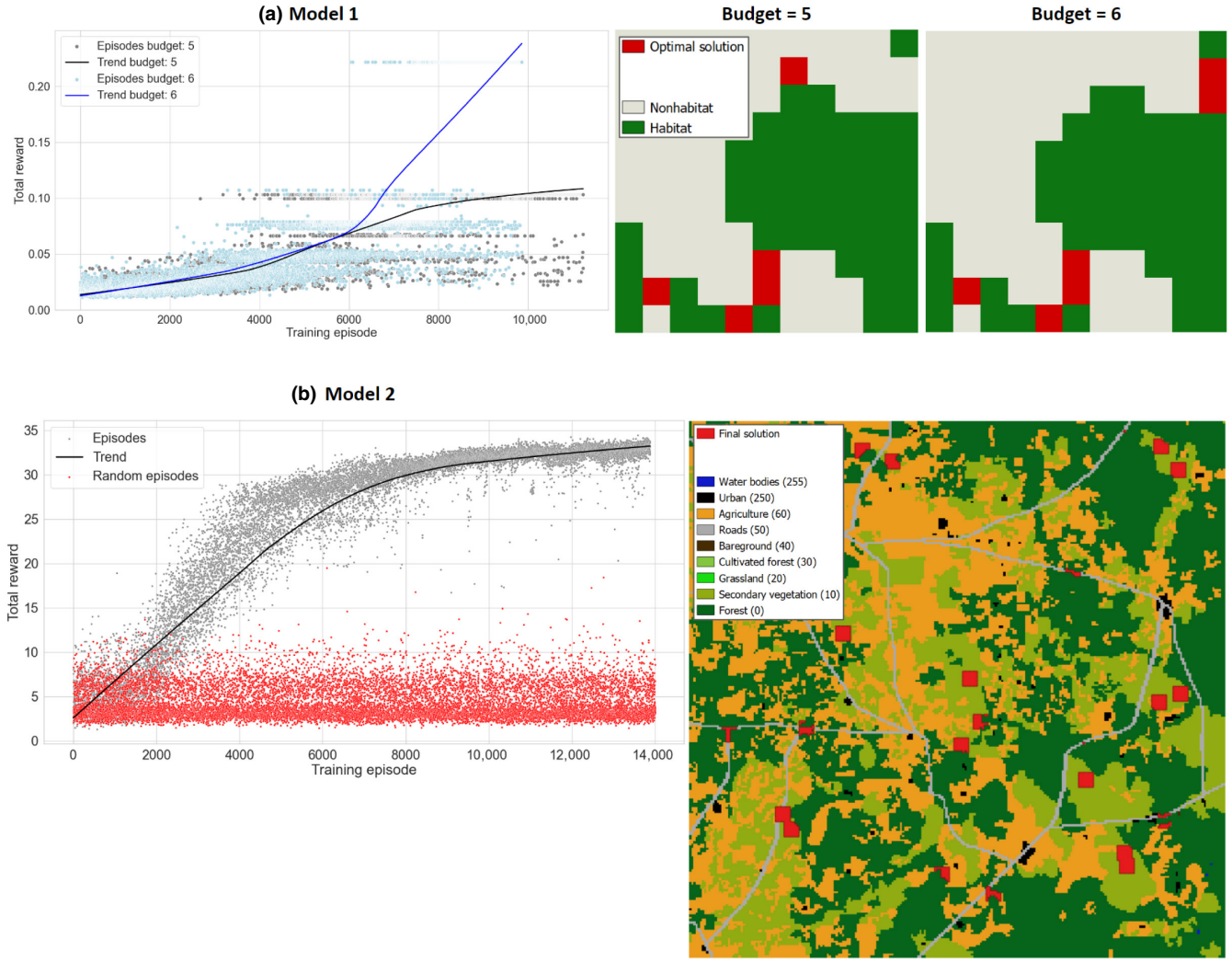

FIGURE 3 Performance of Models 1 and 2. (a) Training process for Model 1. Top left: The agents steadily learn over a span of 10,000 episodes and converge to an optimum solution. Top right: Only one of multiple optimum solutions for each of the budgets is attained. (b) Training process for Model 2. Top left: The DRL agent beats a random strategy after around 6000 training episodes, at 14,000 it is well away from what was achieved by chance. The model is beginning to plateau at 28 h of training but could still reach a better performance albeit with diminishing returns. Top right: The final solution that corresponds to an overall increment in the Probability of Connectivity Index of 66%.

