

Article

Beacon, a Lightweight Deep Reinforcement Learning Benchmark Library for Flow Control

Jonathan Viquerat *, Philippe Meliga, Pablo Jeken-Rico  and Elie Hachem

MINES Paristech, CEMEF, PSL—Research University, 1 Rue Claude Daunesse, 06904 Sophia Antipolis, France; pablo.jeken_rico@mines-paristech.fr (P.J.-R.); elie.hachem@mines-paristech.fr (E.H.)

* Correspondence: jonathan.viquerat@mines-paristech.fr

Abstract: Recently, the increasing use of deep reinforcement learning for flow control problems has led to a new area of research focused on the coupling and adaptation of the existing algorithms to the control of numerical fluid dynamics environments. Although still in its infancy, the field has seen multiple successes in a short time span, and its fast development pace is certainly partly imparted by the open-source effort that drives the expansion of the community. Yet this emerging domain is still missing a common ground to (i) ensure the reproducibility of the results and (ii) offer a proper ad hoc benchmarking basis. To this end, we propose BEACON, an open-source benchmark library composed of seven lightweight one-dimensional and two-dimensional flow control problems with various characteristics, action and observation space characteristics, and CPU requirements. In this contribution, the seven considered problems are described, and reference control solutions are provided. The sources for the following work are publicly available.

Keywords: deep reinforcement learning; fluid mechanics; flow control



Citation: Viquerat, J.; Meliga, P.; Jeken-Rico, P.; Hachem, E. Beacon, a Lightweight Deep Reinforcement Learning Benchmark Library for Flow Control. *Appl. Sci.* **2024**, *14*, 3561. <https://doi.org/10.3390/app14093561>

Academic Editor: Josep Maria Bergada

Received: 27 February 2024

Revised: 29 March 2024

Accepted: 17 April 2024

Published: 23 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, the area of deep-reinforcement-learning-based flow control has undergone a rapid development, with a surge of contributions on topics such as (but not limited to) drag reduction [1], collective swimming [2], and heat transfers [3]. Unlike traditional methods, deep reinforcement learning (DRL) enables the learning of complex control strategies directly from data, thereby alleviating the effects of local minima and the generalizability of the algorithm for other scenarios [4]. Yet the inherent reproducibility issues of DRL algorithms [5], as well as the variety of computational fluid dynamics (CFD) solvers and the possible variability of environment design among the different actors of the community, make it hard to accurately compare algorithm performances, thus hindering the general progress of the field. Moreover, the standard DRL benchmarks (such as the MUJOCO package [6], or the Atari games from arcade learning environments (ALE) [7]) have a limited relevance in the context of benchmarking DRL methods for flow control as their dynamics, observation spaces, computational requirements, and action constraints display substantial differences to those of numerical flow control environments.

From a general point of view, flow control problems are characterized by a simulated physics environment spanning over at least two dimensions, possibly including time. The control is performed by an agent that modifies boundary conditions, source terms, or other components of the domain in order to optimize a given objective. A notable difficulty is, therefore, designing a robust and at the same time efficient environment that is able to cope with a wide range of actions while preserving low and stable runtimes [8]. One way of minimizing the computational cost is lumping the Navier–Stokes equations, the backbone of most fluid mechanics problems, by limiting the dimensionality and restricting its terms to the dominant ones for the problem at hand. This way, it is possible to retain the main features of the flow while tuning the schemes and discretizations towards higher performances.

In the present contribution, we lay the first stone of a numerical flow control benchmark library which will allow DRL algorithms to systematically assess methodological improvements on physically and numerically relevant problems. The design of the test cases voluntarily limits the computational cost of the solvers, making this library a first benchmarking step before testing on more complex and CPU-intensive cases.

The organization is as follows: a short presentation of the library and its general characteristics is proposed in Section 2, after which the environments are introduced in a systematic way in Section 3. For each case, the physics of the problem are described, followed by insights on the discretization. Then, the environment parameters and specificities are described, after which baseline learning curves and details on the solved environment are provided. Finally, the perspectives for the present work are exposed in a conclusive section.

2. The BEACON Library

This library provides self-contained cases for deep-reinforcement-learning-based flow control. The goal is to provide the community with benchmarks that fall within the range of flow control problems while following three constraints: (i) be written in Python to ensure a simple coupling with most DRL implementations, (ii) follow the general GYM application programming interface [8], and (iii) be cheap enough in terms of CPU usage so that training can be performed on a decent computing station. Aligned with the standardized approach of GYM, which streamlines environment setup and facilitates a focused exploration of RL research, this library serves as a first step for prototyping flow control algorithms before moving on to larger problems that will require more efficient CFD solvers and, most probably, a CPU cluster.

The original version of the library contains seven cases, whose main characteristics are presented in Table 1. The selection of the cases was made in order to (i) follow the aforementioned constraints and (ii) propose a variety of problem (episodic or continuous) and control (discrete or continuous) types, as well as different action space dimensionalities. For each case, some parameters can be tuned that can significantly modify the difficulty and the CPU requirements of the problem. In their default configurations, two cases have low CPU requirements and can be run on a standard laptop; three have intermediate computational loads and will require an extended running time on a laptop or a workstation and can benefit from the use of parallel environments; and, finally, two have high computational needs and will require a decent workstation and parallel sample collection [9,10]. Some cases, such as *rayleigh-v0*, *lorenz-v0* [3], *vortex-v0* [11], and *shkadov-v0* [12], were taken from the literature and fully re-implemented, while others were designed specifically for this work. All the environments of the library follow a similar development pattern, and are self-contained to simplify re-usability. If the Python language is not optimal in terms of performance, the core routines are deferred to NUMBA [13] to reduce the execution time. To avoid version conflicts and improve compatibility, additional package requirements are strictly reduced to GYM [8], NUMPY [14], and MATPLOTLIB [15].

Table 1. Overview of the different environments in their default configurations.

Env. Name	Action Dim.	CPU Requirements	Problem Type	Control Type
<i>shkadov-v0</i>	5	moderate	continuous	continuous
<i>rayleigh-v0</i>	10	high	continuous	continuous
<i>mixing-v0</i>	4	high	episodic	discrete
<i>lorenz-v0</i>	1	moderate	continuous	discrete
<i>burgers-v0</i>	1	low	continuous	continuous
<i>sloshing-v0</i>	1	low	episodic	continuous
<i>vortex-v0</i>	2	moderate	continuous	continuous

All environments are solved with an in-house implementation of the proximal policy optimization (PPO) algorithm [16], for which the default parameters are provided in Table 2. Depending on the control type, results obtained from an off-policy algorithm (either deep

Q-networks (DQN) [17] or the time-delayed deep deterministic policy gradient (TD3) [18] are also shown for comparison. The performances of these algorithms are evaluated on standard benchmarks in Appendix A.

Table 2. Default parameters used for the PPO agent.

–	agent type	PPO-clip
γ	discount factor	0.99
λ_a	actor learning rate	5×10^{-4}
λ_c	critic learning rate	5×10^{-3}
–	optimizer	adam
–	weights initialization	orthogonal
–	activation (actor hidden layers)	tanh
–	activation (actor final layer, continuous)	tanh, sigmoid
–	activation (actor final layer, discrete)	softmax
–	activation (critic hidden layers)	relu
–	activation (critic final layer)	linear
ϵ	PPO clip value	0.2
β	entropy bonus	0.01
g	gradient clipping value	0.1
–	actor network	[64, [[64], [64]]]
–	critic network	[64, 64]
–	observation normalization	yes
–	observation clipping	no
–	advantage type	GAE
λ_{GAE}	bias–variance trade-off	0.99
–	advantage normalization	yes
n_{rollout}	no. of transitions per update	env. specific
n_{batch}	no. of minibatches per update	env. specific
n_{epoch}	no. of epochs per update	env. specific
n_{max}	total no. of transitions per training	env. specific
n_{training}	total no. of averaged trainings	5

3. Environments

3.1. Shkadov

3.1.1. Physics

The initial investigation into vertically falling fluid films was conducted by Kapitza and Kapitza [19], sparking extensive experimental exploration in subsequent decades. These experiments revealed that waves on the surface of a descending thin liquid film exhibit strong non-linearity, manifesting the emergence of saturated waves from small perturbations in amplitude, as well as the presence of solitary waves. For low Reynolds numbers ($Re < 300$), it was noted that the wavelength of the non-linear waves greatly exceeded the thickness of the film, allowing for potential simplifications in their physical modeling (referred to as the long-wave regime). Various physical models were proposed, among them the Shkadov model, which was introduced in 1967 [20]. Despite being shown to be inconsistent [21], the model exhibits intriguing spatio-temporal dynamics while remaining computationally affordable. The flow rate q and the fluid height h are simultaneously evolved using the following set of equations:

$$\begin{aligned} \partial_t h &= -\partial_x q, \\ \partial_t q &= -\frac{6}{5} \partial_x \left(\frac{q^2}{h} \right) + \frac{1}{5\delta} \left(h(1 + \partial_{xxx} h) - \frac{q}{h^2} \right), \end{aligned} \quad (1)$$

where the δ parameter encompasses all the physics of the problem,

$$\delta = \frac{1}{15} \left(\frac{3Re^2}{W} \right)^{\frac{1}{3}}, \quad (2)$$

and where Re and W are the Reynolds and the Webber numbers, respectively, defined by the flat film thickness and the flat film average velocity [22]. System (1) is solved on a 1D domain of length L with the following initial and boundary conditions:

$$\begin{aligned} q(x, 0) &= 1 \text{ and } h(x, 0) = 1, \\ q(0, t) &= 1 \text{ and } h(0, t) = 1 + \mathcal{U}(-\varepsilon, \varepsilon), \\ \partial_x q(L, t) &= 0 \text{ and } \partial_x h(L, t) = 0, \end{aligned} \tag{3}$$

with $\varepsilon \ll 1$ being the noise level generated by the uniform distribution \mathcal{U} . As depicted in Figure 1, the introduction of random uniform noise at the inlet initiates the onset of exponentially growing instabilities (blue region), which subsequently exhibit a pseudo-periodic pattern (orange region). In the subsequent area, the periodicity of the waves breaks, and the instabilities transform into pulse-like formations, characterized by a sharp leading edge followed by minor ripples [23]. Of the sharp pulses, some, known as solitary pulses, move faster than others, leading to the amalgamation of upstream pulses in coalescence events. The parameter δ completely governs the dynamics of these solitary pulses, while the position of the transition region may also be linked to the level of noise at the inlet [22]. It is worth noting that the Shkadov equations have a close relationship with the Kuramoto–Sivashinsky system, which was independently discovered later by Kuramoto and Sivashinsky [24].

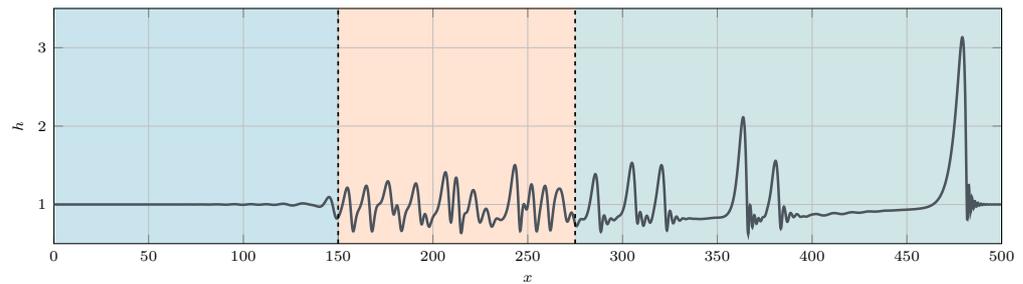


Figure 1. Example of developed flow for the Shkadov equations with $\delta = 0.1$. Three regions can be identified: a first region where the instability grows from a white noise (blue), a second region with pseudo-periodic waves (orange), and a third region with non-periodic, pulse-like waves (green).

