Deep reinforcement learning: a potential new avenue for exploring groundwater management strategies

Philipp Höhn, philipp.hoehn@yahoo.com

This is a preprint of a manuscript currently in preparation. The manuscript is likely to change as it goes through the preparation and a peer review process.

Abstract

To test alternative avenues of control and strategy development in aquifer management, simple examples to couple simulated groundwater environments to deep reinforcement learning frameworks are shown – inspired by recent breakthroughs in other fields. The approach encapsulated an aquifer management task by embodying it in a reward function of a controllable simulated environment, allowing to optimize for total cumulative reward. It is demonstrated that (a) reformulating management objectives in such a way and (b) coupling them to optimization based on reinforcement learning using deep neural networks acting as policy models, can be a novel pathway for computational hydrogeology and can be well worth exploring further. The simulated environments remain available in an open-source repository, thereby supporting free and easy access to run, modify as well as experiment and benchmark with them. Performance benchmarks were provided, currently from repeat control simulation with random actions, human-operated control and control through optimized deep neural networks as policy models. The optimized policy models consistently outperformed random and human-operated control. In the three tested simulated environments, they surpassed the superiority of experienced human control over control from random actions in terms of average cumulative reward by 30.8 %, 25.1 % and 18.4 %, respectively.

1 Introduction

Besides fundamentally understanding and reproducing intelligence in engineered systems, investigating machine performance on tasks associated with human operators or living beings in general is a central part within the field of artificial intelligence (Rajaraman, 2014). Machine learning as a subfield of artificial intelligence deals with improving machine performance through examples or experience. While the definition of (artificial) intelligence remains debated over (Wang, 2019), deep learning (DL) and reinforcement learning (RL) are prominent approaches of machine learning, where DL is associated to improvements from examples and RL from experience. Shen (2018) reviewed deep learning developments across disciplines and summarized the limited number of recent applications in hydrology, including groundwater hydrology. These applications generally exploit the flexibility of artificial neural networks to approximate complex functions, which results from a network of connected units called neurons across layers. In a simplified representation, the network typically passes signals forward while applying linear and non-linear transformations on the signals it received. Similar to parameter search for numerical groundwater models, a multitude of procedures exist to optimize network parameters and procedural parameters, which are cumulatively termed hyperparameters (Bengio, 2012; Hertel, Baldi & Gillen, 2021) and can drastically affect the predictive performance. Early efforts in gradient-based neural network optimization by Rumelhart, Hinton & Williams (1986) date back to when scarce computational resources prohibited a recent DL renaissance owing to its increasing performance (LeCun, Bengio & Hinton, 2015), e.g., from increased computation allowing increases in neural network architecture as well as dataset sizes. The superior performance of DL over traditional approaches in some hydrological applications has been, contrastingly, met with skepticism, enthusiasm and fundamental self-reflection on the future role of hydrology in an age of machine learning (Nearing et al., 2021). Skepticism is partly related to the black box character of the networks, as the flow of information is entirely different from physically-based models, as it is currently mostly uninterpretable, and as it currently remains questionable whether it will ever be similarly interpretable.

Different from performing specific predictive tasks in DL, RL focuses on the development of strategies for sequential decision-making. Sutton & Barto (2018) differentiate between model-based and model-free RL. The difference is the awareness of environmental responses through a model allowing planning, or else unawareness requiring behavioral learning through environmental interactions. RL's quintessential applied target is deriving functions embedding a policy, which maximizes cumulative reward obtained in the environment when taking actions in it. The theories of optimal control, involving planning and dynamic programming, and reinforcement learning share similar concepts and the separation may be blurry, irrespective of distinctly different terminologies being used (Sutton & Barto, 2018). With groundwater running the risk of overexploitation, or with certain aquifer

system states being more desirable in general, need for optimal control has been debated (Gisser & Sánchez, 1980) and investigated for simple cases using differential dynamic programming (Andricevic, 1990; Jones, Willis & Yeh, 1987) – for multiple decades already, since the early advent of numerical computation in hydrogeology. Dynamic programming is highly suited for control problems with low dimensional state and control spaces. With growing state and control spaces, its appeal rapidly decreases from the growing computational complexity, often termed the curse of dimensionality (Castelletti et al., 2010).

When RL and DL are combined, deep reinforcement learning (DRL) uses artificial neural networks as approximators of policy functions in RL contexts. From a subjective perspective, an early DRL milestone has been Mnih et al. (2015) in demonstrating the surpassing of human-level control in a number of Atari video games, rapidly followed by many successful (benchmarking) applications of DRL in other environments as well (a rapidly expanding case list with no claim to completeness: Kober, Bagnell & Peters, 2013; Mnih et al., 2015; Schrittwieser et al., 2020; Silver et al., 2016, 2017, 2018; Vinyals et al., 2019). In a recent and early first study investigating DRL in hydrology, Mullapudi et al. (2020) looked into real-time control of a distributed simulated stormwater system. Using their selected optimization approach, they were successful in controlling an individual stormwater basin, but were confronted with multiple caveats when attempting to control multiple distributed stormwater basins simultaneously. They report a heavy dependence of successful application on the reward formulation. Furthermore, they stress the significant computational burden. This challenge increases further when considering hyperparameter tuning as a subsequent optimization step otherwise widespread and recommended in DL. At the same time, it has an opposingly alleviating perspective with the nowadays increasing availability of distributed computational resources. To date, control of simulated or even real aquifer environments using similar or other RL approaches is yet to be investigated.

When it comes to using optimization procedures in RL for practical applications, a wide variety of approaches is available, many of them are easily accessible as free and open source toolboxes (e.g., as early as RL-Glue: Tanner & White, 2009, Ray RLlib: Liang et al., 2018, rllab: Duan et al., 2016, Stable Baselines: Hill et al., 2018, TF-Agents: Guadarrama et al., 2018, Horizon: Gauci et al., 2019, Bellman: McLeod et al., 2021, ChainerRL: Fujita et al., 2021, MushroomRL: D'Eramo et al., 2021, RLzoo: Ding et al., 2021). Similar to issues of local optima hampering parameter search in groundwater modeling, simulated environments may have deceptive reward structures in DRL, leading the optimization procedure towards a globally suboptimal policy, which subsequent policy updates will not be able to change once converged towards. Approaches additionally operating in the action (i.e., behavioral) space by incentivizing search of novelty (Lehman & Stanley, 2008; Zhang, Yu & Turk, 2019), diversity (Hong et al., 2018; Masood & Doshi-Velez, 2019; Mouret & Clune, 2015), or curiosity (Aljalbout, Ulmer & Triebel, 2021) during exploration, were shown

to help evade converging towards suboptimal policies – for example when using evolutionary optimization (Conti et al., 2018). Such techniques may be worth investigating besides gradient-based methods in future DRL applications in computational hydrogeology. On top of the choice of optimization procedure, a number of factors can affect the total reward obtained, including the choice of hyperparameters, the scale of the environmental rewards and the random seeds used for initialization (Henderson et al., 2017; Hertel, Baldi & Gillen, 2020). This has led to a call for unified and reproducible documentation of RL performance testing (Henderson et al., 2017), which is supported by open access to such toolboxes.

2.1 Scripted simulated groundwater environments coupled to reinforcement learning: a new avenue towards aquifer management optimization?

Groundwater models are frequently at the heart of decision making in groundwater resources management. Exploring suitable groundwater management strategies under the uncertainties of future system states can present overwhelming computational challenges. This is particularly the case when operational decisions (e.g., whether to pump or to artificially recharge, and associated control variables such as pumping or artificial recharge rates) are required in near-real time. With traditional numerical approaches, the evaluation of control variables can entail repeated demands of extensive computational resources each time a decision is required.

Research on artificial intelligence (AI) is generally concerned with the capability of machines to perform tasks that commonly require human operation or are associated with living beings. It also deals with machine capability to take actions within a perceived environment that maximize the probability of achieving a desired objective. AI encompasses a wide range of subfields and methods. These include machine learning (ML), as a means of using examples or experience to improve machine performance. Deep learning (DL) and reinforcement learning (RL) have both become significant subfields of ML. DL focuses on artificial neural networks as models used to perform specific predictive tasks. In contrast, RL deals with strategies for sequential decision-making problems. One way of categorizing RL approaches is into (a) model-based approaches that are aware of how the environment behaves, resulting in the ability to take advantage of this information with planning, and (b) model-free approaches that are unaware of environment behavior and learn from interactions. RL aims to develop policy functions that guide the choice of actions within an environment, in order to maximize cumulative reward returned by the environment (Sutton & Barto, 2018). RL and DL share common ground when artificial neural networks are used in RL sequential decision contexts as policy models in order to approximate such policy functions, which is then termed deep reinforcement learning (DRL). The potential advantage

for operations at a groundwater site is that these policy models can be developed and trained in advance and then quickly queried whenever guidance is needed for decision making, without requiring a great deal of repeated computational effort.

Programming interfaces to numerical groundwater models may have the potential to simplify exploration of reinforcement learning as a promising new tool to guide decision making. Investigating the use of RL avenues in groundwater management has recently been made much easier by the availability of Python programming interfaces for conventional groundwater models (e.g., with FloPy for MODFLOW-based models, or for FEFLOW models). For example, FloPy's wrapper functionality allows efficient processing of input and output files (Bakker et al., 2016) and enables interfacing with other readily-available libraries in the Python ecosystem, including data exchange with well-maintained and rapidly developing libraries for machine learning, such as TensorFlow (Abadi et al., 2016) or PyTorch (Paszke et al., 2019).

