Proceedings of Machine Learning Research vol 283:1–16, 2025 7th Annual Conference on Learning for Dynamics and Control

HydroGym: A Reinforcement Learning Platform for Fluid Dynamics

Christian Lagemann[1,2] CLAGE@UW.EDU

Ludger Paehler[3] LUDGER.PAEHLER@TUM.DE

Jared Callaham[1,2]

Sajeda Mokbel[1,2]

Samuel Ahnert[1,2]

Kai Lagemann[4]

Esther Lagemann[1,2]

Nikolaus A. Adams[3]

Steven L. Brunton[1,2] SBRUNTON@UW.EDU

[1] *AI Institute in Dynamic Systems, University of Washington, Seattle, USA*
[2] *Department of Mechanical Engineering, University of Washington, Seattle, USA*
[3] *School of Engineering and Design, Technical University of Munich, Munich, GER*
[4] *Statistics and Machine Learning, German Center for Neurodegenerative Diseases, Bonn, GER*

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

The modeling and control of fluid flows remain a significant challenge with tremendous potential to advance fields including transportation, energy, and medicine. Effective fluid flow control can lead to drag reduction, enhanced mixing, and noise reduction, among other applications. While reinforcement learning (RL) has shown great success in complex domains, such as robotics and protein folding, its application to flow control is hindered by the lack of standardized platforms and the computational demands of fluid simulations. To address these challenges, we introduce HydroGym[1], a solver-independent RL platform for flow control research. HydroGym integrates sophisticated flow control benchmarks, a scalable runtime, and state-of-the-art RL algorithms. Our platform includes four validated non-differentiable fluid flow environments and one differentiable environment, all evaluated with a variety of modern RL algorithms. HydroGym's scalable design allows computations to run seamlessly from laptops to high-performance computing resources, providing a standardized interface for implementing new flow environments. HydroGym aims to bridge the gap in flow control research, providing a robust platform to support both non-differentiable and differentiable RL techniques, fostering advancements in scientific machine learning.

Keywords: Reinforcement Learning, Flow Control, Fluid Dynamics, Benchmark platform

1. https://github.com/dynamicslab/hydrogym

1. Introduction

The control of fluid flows is a critical challenge in many scientific, technological, and industrial systems (Noack et al., 2011; Brunton and Noack, 2015; Lagemann et al., 2023a, 2024b). Improved flow control has the potential to dramatically improve energy, transportation, security, and medical technologies. However, fluid flow control is notoriously challenging as fluid dynamics is nonlinear and multiscale in space and time, resulting in control formulations that are high-dimensional and non-convex. Rapid progress in machine learning over the past decade has considerably advanced our capabilities to solve these exact types of high-dimensional and non-convex optimization problems (Brunton and Kutz, 2022). Indeed, the intersection of machine learning and fluid dynamics is a rapidly growing area of active research (Verma et al., 2018; Duraisamy et al., 2019; Brenner et al., 2019; Brunton et al., 2020; Brenner and Koumoutsakos, 2021; Vinuesa and Brunton, 2022; Bae and Koumoutsakos, 2022; Lagemann et al., 2023b, 2024a). Advancing flow control with machine learning has been particularly challenging because of the need for a working fluid simulation, which requires considerable expertise and overhead. To address these challenges, we introduce HydroGym, a platform that abstracts the computation of the fluid dynamics from the control scheme, enabling researchers on both sides to advance the field.

Adjacent to progress in flow control, reinforcement learning (RL) has made considerable strides in recent years. Key successes of RL involve games (Silver et al., 2016; Vinyals et al., 2019; Brown and Sandholm, 2018; , FAIR; Berner et al., 2019), such as Go (Silver et al., 2016) and StarCraft 2 (Vinyals et al., 2019), robotics (Kalashnikov et al., 2021; Chebotar et al., 2021), protein design (Lutz et al., 2023), the alignment of large language models to human preferences (Ouyang et al., 2022), and increasingly many scientific applications, such as the control of a plasma in a live Tokamak reactor (Degrave et al., 2022). These advances were enabled by the interplay of algorithm improvements, co-design of RL algorithms (Lu et al., 2022) and machine learning frameworks, and the increased availability and accessibility of RL environments such as Drake (Tedrake et al., 2014), Gymnax (Lange, 2022), Brax (Freeman et al., 2021), Dojo (Howell et al., 2022), and Isaac Gym (Makoviychuk et al., 2021).