3.1.2. Discretization

Equation (1) undergoes discretization employing a finite-difference methodology. Given the presence of abrupt gradients, the convective terms are discretized using a Total Variation Diminishing (TVD) scheme incorporating a minmod flux limiter. The third-order derivative approximation is achieved by combining a second-order centered difference for the second derivative with a second-order forward difference, resulting in the following second-order approximation:

$$\begin{aligned} \partial_{xxx} h|_j &\sim \frac{1}{2\Delta x^3} (-h_{j+3} + 6h_{j+2} - 12h_{j+1} + 10h_j - 3h_{j-1}) \\ &= \partial_{xxx} h(x_j) + \frac{\Delta x^2}{4} \partial_{xxxxx} h(x_j) + O(\Delta x^3). \end{aligned} \tag{4}$$

A second-order Adams–Bashforth method is used to integrate the system in time as follows:

$$h^{n+1} = h^n + \frac{\Delta t}{2} \left(-3f_h(h^n) + f_h(h^{n-1}) \right), \tag{5}$$

where f_h represents the right-hand side of the evolution equation of h in system (1) and is similar for the q field.

3.1.3. Environment

The provided setup is a re-implementation based on the original work by Belus et al. [12], albeit with several distinctions. Notably, the translational invariance aspect introduced in [12] is *not* utilized in this context. Instead, this control scenario is viewed as an opportunity to evaluate algorithms dealing with problems featuring arbitrarily high action dimensionality.

For the control of system (1), a forcing term δq_j is introduced into the equation governing the temporal evolution of the flow rate. Practically, localized jets are inserted at specific locations within the domain, as illustrated in Figure 3 below. The intensities of these jets are determined by the DRL agent. Initially, the first jet is positioned at $x_0 = 150$, with the default jet spacing set to $\Delta x_{\text{jets}} = 10$, akin to in [12]. To conserve computational resources, the domain length is conditioned by the number of jets, n_{jets} , and their spacing as follows:

$$L = L_0 + (n_{\text{jets}} + 2)\Delta x_{\text{jets}}. \tag{6}$$

By default, L_0 is set to be equal to 150 (which corresponds to the beginning of the pseudo-periodic region for $\delta = 0.1$), and n_{jets} is initialized as 1. The spatial discretization step is designated as $\Delta x = 0.5$, while the numerical time-step is $\Delta t = 0.005$ time units. The inlet noise level is configured as $\varepsilon = 5 \times 10^{-4}$, mirroring the setup in [12]. The injected flow rate δq_j takes the following form:

$$\delta q_j(x, t) = Au_j(t) \frac{4(x - x_j^l)(x_j^r - x)}{(x_j^r - x_j^l)^2}, \tag{7}$$

where $A = 5$, an ad hoc non-dimensional amplitude factor, x_j^l and x_j^r represent the left and right limits of jet j , and $u_j(t) \in [-1, 1]$ denotes the agent-provided jet amplitude. Equation (7) corresponds to a parabolic profile of the jet in x , ensuring a zero flow rate at the boundaries. The jet width $x_j^r - x_j^l$ is fixed at 4, in line with [12]. To transition from one action to another, a time dependence is introduced to $u(t)$, involving the following saturated linear variation:

$$u_j(t) = (1 - \alpha(t))u_j^{n-1} + \alpha(t)u_j^n, \text{ with } \alpha(t) = \min\left(\frac{t - t_n}{\Delta t_{\text{int}}}, 1\right), \tag{8}$$

Hence, when the actor provides a new action to the environment at time $t = t_n$, the effective jet amplitude is linearly interpolated from the previous action u_j^{n-1} to the next one u_j^n over period Δt_{int} (set here to 0.01 time units). Following this interpolation, the new jet amplitude remains constant for the remainder of time interval Δt_{const} (set here to 0.04 time units). Consequently, the total action time-step is $\Delta t_{\text{act}} = \Delta t_{\text{int}} + \Delta t_{\text{const}} = 0.05$ time units. The overall episode duration is fixed at 20 time units, corresponding to 400 actions.

The observations provided to the agent consist of the mass flow rates gathered from the combination of regions A_{obs}^j of length $l_{\text{obs}} = 10$, located upstream of each jet. In contrast to the original work, the fluid heights in this area are not relayed to the agent, and the observations remain unclipped.

The reward for each jet j is calculated over region A_{rwd}^j of length $l_{\text{rwd}} = 10$, positioned downstream of it, with the overall reward being the following weighted summation of individual rewards:

$$r(t) = -\frac{1}{l_{\text{rwd}} n_{\text{jets}}} \sum_{j=0}^{n_{\text{jets}}-1} \sum_{x \in A_{\text{rwd}}^j} (h(x, t) - 1)^2 \tag{9}$$

Thus, a perfect flat film yields a maximum reward value of 0. Moreover, the use of a normalization factor enables comparison of scores across different numbers of jets.

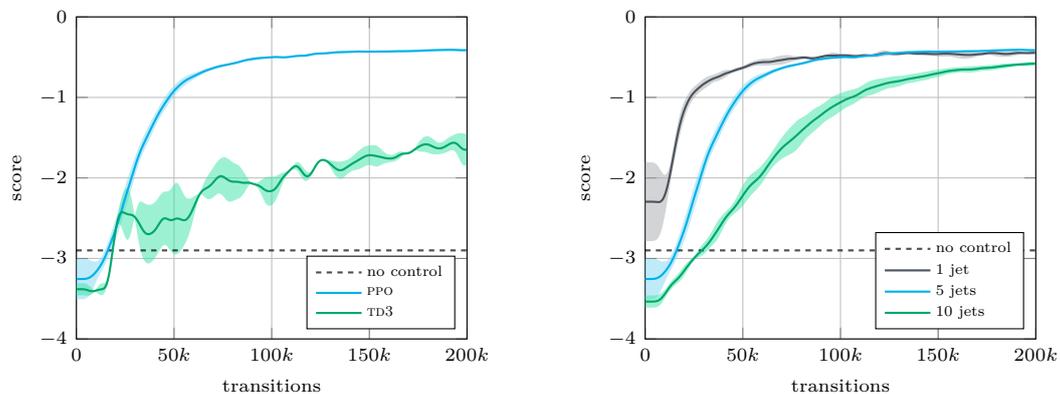
Lastly, an evolved initial state is initialized at the onset of each episode. This initial state is derived by solving the uncontrolled equations from an initial flat film setup over a period of $t_{\text{init}} = 200$ time units. For ease of access, this field is stored in a file and loaded at the commencement of each episode. Optionally, the initial state can be randomized by allowing the loaded initial configuration to evolve freely between 0 and 20 time units.

3.1.4. Results

The previously described environment is referred to as `shkadov-v0`, and its default parameters are provided in Table 3. For the training, we set $n_{\text{rollout}} = 3200$, $n_{\text{batch}} = 2$, $n_{\text{epoch}} = 32$, and $n_{\text{max}} = 200$ k. The related score curves are presented in Figure 2a. We also consider the training on the environment using 1, 5, and 10 jets (see Figure 2b). As could be expected, training is faster for small number of jets, while, for larger numbers of jets, the PPO algorithm struggles due to the increasing dimensionality. In Figure 3, we present the evolution of the field in time under the control of the agent for five jets using the default parameters. As can be observed, the agent quickly constrains the height of the fluid to around $h = 1$ before entering a quasi-stationary state in which a set of minimal jet actuations keeps the flow from developing instabilities in their direct vicinity. In the absence of a jet further downstream, the instability regains amplitude at the outlet of the domain. It must be noted that, due to the random upstream boundary condition, the environment is not deterministic, and, therefore, two exploitation runs with the same trained agent would lead to slightly different final scores.

Table 3. Default parameters for `shkadov-v0`.

L0	base length of domain	150
n_jets	number of jets	5
jet_pos	position of first jet	150
jet_space	spacing between jets	10
delta	physical parameter (2)	0.1



(a) Score curves with default parameters

(b) Score curves for variable number of jets

Figure 2. Score curves for the `shkadov-v0` environment in different configurations. (a) comparison of score curves for PPO and TD3 algorithms in the default configuration using 5 jets. (b) comparison of different number of jets using the PPO algorithm. For each curve, we plot the average (solid color) and the standard deviation (shaded color) obtained from $n_{\text{training}} = 5$ different runs. The dashed line indicates the reward obtained for the uncontrolled environment.

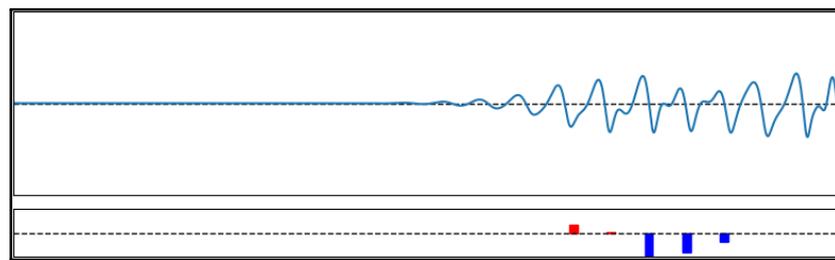
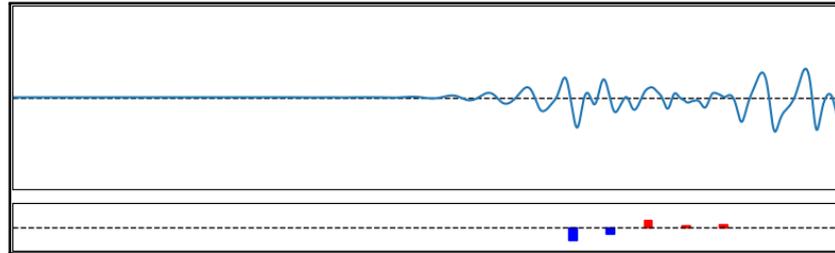
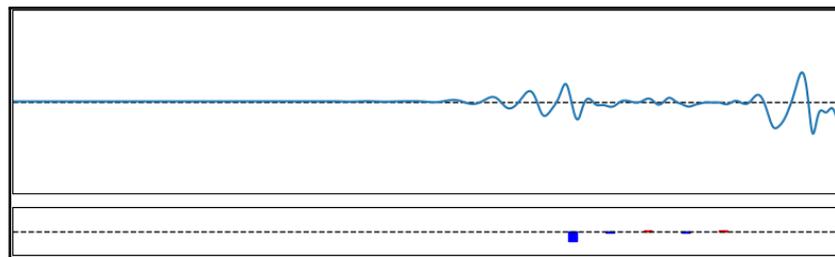
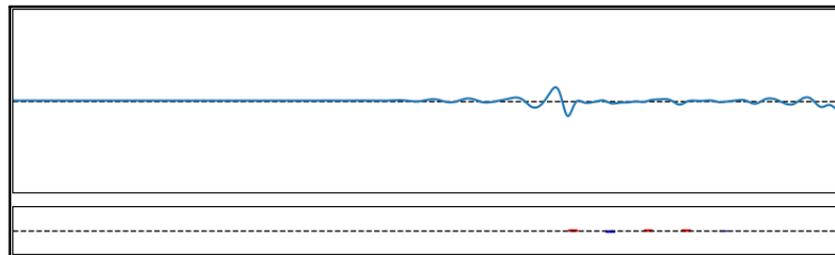
(a) $t = 100$, start of control(b) $t = 120$ (c) $t = 200$ (d) $t = 400$

Figure 3. Evolution of the flow under control of the agent using 5 jets. The jets strengths are represented in the bottom rectangle (red means positive amplitude, blue means negative amplitude). The horizontal and vertical axes are the same as in Figure 1.