To illustrate the possibilities, simple examples of simulated groundwater flow environments were designed and made available in an online repository (FloPyArcade, https://github.com/philipphoehn/flopyarcade). FloPy was used, which allows these examples to be coupled to arbitrary methods of machine learning. Decisions on which actions to take can be controlled in fashion comparable to control in a computer game, by agents such as human operators or policy models. Based on the simulation results, rewards are yielded for success and withheld or removed for failure. An option for manual interaction (i.e., by a human operator) with the environments using strokes on arrow keys in a simple visualization (e.g., as in Figure 1) is included. This can be used to quantify human cumulative reward levels for performance benchmarking or as a simple game for educational purposes to raise awareness concerning groundwater flow.

Simple implementations of two state-of-the-art model-free RL optimization algorithms were included to train (deep) neural networks (using TensorFlow) for their use as policy models. These are (1) the Double Q-learning (van Hasselt, Guez & Silver, 2015) algorithm and (2) a weights-evolving genetic algorithm. The genetic optimization benefits from multiprocessing in parallel search and optional novelty search to help avoid convergence to local policy optima (Conti et al., 2018). Coupling to more standardized implementations of RL algorithms may yield more and earlier success.

The encapsulation of the simulated environments in a game is in reference to early landmark RL success in surpassing human levels in Atari games (Mnih et al., 2015) and to the popularity of games for benchmarking of RL algorithms (see OpenAI Gym: Brockman et al., 2016; http://gym.openai.com/). The simulated environments, rewards, and action spaces can be changed to a higher complexity. RL is simulator-agnostic, but FloPy provides critical ease of modification and customization required when exploring such ways of

alternative model evaluations (Bakker et al., 2016). As an open-source package, FloPy offers the additional benefit of turning RL experiments into transparent, easy to modify and reproducible workflows.

In real-world applications such an approach does not alleviate the need for parameter estimation and uncertainty analyses as key priorities. However, the resulting model (or ensemble of models) can be formulated to allow experimenting with existing or new RL algorithms that have been successfully used in other fields (e.g., Kober, Bagnell & Peters, 2013; Mnih et al., 2015; Schrittwieser et al., 2020). The search for robust neural network policy models might even unveil new and unexpected management strategies.

2.2 Environment descriptions

Here, three custom-made environments (1s-d, 2s-d and 3s-d) are discussed, which mostly differ in their forcing through boundary conditions and in their action space available for control. They share the common objective to minimize the length of the advective flow trajectory of a particle released on the upstream (western) flow boundary within an aquifer, while preventing the particle from being captured in the protected zone surrounding an extraction well or being captured in specific boundaries of the domain.

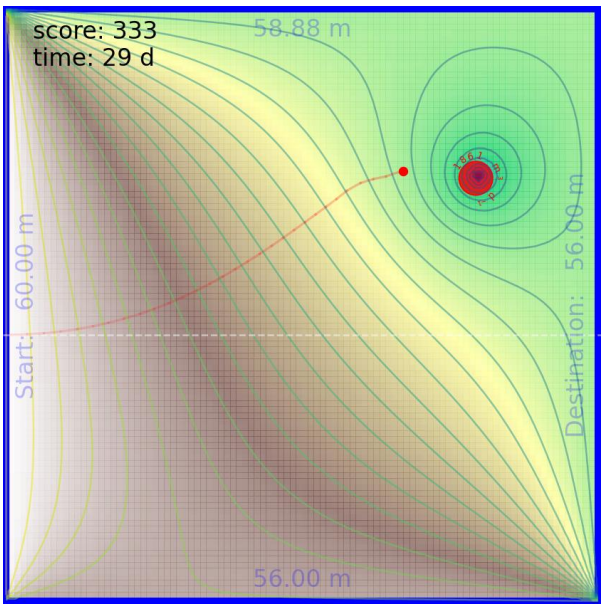

Figure 1: Example visualization of a single intermediate state in the simulated groundwater flow environment 3s-d. The specified hydraulic head boundaries are colored in blue and labeled with their current value. The red particle starts advective travel at the western boundary and is ideally captured in the eastern boundary. Its travel path since its release into the aquifer is colored in light red. A red protection zone surrounds an active extraction well, which is labeled with its current extraction rate. The hydraulic head field is contoured, and cells are colored to indicate its level (on a continuous color gradient from white, brown, yellow and green to lowest values in blue). As the simulation progresses, the current cumulative total reward (score) is reported alongside the current simulation time.

The synthetic aquifer has a single-layer box-like geometry and extends 100 m from the western to the eastern as well as from the northern to the southern boundary – at a thickness of 50 m with the aquifer bottom situated at an elevation of 0 m. The model setup and flow simulation is based on MODFLOW-2005 (Harbaugh, 2005) and particle tracking is based on MODPATH 6 (Pollock, 2012) – both being interfaced with through FloPy. A change in the version of MODFLOW or MODPATH can easily be applied, e.g., with recent advances in FloPy supporting MODFLOW 6 (Langevin et al., 2017) and MODPATH 7 (Pollock, 2016).

Specified hydraulic heads were intended to mimic non-colmated streams exchanging water with the aquifer. In each environment, a specified hydraulic head boundary of 60 m is prescribed along the western boundary, while, similarly, a hydraulic head boundary of 56 m is prescribed along the eastern boundary, forcing a typically westwards-directed flow. In 1s-d and 3s-d, additional specified hydraulic head boundaries extend along the northern and southern boundaries and are initialized using uniformly random values between 56 m and 60 m at the simulation start. In 2s-d, additional specified hydraulic head boundaries only extend along the southern boundary and are initialized using uniformly random values between 56 m and 60 m. In all environments, an extraction well is initially located at uniformly random locations in the aquifer – however, never closer than 50 m to the western boundary or 20 m to the eastern, northern or southern boundary. The extraction rate takes on uniformly random values between 500 m^3 d^{-1} and 2000 m^3 d^{-1}. No groundwater recharge is applied to the simulated environment. The particle release at the western boundary condition takes place

at uniformly random distances, however at minimum distances of 20 m, to the northern and southern boundary.

The groundwater flow situation at simulation start is the steady-state solution resulting from these random initializations. The initialization in turn is controlled by a single random seed. The seed can alternatively be used to reconstruct specific environment simulations, e.g., for validation purposes.

The synthetic aquifer is parameterized with a set of homogeneously distributed hydraulic properties: the horizontal hydraulic conductivity is specified as $1.16 \cdot 10^{-4}$ m s^{-1}, while the vertical hydraulic conductivity is specified at a fraction of it as $1.16 \cdot 10^{-5}$ m s^{-1}. Specific storage is specified as 10^{-5} m^{-1}, while specific yield is specified as 0.15. In this setting, groundwater flow conditions remain predominantly confined. The numerical single-layer domain is spatially discretized in a structured grid with cells of equal horizontal spacing of 10 m. The environment proceeds in time steps spanning 1 d until a maximum time period of 200 d, while a finer time-stepping scheme of 11 steps per environmental time step is used for solving the temporal head field evolution (preconditioned conjugate-gradient solver in MODFLOW-2005). In every environmental time step, the simulation of the groundwater head field is followed by particle tracking to update the trajectory and current location of the single particle.

Ahead of every environmental time step updating the head field and particle location, discrete actions are accepted, allowing for control of the domain at daily intervals in simulation time. 1s-d allows simultaneously controlling the northern and southern specified hydraulic head boundaries. Both boundaries can simultaneously be either (1) kept constant at their current value, (2) increased or (3) decreased by 0.5 m, respectively. Similarly, 2s-d allows controlling the southern specified head boundary by either (1) keeping it constant at its current value, (2) increasing it, or (3) decreasing it by 0.5 m, respectively. In 3s-d, the environmental action space allows controlling the extraction well location by either (1) keeping it at its current location or moving it by 10 m (2) westwards, (3) eastwards, (4) northwards, or (5) southwards. In these simulated groundwater environments, the non-controlled boundary conditions remain constant over time. The levels of the boundaries in 1s-d and 2s-d can be adjusted without limits, except for the limit imposed by the number of adjustments possible during the maximum simulation time of 200 d. In 3s-d, the adjustments to the well location are limited to the aquifer domain, excluding cells with boundary conditions.

The simulation terminates if the particle reaches (1) any of the specified hydraulic head boundaries, (2) a protection zone around the extraction well (at a radius of 2,8 m around the well center), or (3) the simulation period reaches its maximum of 200 d. If the particle has reached the eastern specified hydraulic head boundary, control of the simulation is

considered successful. In all other terminal states, the simulation is considered unsuccessful.