for any state s and any other policy π. But DRL cannot always manage to learn the optimal policy, it can be difficult to know if DRL has converged at all. This can be highly dependent on hyperparameter selection and even on the initial random seed; in general, DRL models have been shown to be notoriously unstable during training (Nikishin et al., 2018). To make matters worse, it is not always clear if the reward that is provided to the agent motivates the desired behaviour, there are many examples of seemingly reasonable reward functions that end up being grossly misaligned in respect to the true optimisation objective, this issue has been called specification gaming (Krakovna et al., 2020). Since DRL comes with no guarantees, the best bet is to produce a large amount of diverse experiments, ideally in combination with automated hyperparameter tuning for which there are software like Ray and Optuna. However, as we have shown in Model 2, DRL has a tendency towards overlong training times. Its use for larger problems is really only possible with high performance computing as deep neural networks can be trained in parallel. For this reason, many other DRL packages, such as RLlib or Tianshou, focus on the scalability of their software frameworks and their learning algorithm implementations but at the cost of being less amicable to the user. The adoption of DRL for serious applications can feel overwhelming as the mathematical and technological background that is required to do so can be imposing. However, due to the current speed and breadth of artificial intelligence research, DRL is achieving milestones that can eventually prove singular for biodiversity conservation. We end with five paths in which DRL can continue to address major issues in SCP.

1. We developed our examples using components that are available out-of-the box in DRL software. For example, the default feature extractors in SB3. These were not developed to process geospatial data so it may be advantageous to build upon network

architectures that have proven useful in remote sensing, such as U-Net (Ronneberger et al., 2015).

2. SCP is evidently subject to conflicting goals. Multi-objective DRL is an active research area (Hayes et al., 2022) that may enable even more realistic applications to SCP.
3. DRL is well-suited for mixing and combining with other methods. This allows the possibility of integrating exact and incomplete algorithms. For example, DRL can be used to design the branching strategy, which is the workhorse of CP methods. Designing a branching strategy is a fundamental and non-trivial issue in CP, as it defines how the space of the problem is explored (Cappart et al., 2020).
4. In this work, we have modelled SCP problems as sequential decision making tasks and solved them using DRL. The DRL agents restore habitat patches sequentially, but in addition to these patch placements, the landscapes are static. In the real world, conservation efforts are embedded in dynamic and uncertain processes such as a changing climate and deforestation. The challenge of overcoming system uncertainty is one of the key promises of DRL, as it arises in many sequential decision-making problems (Lockwood & Si, 2022). The incorporation of recurrent deep learning models into DRL has shown potential to solve partially observable MDPs with greater success than more complicated methods (Ni et al., 2022). Partially observable MDPs not only require good actions but the prediction of states in order to decide on such actions. A first example of this in a SCP setting is Silvestro et al. (2022), in which optimal protected areas are designated in a dynamic and uncertain simulated landscape.
5. There is general consensus that other heuristics, such as evolutionary strategies, do not partake in learning, in the machine learning sense of the word. Learning in DRL provides estimates of the value of individual states and actions, allowing DRL models to plan and come up with surprising strategies to solve problems (Sutton & Barto, 2018). It also arms DRL with the potential to generalise to unforeseen scenarios. This can eventually produce foundation models that can be applied to a wide range of tasks with little retraining, as has happened in computer vision and with large language models. Examples of this in DRL are beginning to be found, such as AdA by DeepMind. Perhaps one day we will have foundation models for conservation planning.