3.2. Rayleigh

3.2.1. Physics

We consider the resolution of the 2D Navier–Stokes equations coupled to the heat equation in a cavity of length L and height H with a hot bottom plate and cold top plate. Under favorable circumstances, this setup is known to lead to the Rayleigh–Bénard

convection cell, illustrated in Figure 4. The resulting system is driven by the following set of equations:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0, \\ \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} &= -\nabla p + \sqrt{\frac{\text{Pr}}{\text{Ra}}} \nabla^2 \mathbf{u} + \theta \hat{\mathbf{y}}, \\ \partial_t \theta + (\mathbf{u} \cdot \nabla) \theta &= \frac{1}{\sqrt{\text{Pr Ra}}} \nabla^2 \theta, \end{aligned} \tag{10}$$

where \mathbf{u} , p , and θ are, respectively, the non-dimensional velocity, pressure, and temperature of the fluid. The adimensional temperature θ is described in terms of the hot and cold reference temperatures, respectively denoted as T_H and T_C , as follows:

$$\theta = \frac{T - \hat{T}}{\Delta T}, \text{ with } \hat{T} = \frac{T_H + T_C}{2} \text{ and } \Delta T = T_H - T_C.$$

The dynamics of system (10) are controlled by two adimensional numbers. First, the Prandtl number Pr , which represents the ratio of the momentum diffusivity over the thermal diffusivity as follows:

$$\text{Pr} = \frac{\nu}{\kappa},$$

where ν is the kinematic viscosity and κ the thermal diffusivity. Second, the Rayleigh number Ra , which compares the characteristic time scales for transport due to diffusion and convection as follows:

$$\text{Ra} = \frac{g\alpha\Delta TH^3}{\kappa\nu},$$

where g is the magnitude of the acceleration of the gravity, and α is the thermal expansion coefficient. We also define the instantaneous Nusselt number, Nu , as the adimensionalized heat flux averaged over the hot wall as follows:

$$\text{Nu}(t) = - \int_0^L \partial_y \theta(x', y = 0, t) dx'. \tag{11}$$

System (10) is completed by the following initial and boundary conditions:

$$\begin{aligned} \mathbf{u}(x, y, 0) &= 0 \text{ and } \theta(x, y, 0) = 0, \\ \mathbf{u}(x = 0, y, t) &= 0 \text{ and } \mathbf{u}(x = L, y, t) = 0, \\ \mathbf{u}(x, y = 0, t) &= 0 \text{ and } \mathbf{u}(x, y = H, t) = 0, \\ \theta(x, y = 0, t) &= \theta_H \text{ and } \theta(x, y = H, t) = \theta_C, \\ \partial_x \theta(x = 0, y, t) &= 0 \text{ and } \partial_x \theta(x = L, y, t) = 0, \end{aligned} \tag{12}$$

In essence, the boundary conditions (12) correspond to (i) a no-slip boundary conditions for the fluid on all boundaries, (ii) imposed hot and cold temperatures on the bottom and top plate, respectively, and (iii) adiabatic boundary conditions on the lateral sides of the domain.

Above critical value Ra_c , natural convection is triggered in the cell, increasing the heat exchange between the bottom and top regions of the cell, thus leading to a $\text{Nu} > 1$. Illustrations of the temperature and velocity fields are proposed in Figure 4.

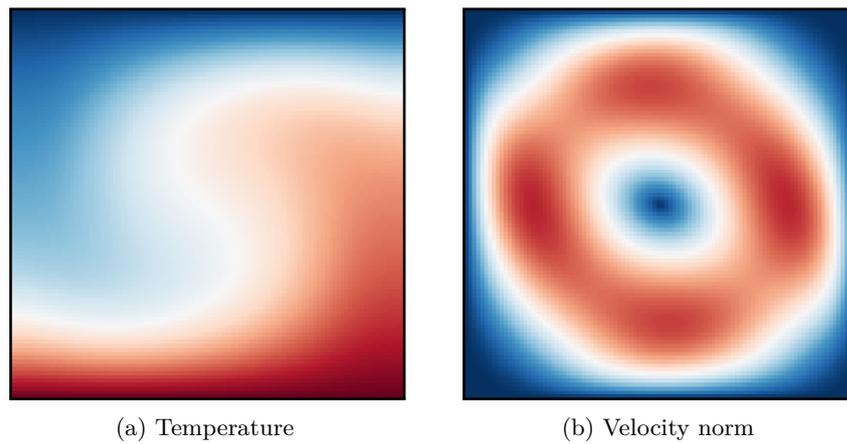


Figure 4. Temperature and velocity profiles for the uncontrolled Rayleigh convection cell with $Ra = 1 \times 10^4$, $Pr = 0.71$, $H = 1$, and $L = 1$. **(Left)** The adimensional temperature ranges from 0 (deep blue) to 1 (deep red). **(Right)** The non-dimensional velocity amplitude ranges from 0 (deep blue) to 1 (deep red).

3.2.2. Discretization

System (10) is discretized using a structured finite-volume incremental projection scheme with centered fluxes, in the fashion of [25]. For simplicity, the scheme is solved in a fully explicit way, except for the resolution of the Poisson equation for pressure. As is standard, a staggered grid is used for the finite-volume scheme; the horizontal velocity is located on the west face of the cells, and the vertical velocity is on the south face of the cells, while the pressure and temperature are located at the center of the cells. The computation of the instantaneous Nusselt number (11) is performed by computing the first-order finite difference in temperature between the center of the first cell at the bottom of the mesh and the reference temperature T_H . Doing so, we obtain $Nu = 2.16$ for $Ra = 1 \times 10^4$ once the permanent regime is reached, which is close to the reference values found in the literature [26].

3.2.3. Environment

The proposed environment is re-implemented based on the original work of Beintema et al. [3]. In the following, we set $Pr = 0.71$, which corresponds to the parameter for air, and $Ra = 1 \times 10^4$ in order to avoid excessive computational loads. Similarly to [3], the control is performed by letting the DRL agent adjust the temperature of $n_s = 10$ individual segments at the bottom of the cavity (see Figure 5). To do so, the actions proposed by the agent are continuous temperature fluctuations $\{\hat{\theta}_i\}_{i \in \llbracket 0, n_s - 1 \rrbracket}$ in the range $[-C, +C]$, with $C = 0.75$, $\theta_H = \frac{1}{2}$, and $\theta_C = -\frac{1}{2}$. To enforce $\langle \theta(y = 0, x, t) \rangle = \theta_H$ and $\theta(y = 0, x, t) \in [\theta_H - C, \theta_H + C]$, the provided $\hat{\theta}_i$ are normalized as follows [3]:

$$\theta_i = \frac{\hat{\theta}_i - \langle \hat{\theta} \rangle}{\max\left(1, \max_j \left(\frac{\hat{\theta}_j - \langle \hat{\theta} \rangle}{C}\right)\right)}. \tag{13}$$

For simplicity, neither spatial nor temporal interpolations are performed between actions. The spatial discretization step is set as $\Delta x = 0.02$, while the numerical time-step is $\Delta t = 0.01$. The action time-step Δt_{act} is equal to 2 time units, with the total episode length being fixed to 200 time units, corresponding to 100 actions.

The observations provided to the agent are the temperatures and the velocity components collected on a grid of $n_p \times n_p$ probes evenly spaced in the computational domain (see Figure 5) plus the three previous observation vectors. The resulting set of observations is flattened in a vector of size $12 n_p^2$ with a default value of $n_p = 4$.

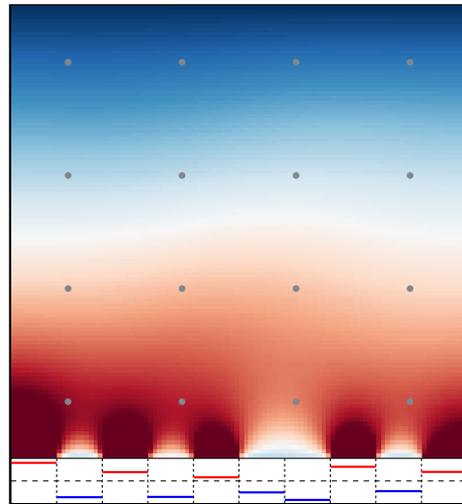


Figure 5. Observation probes and actions imposition for the rayleigh-v0 environment. The observations are collected at the probes regularly positioned in the domain, while the actions are imposed as piecewise constant temperature boundary conditions on the bottom plate with an average value equal to θ_H . Non-dimensional temperature color scale is the same as that of Figure 4.

The reward at each time-step is simply set as the negative instantaneous Nusselt number, such that increasing the reward corresponds to a decrease in the temperature convection as follows:

$$r(t) = -\text{Nu}(t). \quad (14)$$

Finally, each episode starts with the loading of a fully developed initial state obtained by solving the uncontrolled equations during a time of $t_{\text{init}} = 200$ time units. The initial state corresponds to the field shown in Figure 4. For convenience, this field is stored in a file and is loaded at the beginning of each episode.

3.2.4. Results

The environment described in the previous section is referred to as rayleigh-v0, and its default parameters are provided in Table 4. In this section, we note that the entropy bonus for the PPO agent is reduced to $\beta = 0.001$ compared to the default hyperparameters of Table 2. For the training, we set $n_{\text{rollout}} = 1000$, $n_{\text{batch}} = 2$, $n_{\text{epoch}} = 32$, and $n_{\text{max}} = 600$ k. The score curves obtained are presented in Figure 6, while the time evolution of the Nusselt number for the controlled versus uncontrolled cases is shown in Figure 7. As can be observed, the agent manages to devise a set of transition actions toward a stationary state with $\text{Nu}(t) = 1$. The results of Figure 7 are in line with those of [3]. In Figure 8, we present the evolution of the temperature field during the first steps of the environment under the control of the agent using the default parameters.

Table 4. Default parameters used for rayleigh-v0.

L	length of the domain	1
H	height of the domain	1
n_sgts	number of control segments	10
ra	Rayleigh number	1×10^4

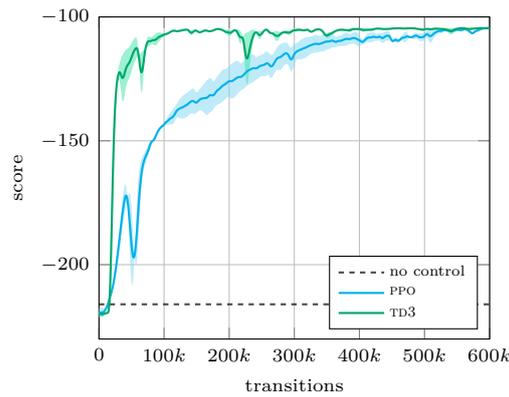


Figure 6. Score curves obtained using the PPO and the TD3 algorithms to solve the rayleigh-v0 environment. The dashed line indicates the reward obtained for the uncontrolled environment.

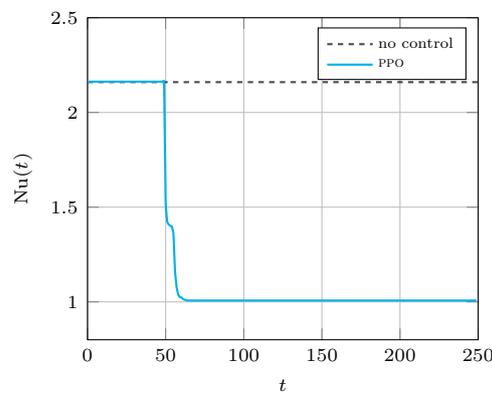


Figure 7. Evolution of the instantaneous Nusselt number during an episode of the rayleigh-v0 environment with and without control. The agent totally disables the convection, leading to a final Nusselt number equal to 1.