In all discussed examples in FloPyArcade, reward is a function of alignment of the travelled particle trajectory with the straightest path from the current position to the eastern boundary, i.e., most reward is obtained when the particle travel is eastwards along short routes. The operator receives reward in the environment per time step t and accumulates it over the total number of time steps t_n to a cumulative total reward r_{total}. A reward scaling coefficient r_s of 1000 is used, which may be rescaled during optimization procedures, if necessary. In every environmental time step, reward per time step r_t is calculated. It uses the difference in distance of the particle from the eastern boundary at the previous and the current location as a metric of the length of the straightest possible travel path x_s. The reward function then can be described using r_s, the ratio between x_s and the length of the actually travelled path x_a, the distance between eastern and western specified hydraulic head boundary x_{max} of 98 m, coefficient d1, d2 and d3, as well as penalty coefficients c_{p1} and c_{p2}:

$$r_{total} = \sum_{t=1}^{t_n} r_t = \sum_{t=1}^{t_n} d\left(x_{s_t} x_{max}^{-1} r_s \left(x_{s_t} x_{a_t}^{-1}\right)^{c_{p1}}\right)^{c_{p2}} \quad \text{(Equation 4.1)}$$

$$\text{with } d = 1, c_{p1} = 10 \text{ and } c_{p2} = 1 \text{ if } x_{s_t} \geq 0$$

$$\text{and } d = 5, c_{p1} = 0 \text{ and } c_{p2} = 1 \text{ if } x_{s_t} < 0$$

```python
from flopyarcade import FloPyEnv
from numpy.random import choice

env = FloPyEnv(ENVTYPE='3s-d')
reward_total = 0.
while not env.done:
  action = choice(env.actionSpace)
  observations, reward, done, info = env.step(action)
  reward_total += reward
```

Figure 2: Minimal code example in the Python programming language to set up and run a randomly initialized instance of the simulated groundwater flow environment 3s-d by passing random actions from the available action space.

With eastward particle travel, x_a and x_s have positive sign, while they have negative sign with westward travel. The simulation starts with r_{total} of 0. If reaching termination with successful control, r_{total} is unchanged, while in the case of unsuccessful control it is reset to 0 if positive total reward was accumulated. The reward function allows negative rewards and negative cumulative total reward, while scores are commonly expected to range between 0 and 1000. For optimization, the target is to maximize r_{total} from travel behavior in the environment.

The choice and formulation of the reward function is of significant importance for guiding optimization in deep reinforcement learning (e.g., Mullapudi et al., 2020; Ng, Harada & Russell, 1999). Modification of the reward function for experimentation is enabled through the openly accessible source code of FloPyArcade.

Figure 2 provides a minimal example of the programming-based initialization, simulation and control of a groundwater environment in FloPyArcade. After every simulation step, the

environment returns state observations in a vector of normalized values, reward from the simulated time step as a decimal number, the current state of the termination criterion and optional information – similar to the environment encapsulation by OpenAI's toolkit Gym. Actions can similarly be easily passed from various sources enacting control, such as policy models. The returned normalized observations pertain to the current simulation state and are a vectorized collection comprised of the hydraulic head values at every third grid cell, the values at the specified hydraulic head boundary conditions, the coordinates as well as the pumping rate of the extraction well and the coordinates of the current particle location.

More environments have meanwhile been added to the FloPyArcade repository. However, their discussion is omitted, as results dealing with their control are not presented here. While similar, they mostly differ in their type of forcing on the system, their level of complexity in the action space available for control and the availability of continuous-valued actions.

2.3 Performance benchmarks

In DRL, it is common to compare average performances across a number of random initializations between control operators. To allow the same for the discussed simulated environments in FloPyArcade, such average performances of multiple operators are reported in Figure 3 as benchmarks.

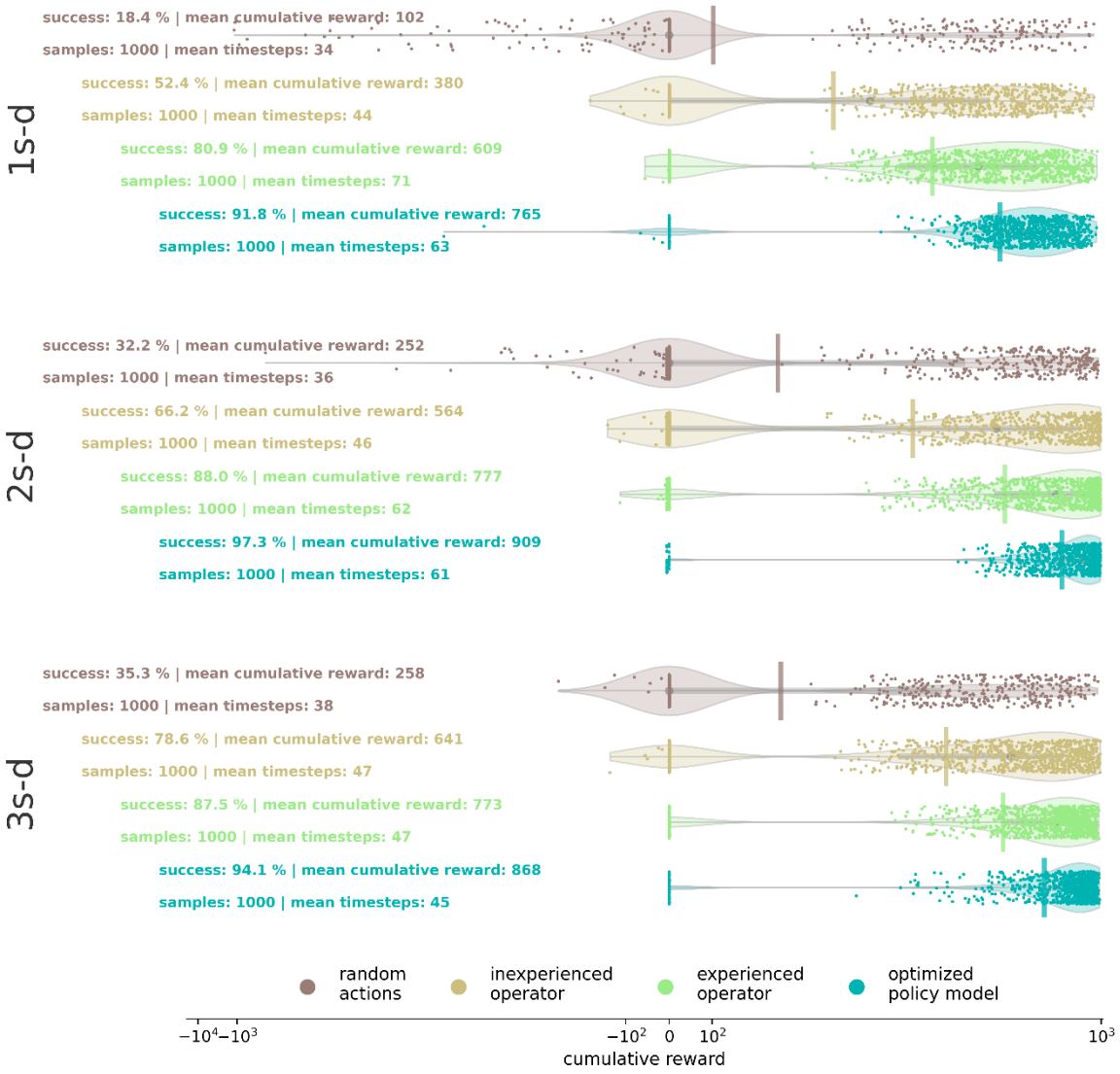

Figure 3: Benchmarks of performance of control from (1) random actions, an (2) inexperienced and (3) experienced human operator, as well as (4) an example of a trained deep neural network policy model in the three simulated FloPyArcade environments 1s-d, 2s-d and 3s-d. The mean cumulative reward is reported as a major benchmark in text on the left, together with metrics on the respective success rate, number of samples for averaging and mean of the environmental time steps leading to the reported data. The mean cumulative reward is additionally displayed as a vertical bar crossing the violin plots, which visualize the distribution of the cumulative rewards (jittered data).

Four benchmarks were obtained per simulated environment: mean total reward obtained from (a) random actions, an (b) inexperienced and (c) experienced human operation, as well as from (d) a trained policy model encoded in a deep neural network.

The benchmarks from random actions were included to form a lower baseline for performance. The inexperienced human operator had no prior insights into the development of the simulated environments and no education about groundwater flow. Beforehand, the inexperienced human operator was merely informed about the general setup of the simulated experiment and the available action space for control. However, the inexperienced human operator was not informed about specifics of the mathematical reward formulation or potentially occurring groundwater flow situations. In contrast, the experienced operator had insight into the development of the simulated environments and background in hydrogeology to allow for a certain degree of prior intuition about groundwater flow.