A number of recent works have begun to combine RL with flow control, focusing on controlling individual environments (Wang et al., 2022; Rabault et al., 2019; Noack and Morzyński, 2017; Guastoni et al., 2023; Pivot et al., 2017), finding single- or multi-agent strategies to navigate through flow environments (Gunnarson et al., 2020; Verma et al., 2018) and the learning and tuning of numerical schemes in fluid dynamics (Bae and Koumoutsakos, 2022; Feng et al., 2023). These works explore individual flow control environments, lacking the variety of environments found in MuJoCo (Todorov et al., 2012; Xian et al., 2023) or in the adjacent autoregressive time-series prediction benchmarks for fluid mechanics (Takamoto et al., 2022; Gupta and Brandstetter, 2022). Moreover, while other general PDE-based RL environments are emerging for diverse control tasks, e.g., PDE Control Gym (Bhan et al., 2024) or ControlGym (Zhang et al., 2024), HydroGym distinguishes itself through its dedicated focus on fluid dynamics and flow control. Distinct from these broader platforms, HydroGym offers a curated suite of validated fluid dynamics benchmarks, specifically designed to address the complexities of this domain. Furthermore, HydroGym's solver-independent architecture and scalable design, coupled with its support for both differentiable and non-differentiable RL methods, positions it as a specialized and powerful tool for advancing research specifically within fluid flow control, rather than a general-purpose RL environment. That is, to bring modern RL (Agarwal et al., 2022) to flow control, policies must be trained and compared across a *diversity of environments* to amortize computation in future downstream RL applications

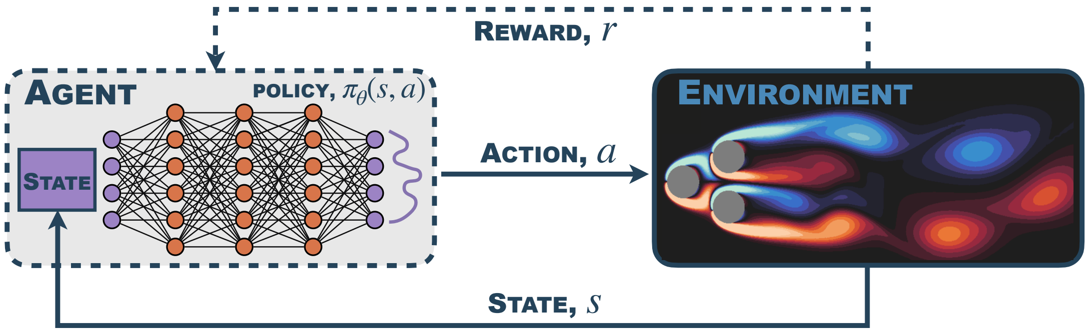

Figure 1: HydroGym provides a flexible, solver-independent reinforcement learning platform for advanced fluid flow control, including differentiable and non-differentiable environments.

for which we will then only need to fine-tune our existing agent. The computational expense of fluid dynamics simulation environments poses an additional challenge to integrating flow control environments of varying complexity. This necessitates scaling not only in the environment through environment copies, but also in the RL agent to optimally utilize available computational capability.

2. Core API design and implementation details

This section outlines the design and implementation of HydroGym's core API. First, we introduce the three core API classes - `PDEBase`, `TransientSolver`, and `FlowEnv` - which provide the framework for defining and solving flow control problems, managing transient simulations, and integrating with RL environments. This is followed by a description of a prototypical Computational Fluid Dynamics (CFD) backend using the `Firedrake` solver.