In this paper, we have elaborated a starting point with which we hope to spark interest in the development of DRL applications for SCP. It is one of two examples in this realm. It is known that there are no one-size-fits all solutions to SCP problems and DRL shows grand potential to solve particularly complex and large problems, especially if paired with high performance computing. For this to happen, the usual route other approaches have followed must be paved. We outlined how to specify a SCP problem using an MDP and solve it using DRL as well as how to impose some additional constraints to the problem. However, many other constraints must be devised, as well as other conservation objectives, which boil down to designing the appropriate environments and rewards. Numerous things must still be developed for DRL to address a wider array of SCP problems and will require a whole new body of work. We will continue to work on maturing the approach presented here as well as studying the potential and limitations of DRL for other common SCP problems in static and dynamic, fully observable and uncertain settings. Although DRL opens up the possibility of tackling SCP problems that we thought were intractable with current technologies, it also opens up new concerns. For example, and as happens with any other approach based on deep neural networks, the solutions it offers are difficult or impossible to explain. A very justified concern especially in high-stakes decision making (Rudin, 2019) such as those in conservation. Another point of concern is that the most impressive achievements of DRL have been produced by big tech companies (Meta, Google, Microsoft, Amazon, Baidu, Alibaba, etc.). As discussed in Lapeyrolerie et al. (2022), this shows that the bleeding edge of AI research is in the private sector and the understanding of the capabilities of such technologies is in danger of moving out of grasp of traditional academia. We believe that ecologists and academics in general should be aware of these developments not only to take advantage of their great potential but also to remain vigilant about them.

AUTHOR CONTRIBUTIONS

Julián Equihua, Michael Beckmann and Ralf Seppelt conceived the idea and designed the methodology; Julián Equihua prepared the data, developed the code, ran the experiments and led the writing of the manuscript. All authors contributed critically to the drafts and gave final approval for publication.

ACKNOWLEDGEMENT

Open Access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST STATEMENT

The authors declare that there are no conflicts of interest.

FUNDING INFORMATION

This research was supported by the Deutscher Akademischer Austauschdienst (DAAD) research grant: 91713889.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210X.14300.

DATA AVAILABILITY STATEMENT

All the code and data to train and deploy the DRL models, as well as for the use of restoptr, are provided in the following GitHub repository: https://github.com/jequihua/ccp-drl. This first version of the code has been archived on Zenodo (Equihua, 2024).

ORCID

Julián Equihua https://orcid.org/0009-0006-0072-166X
Ralf Seppelt https://orcid.org/0000-0002-2723-7150

REFERENCES

Alagador, D., & Cerdeira, J. O. (2022). Operations research applicability in spatial conservation planning. *Journal of Environmental Management*, *315*, 115172. https://doi.org/10.1016/j.jenvman.2022.115172

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

Awade, M., Boscolo, D., & Metzger, J. P. (2012). Using binary and probabilistic habitat availability indices derived from graph theory to model bird occurrence in fragmented forests. *Landscape Ecology*, *27*(2), 185–198. https://doi.org/10.1007/s10980-011-9667-2

Bellman, R. (1957). A markovian decision process. *Journal of Mathematics and Mechanics*, *6*, 679–684.

Bello, I., Pham, H., Le, Q. V., Norouzi, M., & Bengio, S. (2017). *Neural combinatorial optimization with reinforcement learning*. http://arxiv.org/abs/1611.09940

Billionnet, A. (2013). Mathematical optimization ideas for biodiversity conservation. *European Journal of Operational Research*, *231*(3), 514–534. https://doi.org/10.1016/j.ejor.2013.03.025

Biosafety Unit. (2023). *Kunming-Montreal global biodiversity framework*. Secretariat of the Convention on Biological Diversity. https://www.cbd.int/gbf/

Bodin, Ö., & Saura, S. (2010). Ranking individual habitat patches as connectivity providers: Integrating network analysis and patch removal experiments. *Ecological Modelling*, *221*(19), 2393–2405. https://doi.org/10.1016/j.ecolmodel.2010.06.017

Cappart, Q., Moisan, T., Rousseau, L.-M., Prémont-Schwarz, I., & Cire, A. *Combining reinforcement learning and constraint programming for combinatorial optimization*. 2020. http://arxiv.org/abs/2006.01610

Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., & Amodei, D. (2017). *Deep reinforcement learning from human preferences*. http://arxiv.org/abs/1706.03741