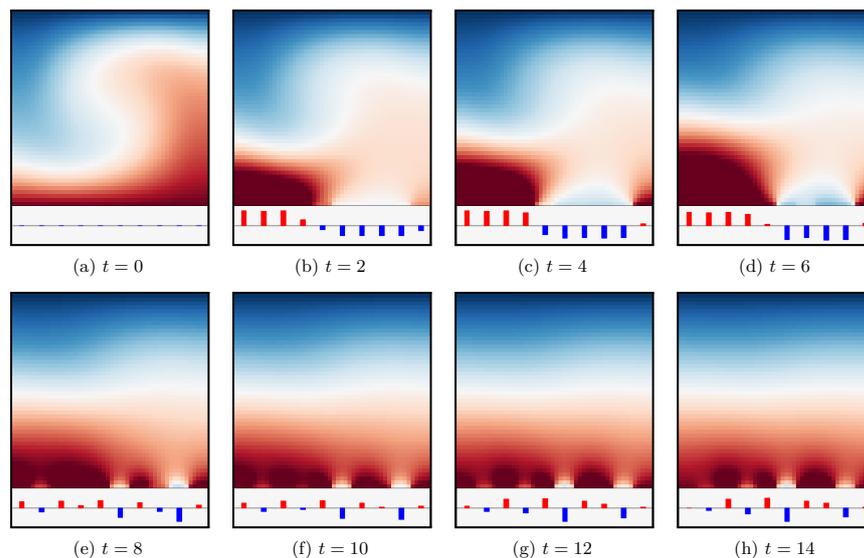


Figure 8. Evolution of the convection cell under control of the agent during the first steps of the environment. After a strong initial forcing, the agent establishes a pattern of alternating hot and cold actions that leads to a stationary configuration with $Nu(t) = 1$. The control of the agent remains the same for the rest of the environment. The local temperature variations due to the control of the agent are represented in the bottom rectangle (red means positive amplitude, blue means negative amplitude). Non-dimensional temperature color scale is the same as that of Figure 4.

3.3. Mixing

3.3.1. Physics

We consider the resolution of the 2D Navier–Stokes equations coupled to a passive scalar convection–diffusion equation in a cavity of length L and height H with moving boundary conditions on all sides. The resulting system is driven by the following set of equations:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0, \\ \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} &= -\nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{u}, \\ \partial_t c + (\mathbf{u} \cdot \nabla) c &= \frac{1}{\text{Pe}} \nabla^2 c, \end{aligned} \tag{15}$$

where \mathbf{u} and p are, respectively, the non-dimensional velocity and pressure of the fluid, and c is the concentration of a passive species. The dynamics of system (15) are controlled by two adimensional numbers. First, the Reynolds number Re , which represents the ratio between inertial and viscous forces as follows:

$$\text{Re} = \frac{UL}{\nu},$$

where U and L are, respectively, the reference velocity and length values, and ν is the kinematic viscosity of the fluid. Second, the Péclet number Pe , which represents the ratio between the advective and diffusive transport rates as follows:

$$\text{Pe} = \frac{UL}{D},$$

where D is the diffusion coefficient of the considered species. In essence, a system with a high Pe value presents a negligible diffusion, and scalar quantities move primarily due to fluid convection. System (15) is completed by the following initial and boundary conditions:

$$\begin{aligned} \mathbf{u}(x, y, 0) &= \mathbf{0} \text{ and } c(x, y, 0) = c_0 \mathbb{1}_{\substack{x_{\min} \leq x \leq x_{\max} \\ y_{\min} \leq y \leq y_{\max}}} \\ \mathbf{u}(x = 0, y, t) &= (0, v_l) \text{ and } \mathbf{u}(x = L, y, t) = (0, v_r), \\ \mathbf{u}(x, y = 0, t) &= (u_b, 0) \text{ and } \mathbf{u}(x, y = H, t) = (u_t, 0), \\ \partial_y c(x, y = 0, t) &= 0 \text{ and } \partial_y c(x, y = H, t) = 0, \\ \partial_x c(x = 0, y, t) &= 0 \text{ and } \partial_x c(x = L, y, t) = 0, \end{aligned} \tag{16}$$

where $\mathbb{1}$ is the indicator function, and $v_l, v_r, u_b,$ and u_t are user-defined values. In essence, the boundary conditions (16) correspond to a multiple-lid-driven cavity, where tangential velocity can be imposed independently on all sides with an initial patch of concentration in the center of the domain. Snapshots of the evolution of the system in time with $(v_l, v_r, u_b, u_t) = (0, 0, 1.0, -1.0)$ are presented in Figure 9.

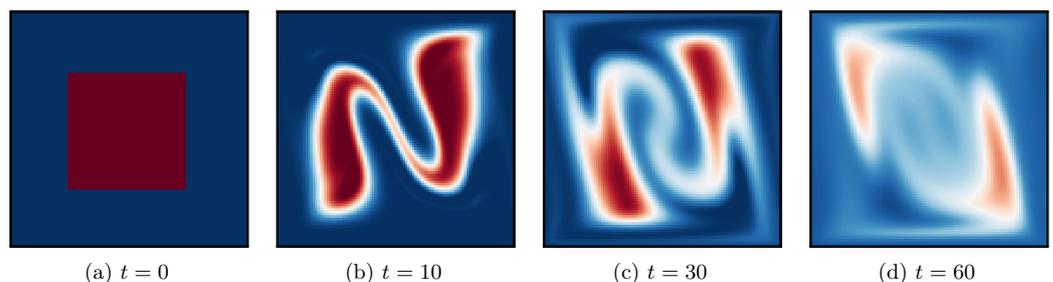


Figure 9. Evolution of the concentration in time with constant boundary conditions $(v_l, v_r, u_b, u_t) = (0, 0, 1.0, -1.0)$. The non-dimensional concentration ranges from 0 (deep blue) to 1 (deep red).

3.3.2. Discretization

System (15) is discretized using a structured finite-volume incremental projection scheme with centered fluxes. For simplicity, the scheme is solved in a fully explicit way, except for the resolution of the Poisson equation for pressure. As is standard, a staggered grid is used for the finite-volume scheme; the horizontal velocity is located on the west face of the cells, and the vertical velocity is on the south face of the cells, while the pressure and concentration are located at the center of the cells.

3.3.3. Environment

In the following, we set $H = 1, L = 1, \Delta X = x_{\max} - x_{\min} = 0.5, \Delta Y = y_{\max} - y_{\min} = 0.5$, and $c_0 = 1$. The control is performed by letting the agent adjust the tangential velocities at the boundaries of the domain. We use a discrete action space of dimension 4 with the following actions:

$$\begin{aligned}
 a = 0 &\iff (v_l, v_r, u_b, u_t) = (0, 0, u_{\max}, -u_{\max}) \\
 a = 1 &\iff (v_l, v_r, u_b, u_t) = (0, 0, -u_{\max}, u_{\max}) \\
 a = 2 &\iff (v_l, v_r, u_b, u_t) = (-u_{\max}, u_{\max}, 0, 0) \\
 a = 3 &\iff (v_l, v_r, u_b, u_t) = (u_{\max}, -u_{\max}, 0, 0),
 \end{aligned}
 \tag{17}$$

where $u_{\max} = \frac{Re v}{L}$. The non-dimensional numbers are chosen as $Re = 100$ and $Pe = 1 \times 10^4$, corresponding to a low-diffusion species. For simplicity, no temporal interpolation is performed between actions. The spatial discretization step is set as $\Delta x = 0.01$, while the numerical time step is $\Delta t = 0.002$. The action time-step Δt_{act} is equal to 0.5 time units, with the total episode length being fixed to 50 time units, corresponding to 100 actions. The observations provided to the agent are the concentration and the velocity components collected on a grid of $n_p \times n_p$ probes evenly spaced in the computational domain plus the three previous observation vectors. The resulting set of observations is flattened in a vector of size $12 n_p^2$ with a default value of $n_p = 4$. The reward at each time-step is simply set as the average absolute distance of the concentration field to a target uniform value as follows:

$$r(t) = -\|c - c_t\|_1 \text{ with } c_t = \frac{\Delta X \Delta Y}{LH} c_0.
 \tag{18}$$

Finally, each episode starts with a null velocity field and an initial square patch of concentration c_0 , as shown in Figure 9a.

3.3.4. Results

The environment described in the previous section is referred to as `mixing-v0`, and its default parameters are provided in Table 5. For the training, we set $n_{rollout} = 2000$, $n_{batch} = 2$, $n_{epoch} = 32$, and $n_{max} = 100$ k. The score curves are presented in Figure 10, along with the score obtained with constant control $a = 0$, which leads to a score of approximately -17.47 . For comparison, the score with no mixing at all (i.e., pure diffusion) yields a score of -28.78 . In Figure 11, we present the evolution of the concentration field of the environment under the control of the agent using the default parameters.

Table 5. Default parameters used for `mixing-v0`.

L	length of the domain	1
H	height of the domain	1
re	Reynolds number	1×10^2
pe	Péclet number	1×10^4
side	initial side length of concentration patch	0.5
c0	initial concentration	1

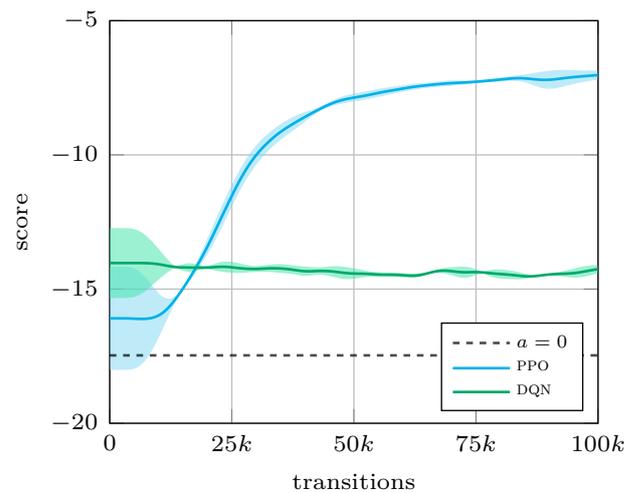


Figure 10. Score curves obtained using the PPO and the DQN algorithms to solve the mixing-v0 environment. The dashed line indicates the reward obtained for the uncontrolled environment.

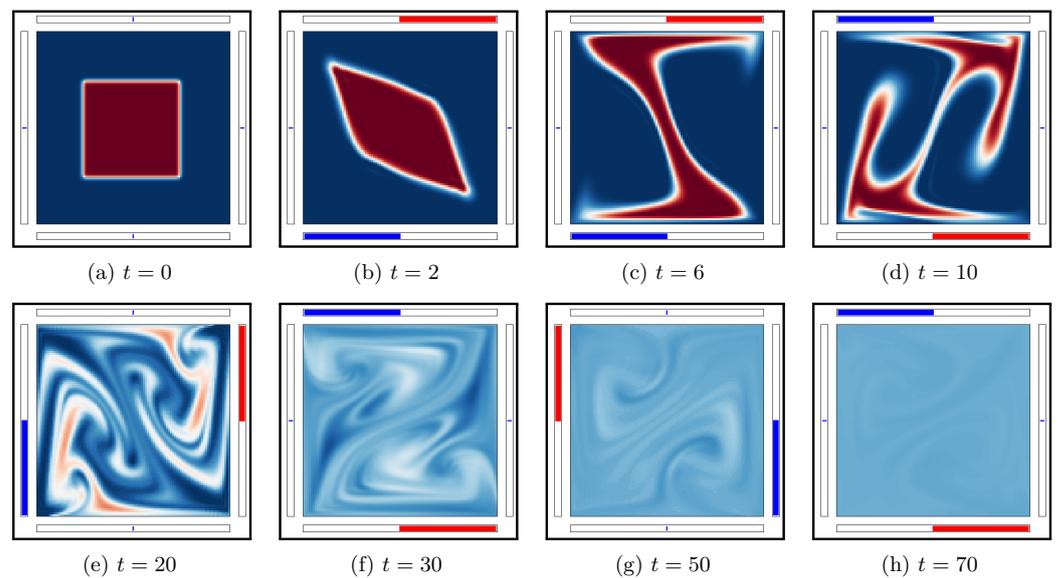


Figure 11. Evolution of the mixing cell under control of the agent. The instantaneous controls are indicated with the four colored bars. Non-dimensional concentration color scale is the same as that of Figure 9.