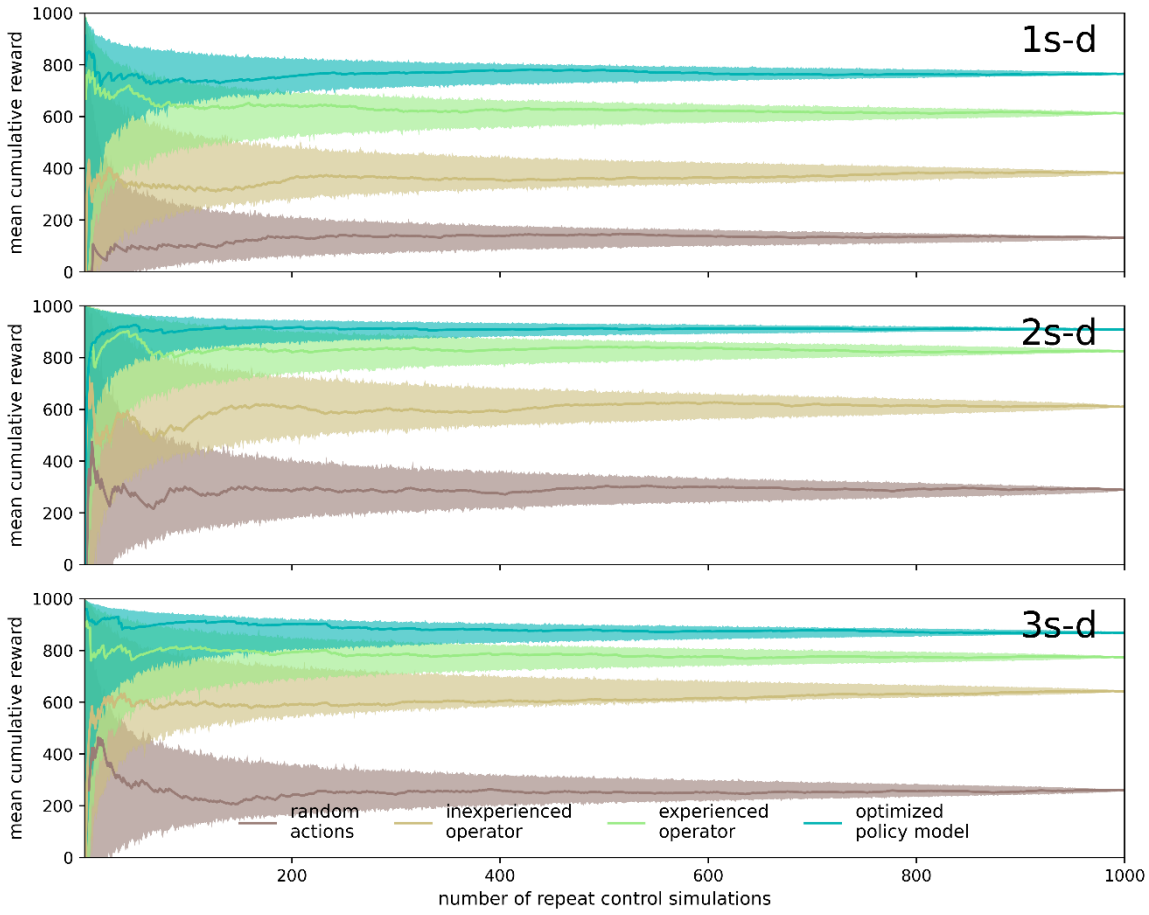

Figure 4: Evolution of the mean cumulative reward benchmark per control operation as a function of the number of repeat control simulations considered for averaging, depicted for each simulated groundwater environments 1s-d, 2s-d and 3s-d. Similar-colored shading indicates the highest and lowest possible mean cumulative reward from different sample ordering, in order to indicate the number of repetitions required to differentiate mean cumulative reward of the benchmarked control operations.

Furthermore, given the returned observations and obtained rewards, a deep neural network was optimized (i.e., trained) in a reinforcement learning framework for each environment. For the presented benchmarks, training was performed using an openly available RL implementation (Ray RLlib: Liang et al., 2018) of Double Q-learning (Double DQN, for a more detailed description see van Hasselt, Guez & Silver, 2015), an off-policy reinforcement learning algorithm, alongside distributed prioritized experience replay (Ape-X: Horgan et al., 2018). Ape-X allows scaling across distributed computing infrastructure and has been shown to foster promising performance metrics in other contexts. Fully-connected feed-forward neural networks (using the openly available implementation through TensorFlow: Abadi et al., 2016) were chosen as policy models. Their architecture was comparably simple in that it contained 2 consecutive hidden layers of 256 units each, using the common hyperbolic tangent activation function.

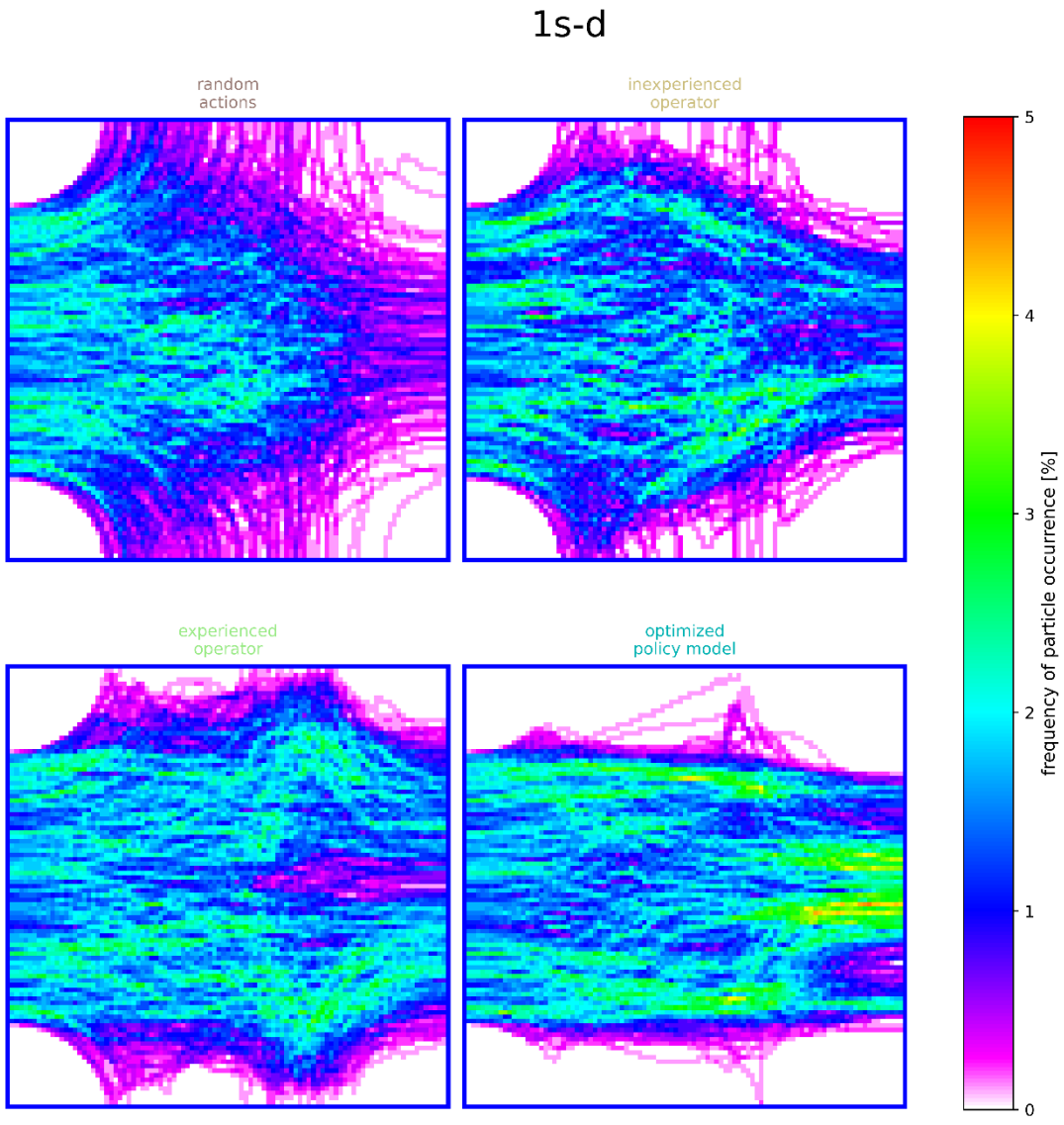

Figure 5: Heatmap indicating how frequently grid cells in the aquifer domain were visited by the travelling particle across the collected control simulations in environment 1s-d.

The Ape-X DQN optimization was configured to start after an initial collection of data from $5 \cdot 10^4$ environmental time steps and to store data of up to $3 \cdot 10^6$ timesteps in its shared circular experience replay memory. Distributed workers were configured to update the shared memory every 50 timesteps, to collect experiences with epsilon-greedy exploration using a discount factor of 0.99, while the optimization updated its model for action evaluation every 10^4 timesteps and its target model for action selection every 10^5 timesteps, using batches of size 512 for stochastic gradient descent at a learning rate of $5 \cdot 10^{-5}$. Optimization was run until at least $15 \cdot 10^6$ timesteps were simulated and checkpoints were used to select the policy model yielding highest cumulative reward during training for evaluation. In evaluation, this policy model was used to revisit and control the same initializations of the environments as the other operators to come up with an average cumulative reward benchmark.