2.1. HydroGym - a Standardized `Env` API for Fluid Flow Interactions

HydroGym is intended to serve as a solver-independent and easily extendable platform. To provide a standardized interaction between CFD solvers and RL environments, the core API defines three critical classes that form a flow control problem: `PDEBase`, `TransientSolver`, and `FlowEnv`.

The `PDEBase` class is a template for implementing a PDE control problem and defines the core functionality required to interact with the underlying solver. This class encapsulates the flow control configuration, including domain discretization, boundary conditions, actuation models, and other relevant details. Information about solving the equations is delegated to the solver implementations. Therefore, abstract methods defined by this class fall into two categories. First, many implementation building blocks are common across all flow configurations, i.e., mesh and checkpoint loading; these are implemented by a generic backend. The second category defines classes that vary across flow environments, including boundary conditions, actuation and observation models, and objective function evaluation. Environments may also define an optional `render` method for visualization. Hence, the `PDEBase` class contains a complete environment definition, similar to other RL environments, defining an interface for agent - solver interactions.

However, CFD algorithms are more specialized and complex than general ODE solvers. To address this, we introduce another core class in HydroGym - the `TransientSolver`. This class provides a separate interface for the time-stepping routine that updates the transient flow state based on actuation inputs. It includes a `solve` method to integrate the flow over a specified time interval, with optional support to define a feedback controller of the form $u(t, y)$, where u is the control output, t the time step, and y the state variable. Specific time-stepping schemes can be implemented

```
import hydrogym

env_config = {
    "flow": hydrogym.Cylinder,   # PDEBase class
    "solver": hydrogym.BDF,     # TransientSolverClass
}
env = hydrogym.FlowEnv(env_config)

# Interact with flow via Gym API
obs = env.reset(), act = 0.0
for i in range(num_steps):
    obs, reward, done, info = env.step(act)
```

Figure 2: A minimal definition of HydroGym's API. `PDEBase` and `TransientSolver` are configured before initializing `FlowEnv`. Interaction with the environment then follows the Gym convention.

in the backend by defining the `step` method for state updates in a `PDEModel` and the `reset` method to restore initial states at the end of a training episode.

The `PDEBase` and `TransientSolver` classes are sufficient for simulating and controlling individual flow configurations using traditional controller designs. HydroGym also includes a `FlowEnv` class to enable RL training. Based on a user-specified `env_config` dictionary, the `FlowEnv` connects `PDEBase` and `TransientSolver` instances following the Farama Foundation Gymnasium[2] `Env` API. Thus, new environments and solvers defined entirely at HydroGym's backend level can be automatically instantiated as Gym environments. Additionally, both `TransientSolver` and `FlowEnv` support construction with a list of callbacks defined in the `CallbackBase` class. These callbacks typically store data for later visualization but can also modify the flow object itself. An example setup of a HydroGym environment is shown in Fig. 2.

2.2. Utilizing the `Firedrake` Solver as a Prototypical CFD Backend

We provide an implementation of the core API using Firedrake as a backend. Firedrake is a Python framework for defining and solving PDEs using the finite element method (FEM) (Ham et al., 2023). While PDEs can be defined in near-math syntax using the Unified Form Language (UFL) for variational forms, Firedrake combines sophisticated code generation with PETSc linear algebra for scalable, parallelizable high-performance solvers (Alnaes et al., 2013).

The backend `PDEBase` methods are implemented in a generic `FlowConfig` class which defines the test and trial spaces for our finite element solver, provides control input handling, and gives several CFD-specific utilities. New flow environments can be implemented by subclassing `FlowConfig` and defining the flow geometry, using the open-source Gmsh format (Geuzaine and Remacle, 2009), boundary conditions, objective functions, and models for observation and actuation. Using Firedrake, we provide a Newton-Krylov solver to find steady state solutions and a semi-implicit backward differentiation formula (BDF) (Forti and Dedè, 2015) for transient flows. The BDF solver is favored over the classical prediction-correction schemes as it allows for the use of first order elements to construct both the velocity and pressure spaces instead of the normal Taylor-Hood elements (which are second order in velocity and first order in pressure). This is much more computationally efficient, but can be less accurate for flows which contain strong velocity gradients. As such, the