3.4. Lorenz

3.4.1. Physics

We consider the Lorenz attractor equations, a simple non-linear dynamical system representative of thermal convection in a two-dimensional cell [27]. The set of governing ordinary differential equations reads:

$$\begin{aligned}
 \partial_t x &= \sigma(y - x), \\
 \partial_t y &= x(\rho - z) - y, \\
 \partial_t z &= xy - \beta z,
 \end{aligned}
 \tag{19}$$

where σ is related to the Prandtl number, ρ is a ratio of Rayleigh numbers, and β is a geometric factor. Depending on the values of the triplet (σ, ρ, β) , the solutions to Equation (20) may exhibit chaotic behavior, meaning that arbitrarily close initial conditions can lead to significantly different trajectories [28], of which one common triplet is $(\sigma, \rho, \beta) = (10, 28, 8/3)$,

which leads to the well-known butterfly shape presented in Figure 12. The system has three possible equilibrium points, one in $(0, 0, 0)$ and one at the center of each “wing” of the butterfly shape, the characteristics of which depend on the values of σ , ρ , and β .

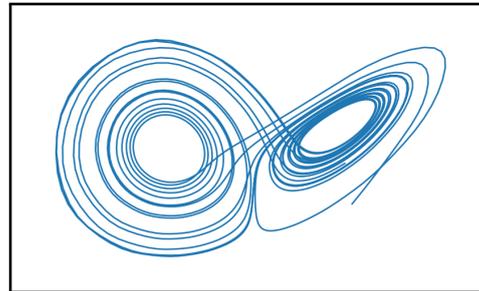


Figure 12. Control-free evolution of the Lorenz system. The attractor’s trajectory forms a distinctive, butterfly-like shape that consists of two large, symmetrically arranged lobes.

3.4.2. Discretization

The ODE system (20) is integrated in time with a five-stage, fourth-order, low-storage Runge–Kutta scheme from Carpenter et al. [29] using constant time-step $\Delta t = 0.05$. Similarly to in [3], the numerical and action time-steps are taken to be equal.

3.4.3. Environment

The proposed environment is re-implemented based on the original work of Beintema et al. [3], where the goal was to maintain the system in the $x < 0$ quadrant. Control (20) is performed by adding an external forcing term to the \dot{y} equation as follows:

$$\begin{aligned}\dot{x} &= \sigma(y - x), \\ \dot{y} &= x(\rho - z) - y + u, \\ \dot{z} &= xy - \beta z,\end{aligned}\tag{20}$$

where u is a discrete action in $\llbracket -1, 0, 1 \rrbracket$. The action time-step is set to $\Delta t_{\text{act}} = 0.05$ time units, and, for simplicity, no interpolation is performed between successive actions. A full episode lasts 25 time units, corresponding to 500 actions. The observations are the variables (x, y, z) and their time derivatives $\dot{x}, \dot{y}, \dot{z}$, while the reward is set to $+1$ for each step with $x < 0$ or 0 otherwise. Each episode is started using the same initial condition $(x_0, y_0, z_0) = (10, 10, 10)$.

3.4.4. Results

The environment described in the previous section is referred to as `lorenz-v0`, and its default parameters are provided in Table 6. In this section, we note that the entropy bonus for the PPO agent is reduced to $\beta = 0.005$ compared to the default hyperparameters of Table 2. For the training, we set $n_{\text{rollout}} = 10,000$, $n_{\text{batch}} = 2$, $n_{\text{epoch}} = 32$, and $n_{\text{max}} = 2000$ k. The related score curves are presented in Figure 13. As can be observed, although the learning is successful, it is particularly noisy compared to some other environments presented in this library. This can be attributed to the chaotic behavior of the attractor, which makes the credit assignment difficult for the agent. A plot of the time evolutions of the controlled versus uncontrolled x parameter is shown in Figure 14. As can be observed, the agent successfully locks the system in the $x < 0$ quadrant, with the typical control peak also observed by Beintema et al. noted with a red dot in Figure 14. For a better visualization, several 3D snapshots of the controlled system are proposed in Figure 15.

Table 6. Default parameters used for lorenz-v0.

sigma	Lorenz parameter	10
rho	Lorenz parameter	28
beta	Lorenz parameter	$\frac{8}{3}$

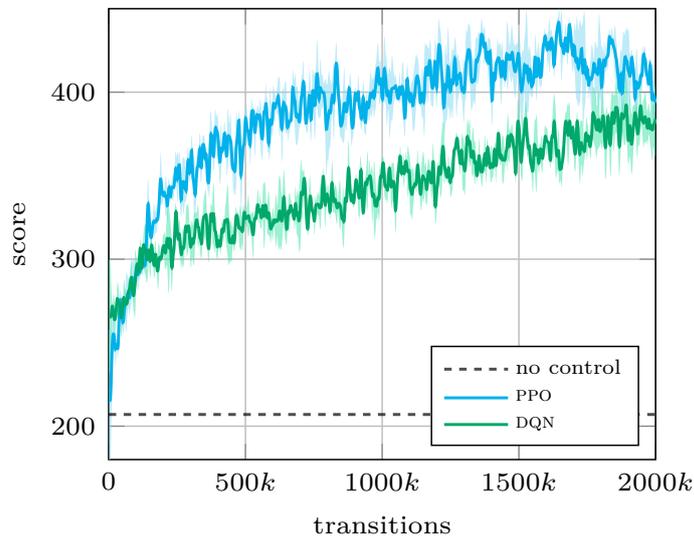


Figure 13. Score curves for the PPO and DQN algorithms when solving the lorenz-v0 environment. The dashed line indicates the reward obtained in the uncontrolled case.

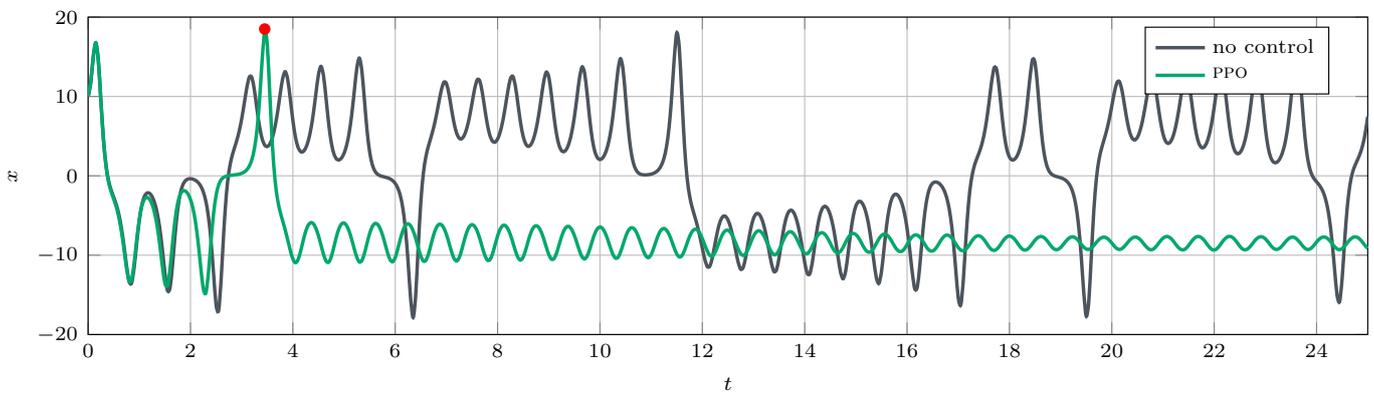


Figure 14. Controlled versus uncontrolled time evolution of the x parameter. The red dot corresponds to the typical control peak that precedes the locking of the system, also observed in [3].

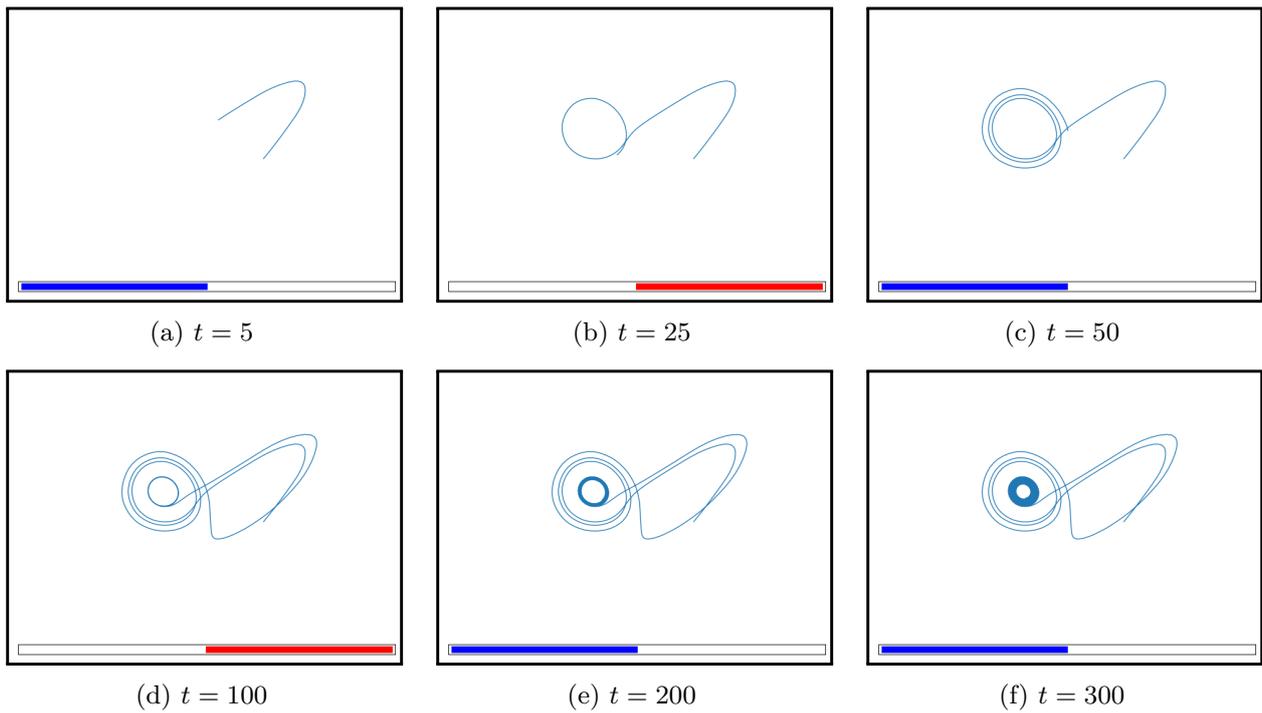


Figure 15. Evolution of the controlled Lorenz system using the PPO algorithm. The instantaneous control value is indicated at the bottom by the colored bar (blue is for -1 , red is for $+1$).

3.5. Burgers

3.5.1. Physics

The inviscid Burgers equation was first introduced by Bateman in 1915 and models the behavior of a one-dimensional inviscid incompressible fluid flow [30] and was then studied by Burgers in 1948 [31].

$$\partial_t u + u \partial_x u = 0. \tag{21}$$

We consider the resolution of the Burgers equation on a domain of length L along with the following initial and boundary conditions:

$$\begin{aligned} u(x, 0) &= u_{\text{target}}, \\ u(0, t) &= u_{\text{target}} + \mathcal{U}(-\sigma, \sigma), \\ \partial_x u(L, t) &= 0, \end{aligned} \tag{22}$$

where σ is the noise level introduced at the inlet, and u_{target} is a constant value. The convection of the random inlet signal leads to a noisy solution in the domain, as is depicted in Figure 16. The initial perturbations steepen while propagating downstream to eventually form shocks.

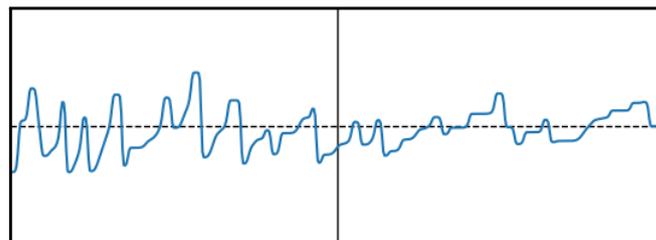


Figure 16. Uncontrolled solution of the Burgers equation with uniform random noise excitation at the inlet. The horizontal axis represents the x coordinate, with the vertical bar indicating the position of the controller. The vertical axis represents the velocity of the fluid.