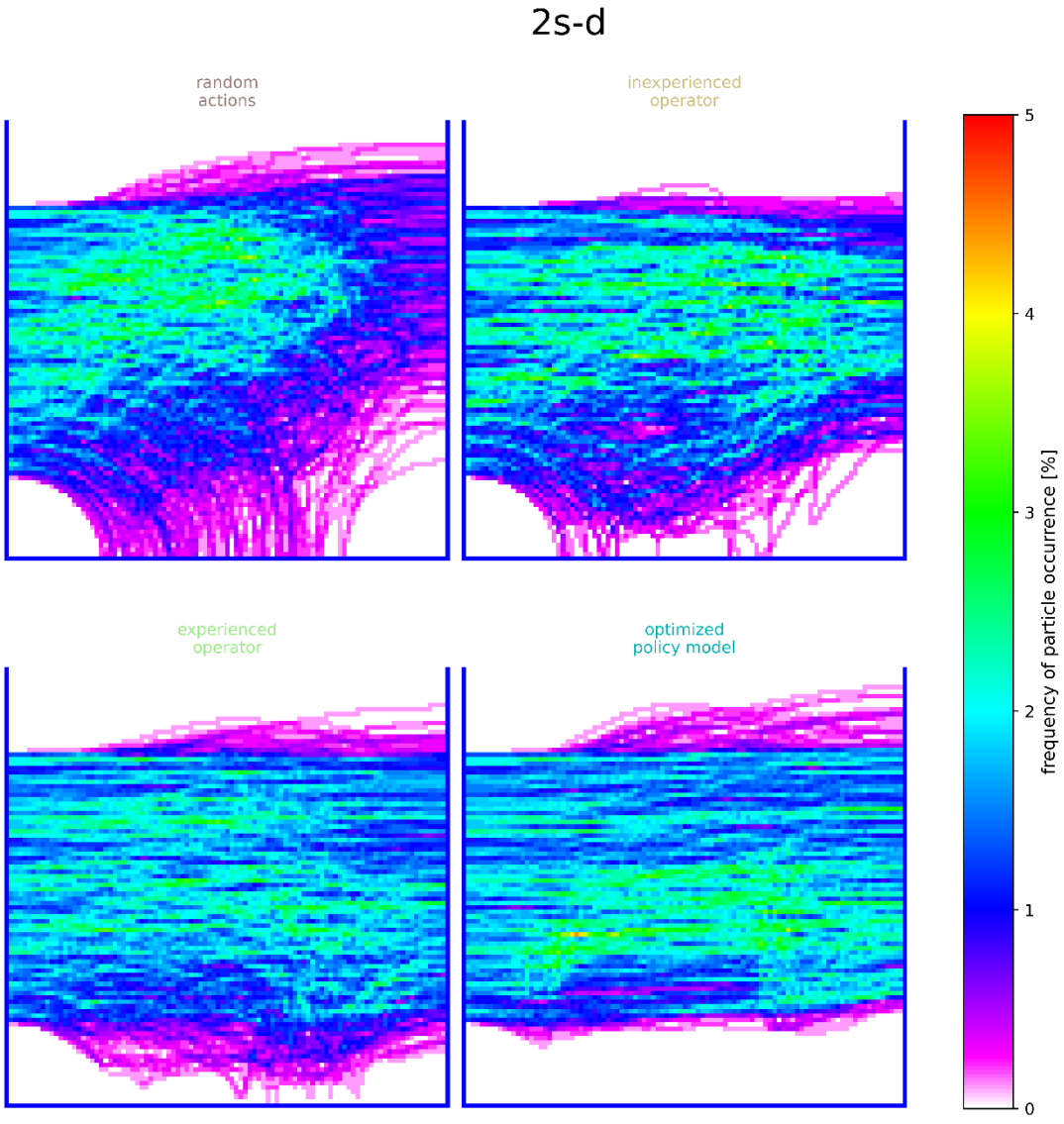

Figure 6: Heatmap indicating how frequently grid cells in the aquifer domain were visited by the travelling particle across the collected control simulations in environment 2s-d.

For further implementation details, a reproducible workflow for the optimization as well as the trained policy models are made available in the FloPyArcade online repository. Note that slight deviations when reproducing may be possible from variations in the seed used in the initialization of the neural network parameters and the simulated environments during optimization. The discussed deep neural networks are considered to provide example benchmarks for policy models here. Depending on the specifications (i.e., hyperparameters) of the optimization procedure, different average cumulative rewards are expected. Subsequent improvements to their performance, outdating the reported metrics in Figure 4 may be reported in the online repository at a later point in time. Hyperparameter tuning is more commonplace in DL, remaining extremely computationally demanding in current DRL practice.

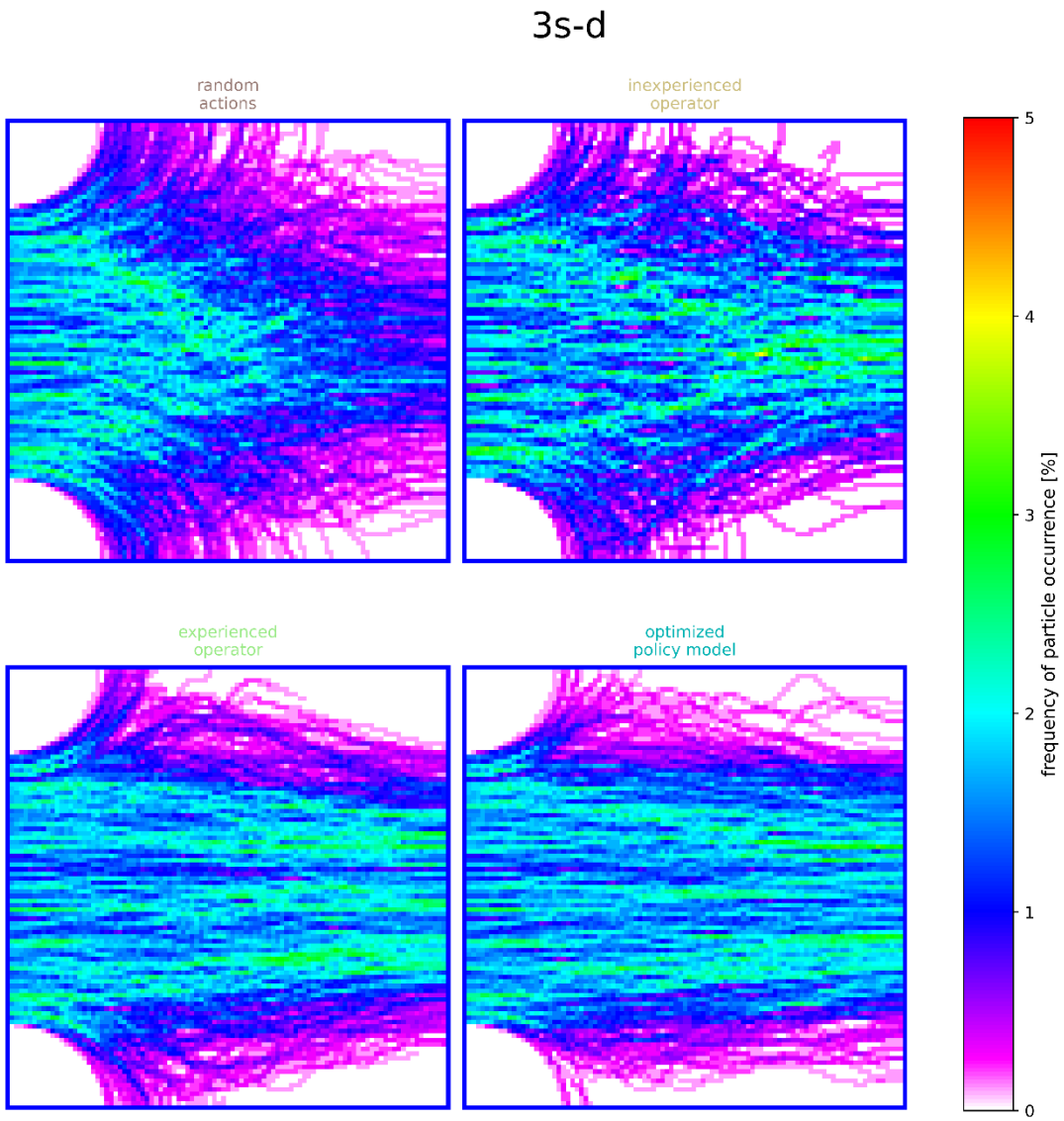

Figure 7: Heatmap indicating how frequently grid cells in the aquifer domain were visited by the travelling particle across the collected control simulations in environment 3s-d.

Following recommendations for reproducible DRL (Henderson et al., 2017), the presented results were obtained in environments, which were initialized with the same set of random seeds for all four operators. The number of repeat control simulations to obtain the benchmarks (Figure 3) per environment were 1000 for all four operators.

As Figure 3 and Figure 4 show, taking random actions consistently yielded the lowest benchmark in terms of average cumulative rewards, which serves as a reproducible lower baseline. The inexperienced human operator consistently ranked between the random baseline and the baseline of the experienced human operator. Strikingly, the deep neural network as a policy model consistently and significantly outperformed the experienced human operator (example visualization in Figure 8) – both in terms of the average cumulative reward and the success rate (Figure 3). When comparing the superiority of the optimized policy models to the random baseline with superiority of experienced human operation to the random baseline in terms of average cumulative reward, the optimized policy model outperformed the experienced human control by 30.8 % in environment 1s-d, by 25.1 % in environment 2s-d and by 18.4 % in environment 3s-d.

Figure 8: Example (generated using seed 347) from the benchmarking simulations, resemblant of the average differences in performance between controlling agents. The continuous colors for the hydraulic head field, the particle trajectory throughout the simulation in light red and other visualizations are consistent with the general description given for Figure 1.

This means that in these simulated environments, on average, the policy encoded in the respective deep neural network leads to a higher likelihood of successful control than the control from the experienced human operator.

A number of questions, such as (1) the performance and burden of optimization in more complex simulated environments or (2) the policy model behavior and associated risk in edge cases, may arise from these results and provide further research questions. However, the results suggest that, for simple groundwater management tasks, optimizing a neural network policy model could be a new avenue in support of real-time site operation and could act superior in control when compared to hand-crafted or manual schemes. On top of that, in the presented simulated environments, actions can be queried per timestep within a few

milliseconds as opposed to repeated computational burden when applying other types of strategy evaluation.