2. https://github.com/Farama-Foundation/Gymnasium

Environment	Environment Characteristics						
	State Dimension	Observation type	Default observation dimension	Actuation type / range	Actuation dimension	Validated dynamics regime	Timestep dt
Cylinder - Jet	26,232	velocity, pressure, vorticity, integral forces	12	blowing/suction $[-0.1, 0.1]$	1	$10 < Re < 250$	0.15
Rotational Cylinder	26,232	velocity, pressure, vorticity, integral forces	12	rotation $[-5.0, 5.0]$	1	$10 < Re < 250$	0.15
Pinball	233,492	velocity, pressure, vorticity, integral forces	24	rotation $[-5.0, 5.0]$	3	$10 < Re < 150$	0.15
Cavity	339,858	velocity, pressure, vorticity turbulent kinetic energy	12	blowing/suction $[-0.1, 0.1]$	1	$Re = 7500$	0.025
Kolmogorov	65,536	velocity	8	sinusoidal forcing $[-0.1, 0.5]$	4	$10 < Re < 200$	0.001

Table 1: Overview of characteristics for four non-differentiable (cylinder-jet/rotation, pinball, cavity) and one differentiable (Kolmogorov) environments.

solver requires a stabilized formulation to maintain accuracy, and both the SUPG and GLS formulations are implemented in the `FlowConfig` backend.

The actuation is represented as time-varying Dirichlet boundary conditions, such as rotation or blowing/suction. This presents more challenges compared to body force actuation models which are often used in theoretical studies, but it more closely approximates physically plausible actuators. Numerically, each actuator is represented by a vector obtained by interpolating a zero velocity field onto the finite element space with a unit value for the corresponding Dirichlet boundary condition. To simulate physical damping in the actuator and prevent numerical instabilities from high-frequency inputs, we model the actuators as leaky integrators. These act as first-order low-pass filters for the action inputs, with cutoff frequencies approximately 100 times faster than the linear instability frequencies in each flow.

3. Environments

To provide a variety of flow control problems for RL training, an initial set of four non-differentiable and one differentiable environment is implemented, representing a wide variety of flow phenomena including global instabilities, vortex shedding, wake interaction, shear layer instabilities, and chaotic dissipation events. All environments are formulated as discrete-time Markov Decision Processes. This discretization is inherent in the numerical simulation of fluid dynamics, which proceeds in discrete time steps. Each environment's dynamics are governed by the Navier-Stokes equations and solved numerically using the methods described in Sec. 2.2. Starting from a given state (flow field), applying a control action results in a transition to a new state after a fixed time step, dictated by the chosen numerical solver and configuration. For all environments, both state and action spaces are continuous and normalized in the range $[-1.0, 1.0]$. The observations include a *user-specified* setting of point probes (e.g. location and number of probes) that extract velocity, vorticity, and/or pressure information. An overview of the environments is provided in Tab. 1.

3.1. Non-Differentiable Flow Environments

3.1.1. FLOW PAST A CIRCULAR CYLINDER

The flow past a cylinder is a canonical system characterized by periodic vortex shedding in the wake of a bluff body. Vortices are formed via an adverse pressure gradient on the cylinder surface which leads to boundary layer separation. In the uncontrolled case, vortices shed in a staggered pattern as shown in Fig. 3. The resulting oscillating forces may contribute to structural fatigue and failure. Despite its simple geometry, the flow exhibits several key features of more complex wakes, including global instabilities, vortex shedding, and the stabilizing effects of nonlinearity (Noack et al., 2003; Sipp and Lebedev, 2007; Sipp et al., 2010; Noack et al., 2011). This flow is often studied