3.5.2. Discretization

The Burgers equation, Equation (21), is discretized in time with a finite-volume approach. The convective term is discretized using a TVD scheme with a Van Leer flux limiter, while the time marching is performed using a second-order finite-difference scheme.

3.5.3. Environment

The goal of the environment is to control pointwise forcing source term $a(t)$ on the right-hand side of (21) in order to damp the noise transported from the inlet. The forcing is applied at $x_p = 1$, while the length of the domain is set to $L = 2$. The actions provided to the environment, which are expected to be in $[-1, 1]$, are then scaled by an ad hoc non-dimensional amplitude factor $A = 10$. The field is initially set to be equal to $u_{\text{target}} = 0.5$, and the variance of the inlet noise is chosen to be $\sigma = 0.1$. The spatial discretization step is set to $\Delta x = 4 \times 10^{-3}$, while the numerical time-step is equal to $\Delta t = 8 \times 10^{-4}$ time units. The action duration is set to $\Delta t_{\text{act}} = 0.05$ time units for a total episode duration equal to 10 time units, corresponding to 200 actions. The observations provided to the agent are the $n_{\text{obs}} = 5$ values of u upstream of the actuator. Finally, the reward is computed as:

$$r(t) = -\Delta x \|u(x, t) - u_{\text{target}}\|_1 \text{ with } x \in [x_p, L]. \quad (23)$$

3.5.4. Results

The environment described in the previous section is referred to as `burgers-v0`, and its default parameters are provided in Table 7. For the training, we set $n_{\text{rollout}} = 4000$, $n_{\text{batch}} = 2$, $n_{\text{epoch}} = 32$, and $n_{\text{max}} = 500$ k. The score curves obtained are shown in Figure 17, while snapshots of the evolution of the controlled environment are shown in Figure 18. As can be observed, the agent successfully damps the transported inlet noise following an opposition control strategy.

Table 7. Default parameters used for `burgers-v0`.

L	domain length	2
u_target	target value	0.5
sigma	inlet noise level	0.1
amp	control amplitude	10
ctrl_pos	control position	1

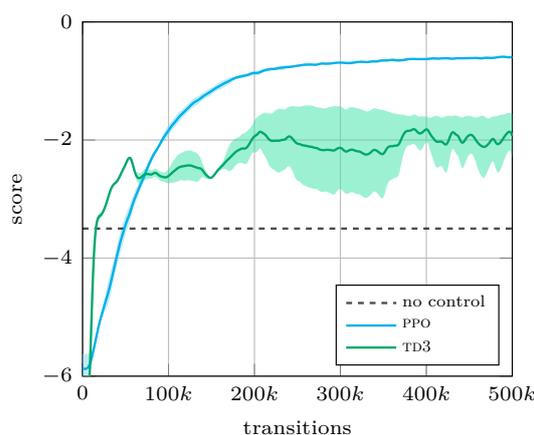


Figure 17. Score curves for the PPO and the TD3 algorithms when solving the `burgers-v0` environment. The dashed line indicates the reward obtained for the uncontrolled case.

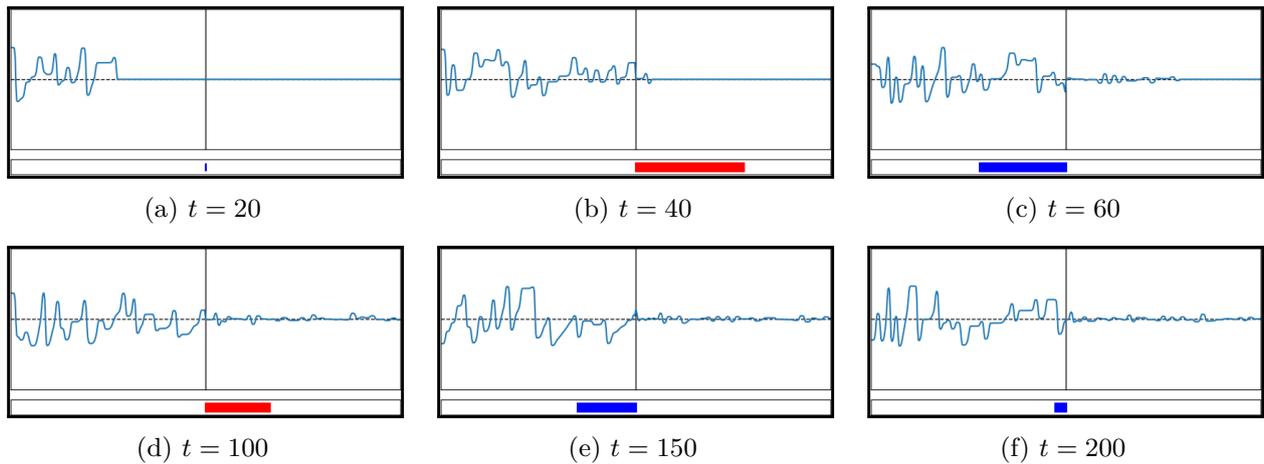


Figure 18. Evolution of the controlled burgers system using the PPO algorithm. The instantaneous control value is indicated at the bottom by the colored bar (blue is for negative actuations, while red is for positive ones). The axes are the same as in Figure 16.

3.6. Sloshing
3.6.1. Physics

We consider the resolution of the 1D Saint–Venant equations (or shallow-water equations), established in 1871 [32], which describe a shallow layer of fluid in hydrostatic balance with constant density. This system is considered in the context of a mobile water tank of length L subjected to an acceleration \ddot{y} , leading to the following equations in the tank referential [33]:

$$\begin{aligned} \partial_t h &= -\partial_x q, \\ \partial_t q &= -\partial_x \left(\frac{q^2}{h} + \frac{1}{2} g h^2 \right) - \ddot{y}, \end{aligned} \tag{24}$$

where h is the fluid height, and q is the fluid flow rate. System (24) is completed by the following initial and boundary conditions:

$$\begin{aligned} q(x, 0) &= 0 \text{ and } h(x, 0) = 1, \\ q(0, t) &= 0 \text{ and } \partial_x h(0, t) = 0, \\ q(L, t) &= 0 \text{ and } \partial_x h(L, t) = 0. \end{aligned} \tag{25}$$

The situation is summed up in Figure 19. When laterally excited, the surface of the fluid sloshes back and forth in the tank, generating complex patterns at the fluid surface, as shown in Figure 20. When the excitation stops, a relaxation phase is observed, usually leaving a single wavefront traveling back and forth in the tank until it dissipates entirely. Due to its simplicity, model (24) does not allow wave breaking nor the formation of drops on the sides of the domain.

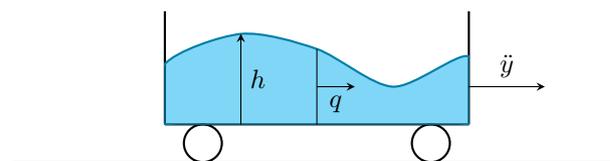


Figure 19. Configuration of the sloshing tank. The fluid flow is determined by the fluid height $h(x, t)$ and by its mass flow rate $q(x, t)$. The movement of the tank is controlled by its acceleration $\ddot{y}(t)$.

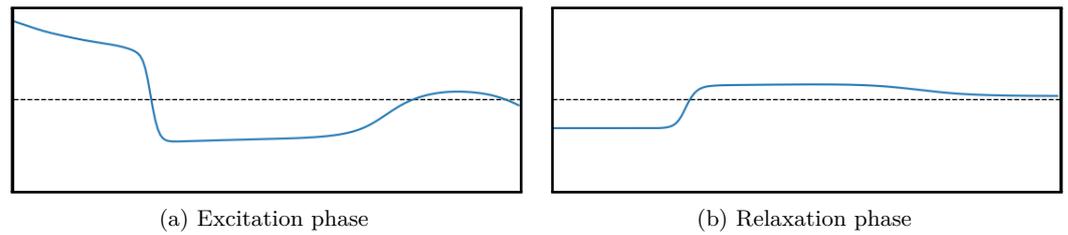


Figure 20. Examples of fluid surface during the excitation phase (left) and the relaxation phase (right). The horizontal axis represents the x coordinate, while the vertical axis represents the height of the fluid. The horizontal line indicates the height of the fluid at rest ($h = 1$).

3.6.2. Discretization

System (24) is discretized using a finite-volume scheme with Rusanov fluxes [34], which is an improved form of the Lax–Friedrichs flux. The time derivatives for both h and q are discretized using the second-order Adams–Bashforth scheme.

3.6.3. Environment

The control of system (24) is performed through the cart acceleration term \ddot{y} . The system is first set in motion during $t_{exc} = 2$ time units using a sinusoid-based signal as follows:

$$\ddot{y}_{exc}(t) = \frac{1}{2}(\cos(\pi t) + 3 \cos(4\pi t)). \tag{26}$$

The resulting fields are stored in a file for simplicity and loaded at the beginning of each episode. By default, the length of the cart is $L = 2.5$, the spatial discretization corresponds to 100 finite-volume cells per unit of length, and the numerical time-step is $\Delta t = 0.001$ time units. The actions provided to the environment, which are expected to be in $[-1, 1]$, are then scaled by an ad hoc non-dimensional amplitude factor $A = 5$. The interpolation between successive actions is identical to (8), with $\Delta t_{int} = 0.01$ time units and $\Delta t_{act} = 0.05$ time units. The total episode time is fixed to 10 time units, corresponding to 200 actions. The observations provided to the agent are the heights collected on the entire domain. To limit the size of the resulting vector, it is downsampled by a factor of 2. Finally, the reward signal is defined as:

$$r(t) = -\Delta x \|h(x, t) - 1\|_2 - \alpha |\ddot{y}(t)|, \tag{27}$$

where $\|\cdot\|_2$ is the 2-norm, and $\alpha = 0.0005$. The Δx factor allows us to obtain comparable reward values for variable discretization levels.

3.6.4. Results

The environment described in the previous section is referred to as `sloshing-v0`, and its default parameters are provided in Table 8. In this section, we note that the critic learning rate of the PPO agent is reduced to $\lambda_c = 1 \times 10^{-3}$ compared to the default hyperparameters of Table 2. For the training, we set $n_{rollout} = 2000$, $n_{batch} = 2$, $n_{epoch} = 16$, and $n_{max} = 200$ k. The score curves are presented in Figure 21, while the time evolutions of the controlled versus uncontrolled fluid level are shown in Figure 22. As can be observed, the agent manages to roughly cut the uncontrolled reward in half by suppressing the back-and-forth wavefront using large actuations in the early stages of control, after which the control amplitude drops significantly.

Table 8. Default parameters used for `sloshing-v0`.

L	length of the tank	2.5
amp	amplitude of the control	5
alpha	control penalization	0.0005
g	gravity acceleration	9.81

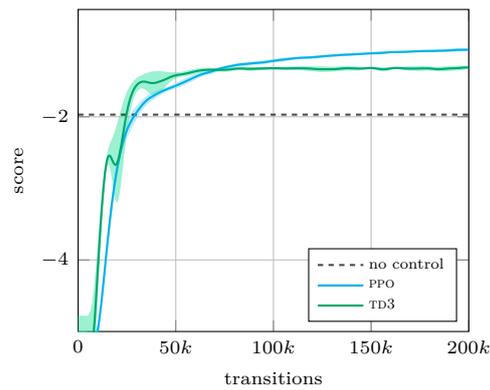


Figure 21. Score curves for the sloshing-v0 environment using the PPO and the TD3 algorithms. The dashed line indicates the reward obtained for the uncontrolled environment.