The different levels of performance shown in the benchmarks are similarly reflected when looking at the heatmaps of particle travel through the aquifer domain in the simulated environment 1s-d (Figure 5), 2s-d (Figure 6) and 3s-d (Figure 7), respectively. They reflect differences in control behavior between operators and reflect more streamlined and consistent travel trajectories reaching the target boundary condition more regularly when controlled by the respective policy model.

3 Summary

Computational hydrogeology supporting aquifer control in near-real time through simulation is challenged by the runtime required when in real-time operation, easily requiring distributed compute resources (Kurtz et al., 2017). Recent successes in system control using deep reinforcement learning in other fields inspired formulating simple aquifer management tasks in simulated groundwater environments, which expose state observations together with environmental rewards based on the actions taken.

The simulated groundwater environments operate in conjunction with FloPy, an open-source library written in the Python programming language to interface with MODFLOW-based groundwater models. The growing number of libraries enabling and simplifying deep reinforcement learning in Python made FloPy seem an obvious choice for connecting groundwater simulation to control algorithms to experiment and explore new avenues.

It was hypothesized and demonstrated that reformulating aquifer management tasks through groundwater simulation in a reinforcement learning framework can assist to explore control strategies in new ways. The developed open-source resource (https://github.com/philipphoehn/flopyarcade) includes simple simulated groundwater environments. It supports free and easy access to run the simulated environments and to couple arbitrary control strategies. It provides performance benchmarks, currently from repeated random, human-operated or machine-learned control of the simulated systems.

4 Outlook

As an interdisciplinary tool, machine learning, which includes deep learning, has seen comparably slow yet widespread adoption throughout geoscience (Dramsch, 2020). Sit et al. (2020) provide a comprehensive and recent review of deep learning throughout hydrology and water resources. In their compilation, they quantify, on the basis of publication records, how both fields have generally started embracing deep learning for tasks involving sequence prediction, classification and regression, while at the same time showing little to no applications of deep reinforcement learning (DRL), yet. They only reported on a single study investigating DRL for control of a membrane bioreactor plant with economic and environmental optimization objectives (Nam et al., 2020). Sit et al. (2020) mentioned that a potential use case could be the operation of dam release control structures. In a sense, a related application was published only a mere three months later, investigating DRL for real-time control of stormwater systems (Mullapudi et al., 2020). A few more studies related to or testing DRL were conducted. An early study still using shallow (i.e., single hidden layer) neural networks in reinforcement learning for control of water systems dates back to 2003 (Bhattacharya, Lobbrecht & Solomatine, 2003). It seems as if, while general interest in using reinforcement learning for control of water reservoir systems remained (e.g., Castelletti et al., 2010; Delipetrev, Jonoski & Solomatine, 2017; Hooshyar et al., 2020), interest in reinforcement learning leveraging neural networks had been lost meanwhile and only been renewed recently (on pump operation in water distribution systems: Hajgató, Paál & Gyires-Tóth, 2020; on urban drainage control: Mullapudi et al., 2020).

Given the tremendous successes in other fields to train DRL systems for control of various kinds of tasks (e.g., Li, 2018), it appears reasonable to expect breakthroughs in aquifer management as well. Promising applications in other fields are quite diverse. While some systems learn from experience gathered in real-world interactions (e.g., in robotics: Mahmood et al., 2018; Riedmiller et al., 2018), many systems share common ground in that they employ some form of a simulator of the system dynamics to process control interactions of the agents with the simulated environment. A subjective selection of promising DRL applications in other fields, is: DRL agents controlling high speed robots (Kober, Bagnell & Peters, 2013), controlling video games near or above human-level performance (Mnih et al., 2015), mastering the game of Go without learning from recorded human experiences (Silver et al., 2017), mastering chess, shogi and Go through self-play (Silver et al., 2018), or even matching the overall skill of top players in the highly complex real-time strategy game of StarCraft II (Vinyals et al., 2019).

As Mullapudi et al. (2020) mentioned, DRL has yet to be applied to the real-time control of an urban drainage system. The same applies to hydrogeological systems. Such control

approaches are expected to (a) generally perform better when benchmarked on critical operational metrics. At the same time, DRL, particularly when designed to allow significant levels of exploration, might (b) even assist the development of unexpected high-reward alternative control strategies, which would seem new or even counterintuitive to human controllers (Chrabaszcz, Loshchilov & Hutter, 2018; Lehman, Clune & Misevic, 2020).

Hydrogeology as a field currently appears to be lagging well behind experimenting with and adopting computational approaches from the forefront of machine learning research. In that sense, computational hydrogeology has every chance of related discovery and merit for the field through mere adoption, following and branching off of related efforts in other fields. Machine learning, including DRL, has the tendency to benefit from increases in parallel computational resources. Vice versa, limited resources can be a limiting factor. However, academic experimentation with elevated parallel computational power afforded by industrial high performance computing resources has already started to be leveraged in hydrogeology a number of years ago (e.g., Hammond & Lichtner, 2010; Hayley, 2017; Kurtz et al., 2017; Lapin et al., 2014). Slightly surprisingly, despite these experiences, DRL has yet to be and is anticipated to be applied for control of managing aquifer systems.

Management of groundwater can typically be assisted or guided using numerical physically-based groundwater models. Supporting management with forward simulations in real-time using these models easily requires significant computational resources and can render its application infeasible – more so, if ensembles for uncertainty quantification are employed. The idea with DRL is that deep neural networks can be trained ahead of real-time application as policy models to provide control recommendations with a quick and computationally inexpensive forward network pass. This is expected to pave a new avenue in simulated aquifer management.

To get started with DRL in computational hydrogeology, it would be reasonable to (a) first get started with showcasing DRL agents solving very simple and easy-to-grasp simulated environments. FloPyArcade (https://github.com/philipphoehn/flopyarcade) is meant to enable creating simple showcases of successfully trained policy models. A favorable aspect is that the DRL framework is very generic in nature and would allow replacement of the simulated groundwater system with systems of arbitrary architecture, conceptualization, complexity, controls, objectives and reward feedback systems. It would seem immensely helpful for the groundwater community to (b) establish a common set of benchmarking environments to provide baselines for experimentation. Similar collections of benchmarking problems already exist in other contexts (e.g., OpenAI Gym: Brockman et al., 2016, DeepMind Lab: Beattie et al., 2016, AI Safety Gridworlds: Leike et al., 2017, CARLA: Dosovitskiy et al., 2017, PySC2: Vinyals et al., 2017, DeepMind Control Suite: Tassa et al., 2018, Behaviour Suite: Osband et al., 2019). Such collections could allow operational

decisions to be reflected in discrete control variables, in control variables ranging on a continuous value spectrum or in a mixture of both. It would seem valuable to include environments with control tasks similar to typical management problems practitioners are faced with. Just to name a few, ideas could be environments mimicking control of pumping operation in contaminant plume containment, dewatering in groundwater level control or managed aquifer storage and recovery.

Objectives optimized towards in DRL contexts must be numerically expressed. However, environments can be envisioned to be associated with single objectives or multiple simultaneously weighted objectives. In a weighted objective function, among many other potential joint objectives, merely to give an example, groundwater quality requirements can be combined with ecological and economic objectives.

Control using real-time simulated policy evaluation (e.g., Bauser et al., 2010; Hendricks Franssen et al., 2012) can function without prior optimization of control policies. However, in applied cases they easily suffer from the curse of dimensionality leading to overwhelming computational requirements – whenever requiring control decisions repeatedly. A driving motivation behind training a policy model ahead of its use in real-time control is to drastically decrease on-demand computational requirements during queries of optimal control actions. Although a number of its algorithms have supervised learning components (i.e., training on a buffer of stored experiences, e.g., Lin, 1993; Mnih et al., 2015), reinforcement learning is different from supervised learning in that it improves its behavior from evaluative feedback given by repeated simulated experience (Littman, 2015). As already mentioned, the feedback collection for training can still lead to substantial or overwhelming computational burden. To decrease such demand, another promising avenue could be the exploration of surrogate reduced-order modeling to replace the simulated system dynamics. When surrogate models are used to replace the underlying simulated system with a reduced-order model, computation time can be decreased at the cost of, ideally minimal, approximation error (Peitz & Dellnitz, 2018). It seems a viable path to explore surrogate modeling in parallel to exploration of reinforcement learning for aquifer control problems. Notably, surrogate modeling is far from new, also not in computational hydrogeology (Asher et al., 2015; Cousquer et al., 2018; Luo & Lu, 2014; Xing et al., 2019; Zhao, Lu & Xiao, 2016). A number of approaches exist (reviewed for hydrogeology in Asher et al., 2015). New deep learning approaches are recently being studied to develop surrogate models driven by simulation data (e.g., Lu et al., 2021; Pfaff et al., 2020; Sanchez-Gonzalez et al., 2020). Pfaff et al. (2020) demonstrated a 1-2 order of magnitude computation speed-up compared to selected simulations of various physical systems when using graph neural networks.