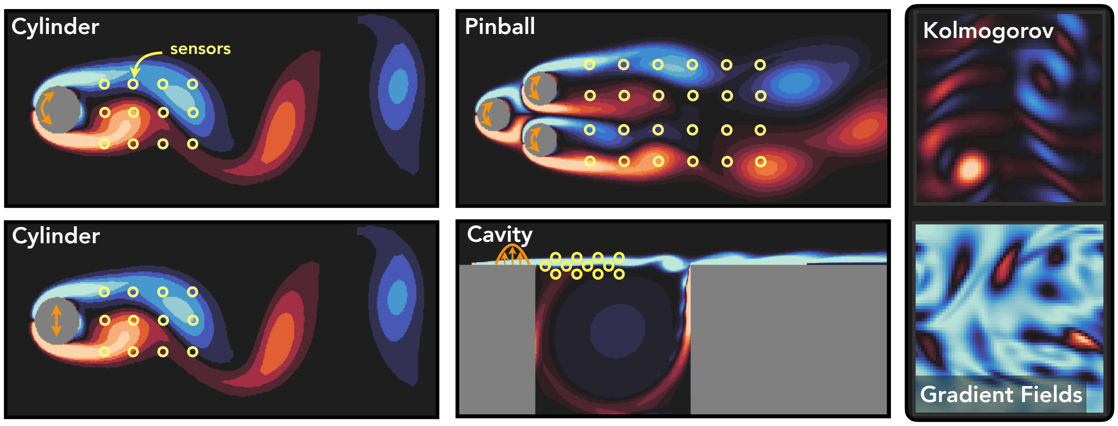

Figure 3: HydroGym includes four non-differentiable environments: two cylinder flows with different actuation, the fluidic pinball with rotating cylinders, and a cavity flow with wall blowing/suction. HydroGym also implements a differentiable Kolmogorov environment.

as an example of emergent low-dimensional structure in high-dimensional systems: at moderate Reynolds numbers the dynamics can be accurately modeled with a two degree-of-freedom Stuart-Landau equation (Noack et al., 2003; Sipp and Lebedev, 2007). Understanding the dynamics of these wakes and how to control them is critical for applications such as improving the efficiency of large-scale wind farms and designing durable ocean structures (Bearman, 2011; Porté-Agel et al., 2013; Araya et al., 2017). The standard rewared objective r for the cylinder wake is drag minimization and the reduction of lift force oscillations and reads $r = -|C_D| - \omega|C_L|$, where C_D, C_L denote the normalized drag and lift coefficients, and $\omega = 0.25$ is a scaling factor. Since the unstable symmetric state is also the minimum-drag configuration, this objective corresponds to stabilizing this unstable fixed point. Two different actuation strategies are implemented: First, a zero-net mass blowing/suction strategy is provided, comprised of two equal and opposite jets - one at the top and one at the bottom of the cylinder. The RL agent only controls the mass flux of the upper jet, which specifies the lower jet. Second, a rotating surface actuator is implemented for which the RL agent controls the angular velocity.

3.1.2. THE FLUIDIC PINBALL

The fluidic pinball was introduced by Noack and Morzyński (2017) and consists of three cylinders arranged in an equilateral triangle. Although still a relatively simple geometry, the dynamics are quite complex due to the interaction of the three wakes. The system undergoes a Hopf bifurcation route to chaos, characterized by a primary supercritical Hopf bifurcation at $Re \approx 18$, followed by a secondary supercritical pitchfork bifurcation at $Re \approx 68$. The Hopf bifurcation corresponds to the destabilization of the steady solution with respect to vortex shedding, while the pitchfork bifurcation occurs when the mean flow breaks the symmetry with respect to the mirror plane (Pastur et al., 2019). This is followed by a transition to a quasi-periodic flow regime, similar to a Neimark–Säcker bifurcation at $Re \approx 100$. Finally, the dynamics eventually bifurcate into a chaotic regime at $Re \approx 115$. The difficulty of the control problem can thus be tuned by adjusting the Reynolds number between 30 (periodic wake) and 120 (fully chaotic regime). As a result, the pinball has emerged as a benchmark problem to test advanced control architectures on chaotic dynamics (Bieker et al., 2020; Peitz et al., 2020; Maceda et al., 2021). Our implementation closely matches that described by Noack and Morzyński (2017); Pastur et al. (2019), including domain size and mesh resolution. Each cylinder is actuated with an independent rotation. Similar to the cylinder flow, the standard reward objective for the pinball is drag minimization and reads $r = -\sum_{i=0}^{2} C_{D_i}$, where C_{D_i} denotes the normalized drag coefficients of the i-th cylinder in the pinball configuration.