Clarke, E. M., Klieber, W., Nováček, M., & Zuliani, P. (2012). Model checking and the state explosion problem. In B. Meyer & M. Nordio (Eds.), *Tools for practical software verification: LASER, international Summer School 2011, Elba Island, Italy, revised tutorial lectures, lecture notes in computer science* (pp. 1–30). Springer. https://doi.org/10.1007/978-3-642-35746-6_1

Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de las Casas, D., Donner, C., Fritz, L., Galperti, C., Huber, A., Keeling, J., Tsimpoukelli, M., Kay, J., Merle, A., Moret, J.-M., … Riedmiller, M. (2022). Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, *602*(7897), 414–419. https://doi.org/10.1038/s41586-021-04301-9

Elbrächter, D., Perekrestenko, D., Grohs, P., & Bölcskei, H. (2021). *Deep neural network approximation theory*. http://arxiv.org/abs/1901.02220

Engelhard, S. L., Huijbers, C. M., Stewart-Koster, B., Olds, A. D., Schlacher, T. A., & Connolly, R. M. (2017). Prioritising seascape connectivity in conservation using network analysis. *Journal of Applied Ecology*, *54*(4), 1130–1141. https://doi.org/10.1111/1365-2664.12824

Equihua, J. (2024). *Jequihua/ccp-drl: Initial release*. https://doi.org/10.5281/zenodo.10618900

FAO. (2022). *FRA 2020 remote sensing survey. Number 186 in FAO forestry papers*. FAO. https://doi.org/10.4060/cb9970en

Fawzi, A., Balog, M., Huang, A., Hubert, T., Romera-Paredes, B., Barekatain, M., Novikov, A., Ruiz, F. J. R., Schrittwieser, J., Swirszcz, G., Silver, D., Hassabis, D., & Kohli, P. (2022). Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, *610*(7930), 47–53. https://doi.org/10.1038/s41586-022-05172-4

Francois-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An introduction to deep reinforcement learning. *Foundations and Trends in Machine Learning*, *11*(3–4), 219–354. https://doi.org/10.1561/2200000071

Haj-Ali, A., Ahmed, N. K., Willke, T., Gonzalez, J., Asanovic, K., & Stoica, I. (2019). *A view on deep reinforcement learning in system optimization*. http://arxiv.org/abs/1908.01275

Hamonic, F., Albert, C., Couëtoux, B., & Vaxès, Y. (2023). Optimizing the ecological connectivity of landscapes. *Networks*, *81*(2), 278–293. https://doi.org/10.1002/net.22131

Hanson, J. O., Schuster, R., Morrell, N., Strimas-Mackey, M., Edwards, B. P. M., Watts, M. E., Arcese, P., Bennett, J., & Possingham, H. P. (2023). *prioritizr: Systematic conservation prioritization in R*. https://prioritizr.net, https://github.com/prioritizr/prioritizr.

Hashemi, R., & Darabi, H. (2022). The review of ecological network indicators in graph theory context: 2014–2021. *International Journal of Environmental Research*, *16*(2), 24. https://doi.org/10.1007/s41742-022-00404-x

Hayes, C. F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., Verstraeten, T., Zintgraf, L. M., Dazeley, R., Heintz, F., Howley, E., Irissappane, A. A., Mannion, P., Nowé, A., Ramos, G., Restelli, M., Vamplew, P., & Roijers, D. M. (2022). A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, *36*(1), 26. https://doi.org/10.1007/s10458-022-09552-y

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

Justeau-Allaire, D., Hanson, J. O., Lannuzel, G., Vismara, P., Lorca, X., & Birnbaum, P. (2023). restoptr: An R package for ecological restoration planning. *Restoration Ecology*, *31*, e13910. https://doi.org/10.1111/rec.13910

Justeau-Allaire, D., Vieilledent, G., Rinck, N., Vismara, P., Lorca, X., & Birnbaum, P. (2021). Constrained optimization of landscape indices in conservation planning to support ecological restoration in New Caledonia. *Journal of Applied Ecology*, *58*(4), 744–754. https://doi.org/10.1111/1365-2664.13803