Although not covering recent advances, Razavi, Tolson & Burn (2012) stress that surrogate models may be computationally demanding to design and optimize, have subjective and

non-intuitive structure and require validation strategies to avoid overfitting. What needs further consideration is the propagation of approximation errors with long simulation times. When the surrogate model requires its own results as input for the next time step (Jatnieks et al., 2016), this might lead to, depending on the simulated environment, a limit on the number of time steps to approximate with a surrogate model. Essentially, considering application of surrogate models together with DRL requires a delineation between opportunities and limitations, as their development is neither error-free nor effort-free. New and promising efforts to develop deep learning based surrogate models will currently be experimental. It should be evaluated, first, if and under which circumstances surrogate models could be accurate enough, and secondly, if and under which circumstances their development would have a beneficial development to saved runtime. Considering the potential benefits, exploration of surrogate modeling in the context of deep reinforcement learning certainly seems well worth the effort to build an experience base for the groundwater community.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I. J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., … Zheng, X. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, *abs/1603.0*. http://arxiv.org/abs/1603.04467

Aljalbout, E., Ulmer, M. & Triebel, R. (2021). *Seeking visual discomfort: Curiosity-driven representations for reinforcement learning*. http://arxiv.org/abs/2110.00784

Asher, M. J., Croke, B. F. W., Jakeman, A. J. & Peeters, L. J. M. (2015). A review of surrogate models and their application to groundwater modeling. *Water Resources Research*, *51*(8), 5957–5973. https://doi.org/10.1002/2015WR016967

Bakker, M., Post, V., Langevin, C. D., Hughes, J. D., White, J. T., Starn, J. J. & Fienen, M. N. (2016). Scripting MODFLOW model development using Python and FloPy. *Groundwater*, *54*(5), 733–739. https://doi.org/10.1111/gwat.12413

Bauser, G., Hendricks Franssen, H. J., Kaiser, H. P., Kuhlmann, U., Stauffer, F. & Kinzelbach, W. (2010). Real-time management of an urban groundwater well field threatened by pollution. *Environmental Science and Technology*, *44*(17), 6802–6807. https://doi.org/10.1021/es100648j

Beattie, C., Leibo, J. Z., Teplyashin, D., Ward, T., Wainwright, M., Küttler, H., Lefrancq, A., Green, S., Valdés, V., Sadik, A., Schrittwieser, J., Anderson, K., York, S., Cant, M., Cain, A., Bolton, A., Gaffney, S., … Petersen, S. (2016). *DeepMind Lab*. 1–11. http://arxiv.org/abs/1612.03801

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *Lecture Notes in Computer Science*, 437–478. https://arxiv.org/abs/1206.5533v2

Bhattacharya, B., Lobbrecht, A. H. & Solomatine, D. P. (2003). Neural networks and reinforcement learning in control of water systems. *Journal of Water Resources Planning and Management*, *129*(6), 458–465. https://doi.org/10.1061/(asce)0733-9496(2003)129:6(458)

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J. & Zaremba, W. (2016). *OpenAI Gym*. 1–4. http://arxiv.org/abs/1606.01540

Castelletti, A., Galelli, S., Restelli, M. & Soncini-Sessa, R. (2010). Tree-based reinforcement learning for optimal water reservoir operation. *Water Resources Research*, *46*(9), 1–19. https://doi.org/10.1029/2009WR008898

Chrabaszcz, P., Loshchilov, I. & Hutter, F. (2018). Back to basics: Benchmarking canonical

evolution strategies for playing atari. *IJCAI International Joint Conference on Artificial Intelligence*, *2018-July*, 1419–1426. https://doi.org/10.24963/ijcai.2018/197

Conti, E., Madhavan, V., Such, F. P., Lehman, J., Stanley, K. O. & Clune, J. (2018). Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *Advances in Neural Information Processing Systems*, *2018-Decem*(NeurIPS), 5027–5038.

Cousquer, Y., Pryet, A., Atteia, O., Ferré, T. P. A., Delbart, C., Valois, R. & Dupuy, A. (2018). Developing a particle tracking surrogate model to improve inversion of ground water – Surface water models. *Journal of Hydrology*, *558*, 356–365. https://doi.org/10.1016/j.jhydrol.2018.01.043

D'Eramo, C., Tateo, D., Bonarini, A., Restelli, M. & Peters, J. (2021). MushroomRL: Simplifying reinforcement learning research. *Journal of Machine Learning Research*, *22*, 1–5.

Delipetrev, B., Jonoski, A. & Solomatine, D. P. (2017). A novel nested stochastic dynamic programming (nSDP) and nested reinforcement learning (nRL) algorithm for multipurpose reservoir optimization. *Journal of Hydroinformatics*, *19*(1), 47–61. https://doi.org/10.2166/hydro.2016.243

Ding, Z., Yu, T., Zhang, H., Huang, Y., Li, G., Guo, Q., Mai, L. & Dong, H. (2021). *Efficient reinforcement learning development with RLzoo*. 3759–3762. https://doi.org/10.1145/3474085.3478323

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A. & Koltun, V. (2017). *CARLA: An open urban driving simulator*. *CoRL*, 1–16. http://arxiv.org/abs/1711.03938

Dramsch, J. S. (2020). 70 years of machine learning in geoscience in review. *Advances in Geophysics*, *61*, 1–55. https://doi.org/10.1016/bs.agph.2020.08.002

Duan, Y., Chen, X., Houthooft, R., Schulman, J. & Abbeel, P. (2016). Benchmarking deep reinforcement learning for continuous control. *33rd International Conference on Machine Learning, ICML 2016*, *3*, 2001–2014.

Fujita, Y., Nagarajan, P., Kataoka, T. & Ishikawa, T. (2021). ChainerRL: A deep reinforcement learning library. *Journal of Machine Learning Research*, *22*, 1–14.

Gauci, J., Conti, E., Liang, Y., Virochsiri, K., He, Y., Kaden, Z., Narayanan, V., Ye, X., Chen, Z. & Fujimoto, S. (2019). *Horizon: Facebook's open source applied reinforcement learning platform*. http://arxiv.org/abs/1811.00260

Gisser, M. & Sánchez, D. A. (1980). Competition versus optimal control in groundwater pumping. *Water Resources Research*, *16*(4), 638–642. https://doi.org/10.1029/WR016i004p00638

Guadarrama, S., Korattikara, A., Ramirez, O., Castro, P., Holly, E., Fishman, S., Wang, K., Gonina, E., Wu, N., Kokiopoulou, E., Sbaiz, L., Smith, J., Bartók, G., Berent, J., Harris, C., Vanhoucke, V. & Brevdo, E. (2018). *TF-Agents: A library for reinforcement learning in TensorFlow*. https://github.com/tensorflow/agents

Hajgató, G., Paál, G. & Gyires-Tóth, B. (2020). Deep reinforcement learning for real-time optimization of pumps in water distribution systems. *Journal of Water Resources Planning and Management*, *146*(11), 04020079. https://doi.org/10.1061/(asce)wr.1943-5452.0001287

Harbaugh, A. W. (2005). MODFLOW-2005 , the U.S . Geological Survey modular ground-water model—the ground-water flow process. *U.S. Geological Survey Techniques and Methods 6–A16*.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D. & Meger, D. (2017). Deep reinforcement learning that matters. *ArXiv*, 3207–3214. http://arxiv.org/abs/1709.06560

Hendricks Franssen, H. J., Bauser, G., Stauffer, F., Kaiser, H. P., Kuhlmann, U. & Kinzelbach, W. (2012). A comparison study of two different control criteria for the real-time management of urban groundwater works. *Journal of Environmental Management*, *105*, 21–29. https://doi.org/10.1016/j.jenvman.2011.12.024

Hertel, L., Baldi, P. & Gillen, D. L. (2020). *Quantity vs. quality: On hyperparameter optimization for deep reinforcement learning*. http://arxiv.org/abs/2007.14604

Hertel, L., Baldi, P. & Gillen, D. L. (2021). Reproducible hyperparameter optimization. *Journal of Computational and Graphical Statistics*, *0*(0), 1–39. https://doi.org/10.1080/10618600.2021.1950004

Hill, A., Raffin, A., Ernestus, M., Gleave, A., Kanervisto, A., Traore, R., Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S. & Wu, Y. (2018). Stable Baselines. In *GitHub repository*. GitHub. https://github.com/hill-a/stable-baselines

Hong, Z. W., Shann, T. Y., Su, S. Y., Chang, Y. H., Fu, T. J. & Lee, C. Y. (2018). Diversity-driven exploration strategy for deep reinforcement learning. *Advances in Neural Information Processing Systems*, *2018-Decem*(Nips), 10489–10500.

Hooshyar, M., Mousavi, S. J., Mahootchi, M. & Ponnambalam, K. (2020). Aggregation-decomposition-based multi-agent reinforcement learning for multi-reservoir operations optimization. *Water*, *12*(10), 2688. https://doi.org/10.3390/w12102688

Horgan, D., Quan, J., Budden, D., Barth-Maron, G., Hessel, M., Van Hasselt, H. & Silver, D. (2018). Distributed prioritized experience replay. *6th International Conference on*

Learning Representations, ICLR 2018 - Conference Track Proceedings, 1–19.