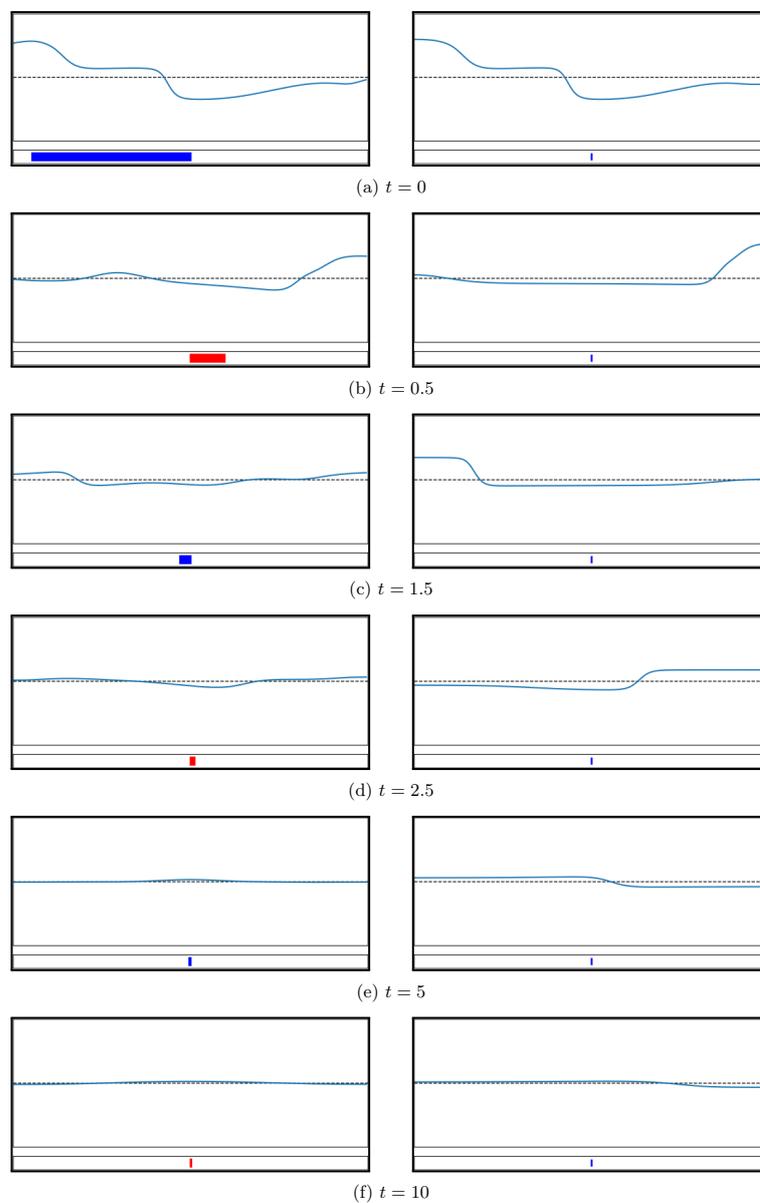


Figure 22. Evolution of the fluid surface with (left) and without (right) agent control. The control amplitude and direction are represented using the rectangle at the bottom (red means positive, blue means negative). The axes are the same as in Figure 20.

3.7. Vortex

3.7.1. Physics

We consider the resolution of a dynamical system modeling the non-linear, vortex-induced vibrations of a rigid circular cylinder. The flow motion is governed by the incompressible Navier–Stokes equations, whereas the cylinder motion is a simple translation governed by a linear mass–damper–spring equation affected by the fluid loading. This is modeled by coupled amplitude equations derived in [11,35] after dominant balance arguments as follows:

$$\begin{aligned}\partial_t A &= \lambda \left(\frac{1}{\text{Re}_*} - \frac{1}{\text{Re}} \right) A - \mu A |A|^2 + \alpha Y \\ \partial_t Y &= (-\omega_* \gamma + i(\omega_s - \omega_*)) Y + \frac{\beta}{\omega_* m} A\end{aligned}\quad (28)$$

where A and Y are unknown slow, time-varying, complex amplitudes modeling, respectively, the flow disturbances and the cylinder center of mass, $\text{Re} = 50$ is the Reynolds number, $\text{Re}_* = 46.6$ is the threshold of instability of the steady cylinder, $\omega_* = 0.74$ is the frequency of the marginally stable eigenmode classically computed from the flow past a fixed cylinder [36], $\omega = 1.1$ is the dimensionless natural frequency of the cylinder in vacuum, $\gamma = 0.023$ is the structural damping coefficient, and $m = 10$ is the ratio of the solid to the fluid densities. The coefficients λ , μ , α , and β in (30) are analytically computable from an asymptotic analysis of the coupled flow–cylinder system, their numerical value being taken from [35] as:

$$\lambda = 9.153 + 3.239i, \quad \mu = 308.9 - 1025i, \quad \alpha = 0.03492 + 0.01472i, \quad \beta = 1. \quad (29)$$

The ability of the model to reproduce the physics of vortex-induced vibrations has been assessed from the study of the non-linear limit cycles, i.e., the periodic, synchronized orbits reached by the system in large periods of time whose analytical expressions are reported in [35]. Of particular importance is the simultaneous existence of multiple stable cycles (either a single limit cycle or three over specific ranges of frequencies), which is shown to trigger a complex hysteretic behavior in the lock-in regime. Since all limit cycle solutions are periodic, the mean mechanical energy averaged over a period is zero, and the mean work received from the fluctuating lift force is entirely dissipated by structural damping. As discussed in [35], it follows that the leading-order mean dissipated energy is a simple quadratic function of the displacement amplitude, meaning that only the upper limit cycle (the one limit cycle yielding the largest displacement amplitude) is of practical interest for the energy extraction problem, that is, in cases where one seeks to leverage such vortex-induced vibrations to generate electrical energy, for instance, by having the oscillation of the cylinder periodically displace a magnet within a coil.

In essence, it can be inferred that there must exist an optimal structural parameter setting for which the dissipated energy is maximum. On the one hand, the flow–cylinder system must be synchronized for the cylinder displacement amplitude to be large. On the other hand, the energy tends to zero in the limit $\gamma \gg 1$, where the work received from the lift force is limited by the low amplitude of the displacement, and in the limit $\gamma \ll 1$, where the displacement is self-limited. As evidenced in [11], the problem shown is that the optimum lies at the edge of a discontinuity, corresponding to parameter settings where the system undergoes a transition from a hysteretic to a non-hysteretic regime. This has important consequences for the application since small inaccuracies in the structural parameters of small external flow disturbances may tip the system outside the hysteresis zone and lead to convergence to cycles of lower energy, resulting in a dramatic drop in the harnessed energy.

3.7.2. Discretization

System (30) is integrated in time using a five-stage, fourth-order, low-storage Runge–Kutta scheme from Carpenter et al. [29] using a constant time-step $\Delta t = 0.1$.

3.7.3. Environment

The proposed environment is re-implemented based on the original work of [11], where the goal was to maximize the cylinder displacement and bypass the existence of low-energy cycles. This is achieved by adding a proportional feedback control in the structure equation, assuming that the state of the system is accessed through measurements of the flow disturbances' position and that an actuator applies a control velocity at the surface of the cylinder as follows:

$$\begin{aligned} \partial_t A &= \lambda \left(\frac{1}{\text{Re}_*} - \frac{1}{\text{Re}} \right) A - \mu A |A|^2 + \alpha Y + k e^{i\phi} A \\ \partial_t Y &= (-\omega_* \gamma + i(\omega_s - \omega_*)) Y + \frac{\beta}{\omega_* m} A \end{aligned} \tag{30}$$

where k is the gain and ϕ the phase shift between the measure and the action. The actions provided to the environment, which are expected to be in $[-1, 1]$, are rescaled to $[0, 0.3]$ for the module and $[-\pi, \pi]$ for the phase. The action time-step is set to $\Delta t_{\text{act}} = 0.5$ time units, a full episode lasting 400 time units, corresponding to 800 actions. The observations are the complex variables A and Y as well as their time derivatives, leading to an observation vector of size 8. The reward at each time-step is computed as:

$$r(t) = 8\omega_s \gamma \left(\frac{d}{dt} \Re [Y e^{i\omega_* t}] \right)^2 - 2w \Re [k e^{i\phi} A e^{i\omega_* t}]^2, \tag{31}$$

where the leftmost term is the mean dissipated energy and is thus associated with performance, the rightmost term estimates the mean kinetic energy expended by the actuator over a limit cycle period and is thus associated with cost, and w is a weighting coefficient set empirically to 50 (a value found to be large enough for cost considerations to impact the optimization procedure but not so large as to dominate the reward signal, in which case, actuating is meaningless). Each episode begins using the same initial condition, $(A_0, Y_0) = (-0.00385 - 0.00378i, 0.00118 - 0.00131i)$, which, in the absence of control, leads to convergence to a low limit cycle, for which the score is equal to 0.00607.

3.7.4. Results

The environment described in the previous section is referred to as `vortex-v0`, and its default parameters are provided in Table 9. In this section, we note that the critic learning rate of the PPO agent is reduced to $\lambda_c = 1 \times 10^{-3}$ compared to the default hyperparameters of Table 2. For the training, we set $n_{\text{rollout}} = 4000$, $n_{\text{batch}} = 2$, $n_{\text{epoch}} = 32$, and $n_{\text{max}} = 1000$ k. The score curves are presented in Figure 23, while the time evolutions of the controlled versus uncontrolled fluid level are shown in Figure 24. As can be observed, the agent manages to lock in a high limit cycle, with a final score more than 3 orders of magnitude larger than that of the uncontrolled low limit cycle, and twice as large as that of the uncontrolled high limit cycle (whose score computed using $A_0 = Y_0 = 0.05(1 + i)$ is 19.25).

Table 9. Default parameters used for `vortex-v0`.

re	Reynolds number	50
w	control penalization	50

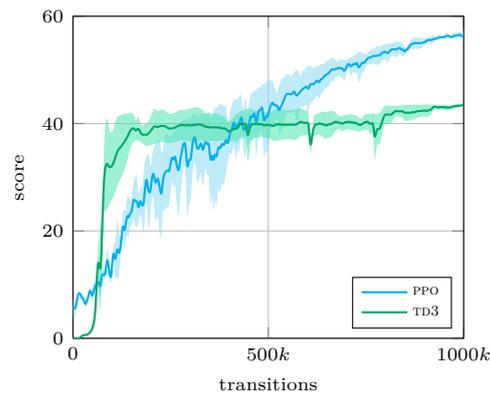


Figure 23. Score curves for the vortex-v0 environment using the PPO and the TD3 algorithms. The uncontrolled environment obtains a score of 0.00607 when locking in on the low limit cycle.

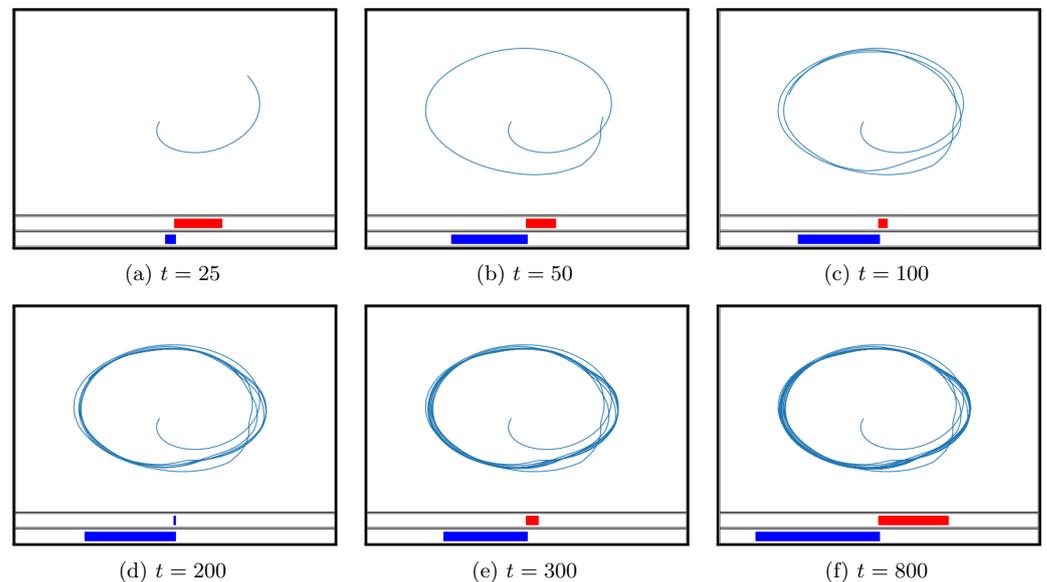


Figure 24. Evolution of the phase portrait of the controlled vortex system using the PPO algorithm. The instantaneous control values are indicated at the bottom by the colored bars (blue means negative control value, red means positive control value).