Keeley, A. T. H., Beier, P., Creech, T., Jones, K., Jongman, R. H. G., Stonecipher, G., & Tabor, G. M. (2019). Thirty years of connectivity conservation planning: An assessment of factors influencing plan implementation. *Environmental Research Letters*, *14*(10), 103001. https://doi.org/10.1088/1748-9326/ab3234

Keeley, A. T. H., Beier, P., & Jenness, J. S. (2021). Connectivity metrics for conservation planning and monitoring. *Biological Conservation*, *255*, 109008. https://doi.org/10.1016/j.biocon.2021.109008

Kirkpatrick, J. B. (1983). An iterative method for establishing priorities for the selection of nature reserves: An example from Tasmania. *Biological Conservation*, *25*(2), 127–134. https://doi.org/10.1016/0006-3207(83)90056-3

Kirkpatrick, J. B. (1986). Conservation of plant species, alliances and associations of the treeless high country of Tasmania, Australia. *Biological Conservation*, *37*(1), 43–57. https://doi.org/10.1016/0006-3207(86)90033-9

Krakovna, V., Uesato, J., Mikulik, V., Rahtz, M., Everitt, T., Kumar, R., Kenton, Z., Leike, J., & Legg, S. (2020). *Specification gaming: The flip side of AI ingenuity*. https://www.deepmind.com/blog/specification-gaming-the-flip-side-of-ai-ingenuity

Lapeyrolerie, M., Chapman, M. S., Norman, K. E. A., & Boettiger, C. (2022). Deep reinforcement learning for conservation decisions. *Methods in Ecology and Evolution*, *13*(11), 2649–2662. https://doi.org/10.1111/2041-210X.13954

Li, K., Zhang, T., Wang, R. W. Y., & Han, Y. (2022). Deep reinforcement learning for combinatorial optimization: Covering salesman problems. *IEEE Transactions on Cybernetics*, *52*(12), 13142–13155. https://doi.org/10.1109/TCYB.2021.3103811

Lockwood, O., & Si, M. (2022). *A review of uncertainty for deep reinforcement learning*. http://arxiv.org/abs/2208.09052

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

Margules, C. R., & Pressey, R. L. (2000). Systematic conservation planning. *Nature*, *405*(6783), 243–253. https://doi.org/10.1038/35012251

Martinez Pardo, J., Saura, S., Insaurralde, A., Di Bitetti, M. S., Paviolo, A., & De Angelo, C. (2023). Much more than forest loss: Four decades of habitat connectivity decline for Atlantic Forest jaguars. *Landscape Ecology*, *38*(1), 41–57. https://doi.org/10.1007/s10980-022-01557-y

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

Ni, T., Eysenbach, B., & Salakhutdinov, R. (2022). *Recurrent model-free RL can be a strong baseline for many POMDPs*. http://arxiv.org/abs/2110.05038

Nikishin, E., Izmailov, P., Athiwaratkun, B., Podoprikhin, D., Garipov, T., Shvechikov, P., Vetrov, D., & Wilson, A. G. (2018). *Improving stability in deep reinforcement learning with weight averaging*. Uncertainty in Artificial Intelligence Workshop on Uncertainty in Deep Learning.

OpenAI. (2023). *Introducing ChatGPT*. https://openai.com/blog/chatgpt

Pascual-Hortal, L., & Saura, S. (2006). Comparison and development of new graph-based landscape connectivity indices: Towards the priorization of habitat patches and corridors for conservation. *Landscape Ecology*, *21*(7), 959–967. https://doi.org/10.1007/s10980-006-0013-z

Pereira, M., Segurado, P., & Neves, N. (2011). Using spatial network structure in landscape management and planning: A case study with pond turtles. *Landscape and Urban Planning*, *100*(1), 67–76. https://doi.org/10.1016/j.landurbplan.2010.11.009

Pimm, S. L., Willigan, E., Kolarova, A., & Huang, R. (2021). Reconnecting nature. *Current Biology*, *31*(19), R1159–R1164. https://doi.org/10.1016/j.cub.2021.07.040

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional networks for biomedical image segmentation*. http://arxiv.org/abs/1505.04597

Rubio, L., Bodin, Ö., Brotons, L., & Saura, S. (2015). Connectivity conservation priorities for individual patches evaluated in the present landscape: How durable and effective are they in the long term? *Ecography*, *38*(8), 782–791. https://doi.org/10.1111/ecog.00935