Jones, L. D., Willis, R. & Yeh, W. W. -G. (1987). Optimal control of nonlinear groundwater hydraulics using differential dynamic programming. *Water Resources Research*, *23*(11), 2097–2106. https://doi.org/10.1029/WR023i011p02097

Kober, J., Bagnell, J. A. & Peters, J. (2013). Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*, *32*(11), 1238–1274. https://doi.org/10.1177/0278364913495721

Kurtz, W., Lapin, A., Schilling, O. S., Tang, Q., Schiller, E., Braun, T., Hunkeler, D., Vereecken, H., Sudicky, E., Kropf, P., Hendricks Franssen, H. J. & Brunner, P. (2017). Integrating hydrological modelling, data assimilation and cloud computing for real-time management of water resources. *Environmental Modelling and Software*, *93*, 418–435. https://doi.org/10.1016/j.envsoft.2017.03.011

Langevin, C. D., Hughes, J. D., Banta, E. R., Niswonger, R. G., Panday, S. & Provost, A. M. (2017). Documentation for the MODFLOW 6 groundwater flow model. *U.S. Geological Survey Techniques and Methods, Book 6, Chap. A55*. https://pubs.usgs.gov/tm/06/a55/tm6a55.pdf

Lapin, A., Schiller, E., Kropf, P., Schilling, O., Brunner, P., Jamaković-Kapić, A. J., Braun, T. & Maffioletti, S. (2014). Real-time environmental monitoring for cloud-based hydrogeological modeling with hydrogeosphere. *2014 IEEE International Conference on High Performance Computing and Communications, 2014 IEEE 6th International Symposium on Cyberspace Safety and Security, 2014 IEEE 11th International Conference on Embedded Software and Systems (HPCC, CSS, ICESS)*, 959–965. https://doi.org/10.1109/HPCC.2014.154

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lehman, J., Clune, J. & Misevic, D. (2020). The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *Artificial Life*, *26*(2), 274–306. https://doi.org/10.1162/artl_a_00319

Lehman, J. & Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. *Artificial Life XI: Proceedings of the 11th International Conference on the Simulation and Syn* This is a preprint of a manuscript currently under peer review. This manuscript is likely to change as it goes through the peer review process. *of Living Systems, ALIFE 2008*, 329–336.

Leike, J., Martic, M., Krakovna, V., Ortega, P. A., Everitt, T., Lefrancq, A., Orseau, L. & Legg, S. (2017). *AI Safety Gridworlds*. http://arxiv.org/abs/1711.09883

Li, Y. (2018). Deep reinforcement learning. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference Tutorial Abstracts*, 19–21. http://arxiv.org/abs/1810.06339

Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Gonzalez, J., Goldberg, K. & Stoica, I. (2018). Ray RLlib: A composable and scalable reinforcement learning library. *Proceedings of the 35th International Conference on Machine Learning*, *Nips*, 3059–3068.

Lin, L.-J. (1993). *Reinforcement learning for robots using neural networks*. School of Computer Science, Carnegie Mellon University.

Littman, M. L. (2015). Reinforcement learning improves behaviour from evaluative feedback. *Nature*, *521*(7553), 445–451. https://doi.org/10.1038/nature14540

Luo, J. & Lu, W. (2014). Comparison of surrogate models with different methods in groundwater remediation process. *Journal of Earth System Science*, *123*(7), 1579–1589. https://doi.org/10.1007/s12040-014-0494-0

Mahmood, A. R., Korenkevych, D., Vasan, G., Ma, W. & Bergstra, J. (2018). Benchmarking reinforcement learning algorithms on real-world robots. *2nd Conference on Robot Learning (CoRL)*, 1–31. http://arxiv.org/abs/1809.07731

Masood, M. & Doshi-Velez, F. (2019). Diversity-inducing policy gradient: Using maximum mean discrepancy to find a set of diverse policies. *IJCAI International Joint Conference on Artificial Intelligence*, *August*, 5923–5929. https://doi.org/10.24963/ijcai.2019/821

McLeod, J., Stojic, H., Adam, V., Kim, D., Grau-Moya, J., Vrancx, P. & Leibfried, F. (2021). *Bellman: A toolbox for model-based reinforcement learning in TensorFlow*. 1–8. http://arxiv.org/abs/2103.14407

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., … Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Mouret, J.-B. & Clune, J. (2015). *Illuminating search spaces by mapping elites*. 1–15. http://arxiv.org/abs/1504.04909

Mullapudi, A., Lewis, M. J., Gruden, C. L. & Kerkez, B. (2020). Deep reinforcement learning for the real time control of stormwater systems. *Advances in Water Resources*, *140*(March). https://doi.org/10.1016/j.advwatres.2020.103600

Nam, K. J., Heo, S. K., Loy-Benitez, J., Ifaei, P. & Yoo, C. K. (2020). An autonomous operational trajectory searching system for an economic and environmental membrane

bioreactor plant using deep reinforcement learning. *Water Science and Technology*, *81*(8), 1578–1587. https://doi.org/10.2166/wst.2020.053

Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C. & Gupta, H. V. (2021). What role does hydrological science play in the age of machine learning? *Water Resources Research*, *57*(3). https://doi.org/10.1029/2020WR028091

Ng, A. Y., Harada, D. & Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. *Sixteenth International Conference on Machine Learning*, *3*, 278–287.

Osband, I., Doron, Y., Hessel, M., Aslanides, J., Sezener, E., Saraiva, A., McKinney, K., Lattimore, T., Szepesvari, C., Singh, S., Van Roy, B., Sutton, R., Silver, D. & Van Hasselt, H. (2019). *Behaviour suite for reinforcement learning*. 1–19. http://arxiv.org/abs/1908.03568

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*(NeurIPS).

Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A. & Battaglia, P. W. (2020). *Learning mesh-based simulation with graph networks*. 1–18. http://arxiv.org/abs/2010.03409

Pollock, D. W. (2012). User guide for MODPATH version 6—a particle-tracking model for MODFLOW. *U.S. Geological Survey Techniques and Methods 6–A41*.

Pollock, D. W. (2016). User guide for MODPATH version 7—a particle-tracking model for MODFLOW. *U.S. Geological Survey Open File Report 2016-1086*, 35.

Rajaraman, V. (2014). JohnMcCarthy — father of artificial intelligence. *Resonance*, *19*(3), 198–207. https://doi.org/10.1007/s12045-014-0027-9

Razavi, S., Tolson, B. A. & Burn, D. H. (2012). Review of surrogate modeling in water resources. *Water Resources Research*, *48*(7). https://doi.org/10.1029/2011WR011527

Riedmiller, M., Hafner, R., Lampe, T., Neunert, M., Degrave, J., van de Wiele, T., Mnih, V., Heess, N. & Springenberg, T. (2018). Learning by playing – solving sparse reward tasks from scratch. *35th International Conference on Machine Learning, ICML 2018*, *10*, 6910–6919.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J. & Battaglia, P. W.

(2020). *Learning to simulate complex physics with graph networks*. http://arxiv.org/abs/2002.09405

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T. & Silver, D. (2020). Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, *588*(7839), 604–609. https://doi.org/10.1038/s41586-020-03051-4

Shen, C. (2018). A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resources Research*, *54*(11), 8558–8593. https://doi.org/10.1029/2018WR022643

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., … 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

Sit, M., Demiray, B. Z., Xiang, Z., Ewing, G. J., Sermet, Y. & Demir, I. (2020). A comprehensive review of deep learning applications in hydrology and water resources. *Water Science and Technology*, *82*(12), 2635–2670. https://doi.org/10.2166/wst.2020.369

Sutton, R. S. & Barto, A. G. (2018). *Reinforcement learning: An introduction* (Second). The MIT Press. http://incompleteideas.net/book/the-book-2nd.html

Tanner, B. & White, A. (2009). RL-Glue: Language-independent software for reinforcement-learning experiments. *Journal of Machine Learning Research*, *10*, 2133–2136.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. de Las, Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T. & Riedmiller, M. (2018). *DeepMind Control Suite*. http://arxiv.org/abs/1801.00690

van Hasselt, H., Guez, A. & Silver, D. (2015). Deep reinforcement learning with Double Q-learning. *Proceedings - IEEE International Conference on Robotics and Automation*,

10695–10701. http://arxiv.org/abs/1509.06461

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., … 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

Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A. S., Yeo, M., Makhzani, A., Küttler, H., Agapiou, J., Schrittwieser, J., Quan, J., Gaffney, S., Petersen, S., Simonyan, K., Schaul, T., van Hasselt, H., Silver, D., … Tsing, R. (2017). *StarCraft II: A new challenge for reinforcement learning*. http://arxiv.org/abs/1708.04782

Wang, P. (2019). On defining artificial intelligence. *Journal of Artificial General Intelligence*, *10*(2), 1–37. https://doi.org/10.2478/jagi-2019-0002

Xing, Z., Qu, R., Zhao, Y., Fu, Q., Ji, Y. & Lu, W. (2019). Identifying the release history of a groundwater contaminant source based on an ensemble surrogate model. *Journal of Hydrology*, *572*(January), 501–516. https://doi.org/10.1016/j.jhydrol.2019.03.020

Zhang, Y., Yu, W. & Turk, G. (2019). Learning novel policies for tasks. *36th International Conference on Machine Learning, ICML 2019*, *2019-June*, 12930–12941.

Zhao, Y., Lu, W. & Xiao, C. (2016). A Kriging surrogate model coupled in simulation-optimization approach for identifying release history of groundwater sources. *Journal of Contaminant Hydrology*, *185–186*, 51–60. https://doi.org/10.1016/j.jconhyd.2016.01.004