4. Conclusions and Availability

In the present work, seven fluid dynamics environments are proposed, each of them being representative of a specific physical phenomenon. We observed that the performance ranking usually observed in standard benchmarks such as GYM and MUJOCO (i.e., off-policy methods usually significantly outperform on-policy approaches) is not necessarily valid in the context of flow control environments.

The environments are implemented in a modular way, following the structure of the GYM library to allow for a seamless integration with existing code. Optimizations have been run on each of the cases, and the results have been commented on in each of the respective sections. The sources of the present work are made open in the following repository: <https://github.com/jviquerat/beacon> (accessed on 26 February 2024).

While the present version of the library presents a modest variety of phenomena and control types, its purpose is to grow with new cases proposed by the community, within the constraints detailed in Section 2. To this end, issues and pull requests are accepted in the library repository. With the present work, we hope to provide a solid foundation for the development of a community-driven library of fluid dynamics environments and to foster the development of new control strategies for fluid dynamics.

Author Contributions: Conceptualization, J.V. and E.H.; Formal analysis, J.V., P.M. and P.J.-R.; Software, J.V., P.M. and P.J.-R.; Validation, J.V.; Visualization, J.V.; Writing, J.V., P.M., P.J.-R. and E.H.; Funding acquisition, E.H. All authors have read and agreed to the published version of the manuscript.

Funding: Funded/co-funded by the European Union (ERC, CURE, 101045042). Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Council; neither the European Union nor the granting authority can be held responsible for them.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available at <https://github.com/jviquerat/beacon>.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Implementations Benchmarks

This section is dedicated to the evaluation of the in-house PPO and TD3 algorithms on standard benchmarks. In this regard, we selected two cases from the GYM library, and two cases from the MUJOCO library. Results are presented in Figure A1. The implementations used to solve the BEACON benchmark cases compare favourably with reference implementations.

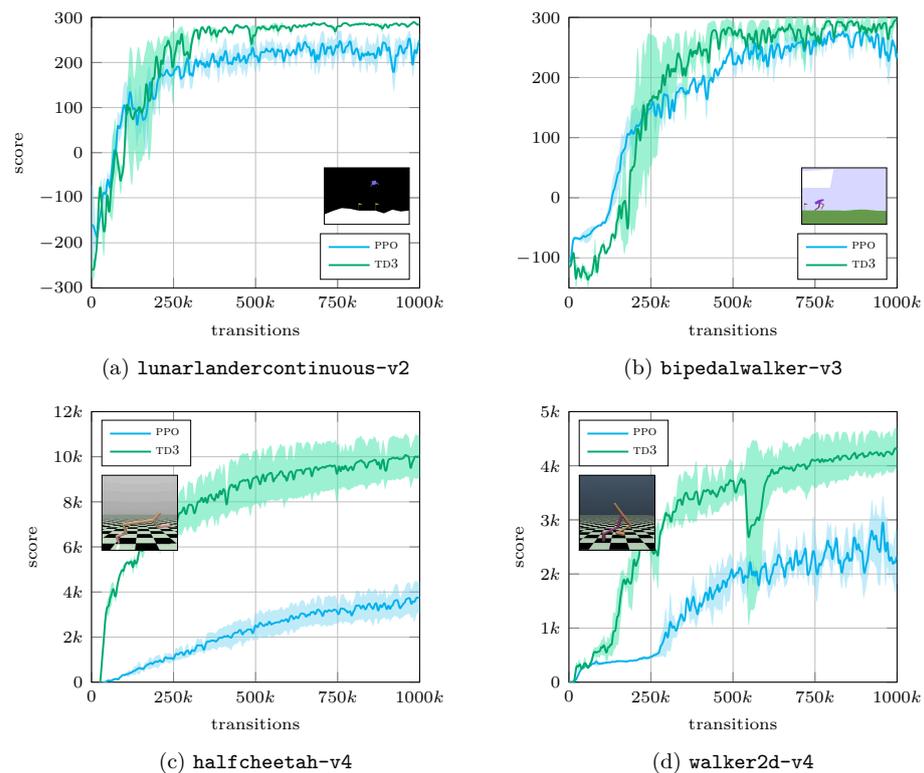


Figure A1. Performance benchmark of our in-house PPO and TD3 algorithms on standard GYM and MUJOCO environments. These results compare favourably with reference implementations [37].

References

1. Wang, Y.Z.; Mei, Y.F.; Aubry, N.; Chen, Z.; Wu, P.; Wu, W.T. Deep reinforcement learning based synthetic jet control on disturbed flow over airfoil. *Phys. Fluids* **2022**, *34*, 033606. [[CrossRef](#)]
2. Novati, G.; Verma, S.; Alexeev, D.; Rossinelli, D.; Van Rees, W.M.; Koumoutsakos, P. Synchronisation through learning for two self-propelled swimmers. *Bioinspir. Biomim.* **2017**, *12*, 036001. [[CrossRef](#)] [[PubMed](#)]

3. Beintema, G.; Corbetta, A.; Biferale, L.; Toschi, F. Controlling rayleigh–bénard convection via reinforcement learning. *J. Turbul.* **2020**, *21*, 585–605. [[CrossRef](#)]
4. Viquerat, J.; Meliga, P.; Larcher, A.; Hachem, E. A review on deep reinforcement learning for fluid mechanics: An update. *Phys. Fluids* **2022**, *34*, 111301. [[CrossRef](#)]
5. Andrychowicz, M.; Raichuk, A.; Stańczyk, P.; Orsini, M.; Girgin, S.; Marinier, R.; Hussenot, L.; Geist, M.; Pietquin, O.; Michalski, M.; et al. What matters in on-policy reinforcement learning? A large-scale empirical study. *arXiv* **2020**, arXiv:2006.05990.
6. Todorov, E.; Erez, T.; Tassa, Y. Mujoco: A physics engine for model-based control. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 5026–5033.
7. Bellemare, M.G.; Naddaf, Y.; Veness, J.; Bowling, M. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Intell. Res.* **2013**, *47*, 253–279. [[CrossRef](#)]
8. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. Openai gym. *arXiv* **2016**, arXiv:1606.01540.
9. Rabault, J.; Kuhnle, A. Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach. *Phys. Fluids* **2019**, *31*, 094105. [[CrossRef](#)]
10. Viquerat, J.; Hachem, E. Parallel bootstrap-based on-policy deep reinforcement learning for continuous fluid flow control applications. *Fluids* **2023**, *8*, 208. [[CrossRef](#)]
11. Meliga, P.; Chomaz, J.M.; Gallaire, F. Extracting energy from a flow: An asymptotic approach using vortex-induced vibrations and feedback control. *J. Fluids Struct.* **2011**, *27*, 861–874. [[CrossRef](#)]
12. Belus, V.; Rabault, J.; Viquerat, J.; Che, Z.; Hachem, E.; Reglade, U. Exploiting locality and translational invariance to design effective deep reinforcement learning control of the 1-dimensional unstable falling liquid film. *AIP Adv.* **2019**, *9*, 125014. [[CrossRef](#)]
13. Lam, S.K.; Pitrou, A.; Seibert, S. Numba: A llvm-based python jit compiler. In Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, Austin, TX, USA, 15 November 2015.
14. Harris, C.R.; Millman, K.J.; Van Der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [[CrossRef](#)] [[PubMed](#)]
15. Hunter, J.D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [[CrossRef](#)]
16. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
17. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)] [[PubMed](#)]
18. Fujimoto, S.; Hoof, H.; Meger, D. Addressing function approximation error in actor-critic methods. *arXiv* **2018**, arXiv:1802.09477.
19. Kapitza, P.L. Wave flow of a thin viscous fluid layers. *Zhurnal Eksperimental'Noi Teor. Fiz.* **1948**, *18*, 3–18.
20. Shkadov, V.Y. Wave flow regimes of a thin layer of viscous fluid subject to gravity. *Fluid Dyn.* **1967**, *2*, 29–34. [[CrossRef](#)]
21. Lavalley, G. Integral Modeling of Liquid Films Sheared by a Gas Flow. Ph.D. Thesis, ISAE—Institut Supérieur de l’Aéronautique et de l’Espace, Toulouse, France, 2014.
22. Chang, H.C.; Demekhin, E.A.; Saprikin, S.S. Noise-driven wave transitions on a vertically falling film. *J. Fluid Mech.* **2002**, *462*, 255–283. [[CrossRef](#)]
23. Chang, H.-H.; Demekhin, E.A. *Complex Wave Dynamics on Thin Films*; Elsevier: Amsterdam, The Netherlands, 2002.
24. Koulago, A.E.; Parségian, D. A propos d’une équation de la dynamique ondulatoire dans les films liquides. *J. Phys. III* **1995**, *5*, 309–312. [[CrossRef](#)]
25. Boivin, S.; Cayré, F.; Herard, J.M. A finite volume method to solve the navier—Stokes equations for incompressible flows on unstructured meshes. *Int. J. Therm. Sci.* **2000**, *39*, 806–825. [[CrossRef](#)]
26. Ouertatani, N.; Cheikh, N.B.; Beya, B.B.; Lili, T. Numerical simulation of two-dimensional rayleigh—Bénard convection in an enclosure. *Comptes Rendus Mécanique* **2008**, *336*, 464–470. [[CrossRef](#)]
27. Saltzman, B. Finite amplitude free convection as an initial value problem. *J. Atmos. Sci.* **1962**, *19*, 329–341. [[CrossRef](#)]
28. Lorenz, E.N. Deterministic nonperiodic flow. *J. Atmos. Sci.* **1963**, *20*, 130–141. [[CrossRef](#)]
29. Carpenter, M.H.; Kennedy, C.A. *Fourth-Order 2n-Storage Runge-Kutta Schemes*; Technical Report; National Aeronautics and Space Administration: Washington, DC, USA, 1994.
30. Bateman, H. Some recent researches on the motion of fluids. *Mon. Weather. Rev.* **1915**, *43*, 163–170. [[CrossRef](#)]
31. Burgers, J.M. A mathematical model illustrating the theory of turbulence. *Adv. Appl. Mech.* **1948**, *1*, 171–199.
32. Saint-Venant, A.J.C. Théorie du mouvement non permanent des eaux, avec application aux crues des rivières et à l’introduction de marées dans leurs lits. *Comptes Rendus Séances Académie Sci.* **1871**, *73*, 148–154.
33. Berger, T.; Puche, M.; Schwenninger, F.L. Funnel control for a moving water tank. *Automatica* **2022**, *135*, 109999. [[CrossRef](#)]
34. Cordier, S.; Darboux, F.; Delestre, O.; James, F. *Etude D’un Modèle de Ruissellement 1D*; Technical Report; University of Orléans: Orléans, France; INRA: Paris, France, 2007.
35. Meliga, P.; Chomaz, J.M. An asymptotic expansion for the vortex-induced vibrations of a circular cylinder. *J. Fluid Mech.* **2011**, *671*, 137–167. [[CrossRef](#)]

36. Barkley, D. Linear analysis of the cylinder wake mean flow. *Europhys. Lett.* **2006**, *75*, 750. [[CrossRef](#)]
37. Achiam, J. Spinning up in Deep Reinforcement Learning, 2018. Available online: <https://spinningup.openai.com/en/latest/index.html> (accessed on 16 April 2024). [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.