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215. https://doi.org/10.1038/s42256-019-0048-x

Salimans, T., & Chen, R. (2018). *Learning Montezuma's revenge from a single demonstration*. http://arxiv.org/abs/1812.03381

Saura, S., & Pascual-Hortal, L. (2007). A new habitat availability index to integrate connectivity in landscape conservation planning: Comparison with existing indices and application to a case study. *Landscape and Urban Planning*, *83*(2–3), 91–103. https://doi.org/10.1016/j.landurbplan.2007.03.005

Schlaepfer, D. R., Braschler, B., Rusterholz, H.-P., & Baur, B. (2018). Genetic effects of anthropogenic habitat fragmentation on remnant animal and plant populations: A meta-analysis. *Ecosphere*, *9*(10), e02488. https://doi.org/10.1002/ecs2.2488

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal policy optimization algorithms*. http://arxiv.org/abs/1707.06347

Schuster, P. (2000). Taming combinatorial explosion. *Proceedings of the National Academy of Sciences*, *97*(14), 7678–7680. https://doi.org/10.1073/pnas.150237097

Sierra-Altamiranda, A., Charkhgard, H., Eaton, M., Martin, J., Yurek, S., & Udell, B. J. (2020). Spatial conservation planning under uncertainty using modern portfolio theory and Nash bargaining solution. *Ecological Modelling*, *423*, 109016. https://doi.org/10.1016/j.ecolmodel.2020.109016

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

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

Skidmore, A. K., Coops, N. C., Neinavaz, E., Ali, A., Schaepman, M. E., Paganini, M., Kissling, W. D., Vihervaara, P., Darvishzadeh, R., Feilhauer, H., Fernandez, M., Fernández, N., Gorelick, N., Geijzendorffer, I., Heiden, U., Heurich, M., Hobern, D., Holzwarth, S., Muller-Karger, F. E., … Wingate, V. (2021). Priority list of biodiversity metrics to observe from space. *Nature Ecology & Evolution*, *5*(7), 896–906. https://doi.org/10.1038/s41559-021-01451-x

Skidmore, A. K., Pettorelli, N., Coops, N. C., Geller, G. N., Hansen, M., Lucas, R., Mücher, C. A., O'Connor, B., Paganini, M., Pereira, H. M., Schaepman, M. E., Turner, W., Wang, T., & Wegmann, M. (2015). Environmental science: Agree on biodiversity metrics to track from space. *Nature*, *523*(7561), 403–405. https://doi.org/10.1038/523403a

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. *Adaptive computation and machine learning series* (2nd ed.). The MIT Press.

Turchetta, M., Corinzia, L., Sussex, S., Burton, A., Herrera, J., Athanasiadis, I., Buhmann, J. M., & Krause, A. (2022). Learning long-term crop management strategies with CyclesGym. In S. M. Koyejo, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (Vol. *35*, pp. 11396–11409). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2022/file/4a22ceafe2dd6e0d32df1f7c0a69ab68-Paper-Datasets_and_Benchmarks.pdf

UN. (2022). *Global land outlook* (2nd ed.). https://www.unccd.int/resources/global-land-outlook/global-land-outlook-2nd-edition

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

Xue, Y., Wu, X., Morin, D., Dilkina, B., Fuller, A., Royle, J., & Gomes, C. (2017). Dynamic optimization of landscape connectivity embedding spatial-capture-recapture information. *Proceedings of the AAAI Conference on Artificial Intelligence*, *31*(1), 4552–4558. https://doi.org/10.1609/aaai.v31i1.11175

Zheng, S., Trott, A., Srinivasa, S., Parkes, D. C., & Socher, R. (2022). The AI economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances*, *8*(18), eabk2607. https://doi.org/10.1126/sciadv.abk2607

How to cite this article: Equihua, J., Beckmann, M., & Seppelt, R. (2024). Connectivity conservation planning through deep reinforcement learning. *Methods in Ecology and Evolution*, *15*, 779–790. https://doi.org/10.1111/2041-210X.14300