3.1.3. SHEAR-DRIVEN CAVITY FLOW

The open cavity flow is another canonical problem in model reduction and model-based flow control, exhibiting rich dynamical behavior with a simple geometry. In practical applications, the flow is often compressible and known for its *Rossiter modes*: shear layer oscillations that become self-exciting and self-sustaining via an acoustic feedback mechanism. Although the acoustic feedback is not present in the incompressible case, it does exhibit a range of interesting dynamics which are mainly driven by two mechanisms: First, a Kelvin-Helmholtz instability causes a shear layer roll-up and the formation of vortices at various frequencies. Second, smaller perturbations caused by the interaction of the shear layer vortices with the downstream cavity edge generate perturbations that recirculate within the cavity towards the leading edge and form a slower feedback cycle. Thus, two relevant time-scales emerge, creating a challenge for control. The control objective for this flow is to minimize the fluctuation kinetic energy, a proxy for the damped acoustics of interest in aerodynamics, and is formulated as $r = -\frac{1}{2}\int_{\Omega} \|\mathbf{u} - \mathbf{u}_B\|^2 \, d\mathbf{x}$, where Ω captures the entire flow field domain and $\mathbf{u}$, $\mathbf{u}_B$ denote the instantaneous and base state velocity fields, respectively. Since the fluctuation of the kinetic energy is defined relative to a base state, this environment requires first solving for a steady-state solution which is subtracted from the instantaneous field.

3.2. Differentiable Flow Environment - The Chaotic Kolomogorov Flow

Dynamical systems can experience extreme events where key quantities deviate significantly from the mean for finite periods. Examples include rogue waves, power grid shocks, high-energy acoustic bursts, and extreme energy dissipation and enstrophy events in fluid flows (Schmidt and Schmid, 2019; Yeung et al., 2015). Therefore, identifying and controlling these events is crucial.

To mimic these challenges, we provide a differentiable implementation of the “Kolmogorov" flow which exhibits sporadic energy dissipation bursts due to a nonlinear energy transfer between scales (Blanchard and Sapsis, 2019). The numerical setup comprises a doubly-periodic domain of $[(0, 2\pi) \times (0, 2\pi)]$ with a sinusoidal forcing term of $\mathbf{f} = (\sin \mathrm{k}_f y, 0)$, where k_f is the forcing wavenumber. The dynamics are simulated in the Fourier space using a pseudo-spectral method with 2/3 aliasing (Yin et al., 2004). A fourth order Runge Kutta and Crank Nicolson time integration is adapted from Dresdner et al. (2023). The solver is developed in JAX (Bradbury et al., 2018), leverages `XLA` for GPU acceleration, and supports high-order gradients via `autodiff`. This fully differentiable framework enables gradient-enhanced RL within HydroGym.

The objective of this environment is to maximize the turbulent kinetic energy (TKE) of the system using minimal actions. The average TKE is defined as $\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}[\frac{1}{2}(u_i - U)^2 + (v_i - V)^2]$, where (u, v) are the x and y velocity components with mean fields of U and V, respectively. To maximize the TKE, the actions are formulated as amplitudes of a forcing wavenumber. Four wavenumbers were chosen to manipulate the system. The observation space consists of the velocity magnitude at eight randomly distributed points in the domain. The standard reward function is formulated as $r_{\text{no grad}} = \alpha TKE - \beta||a||^2$ where a represents the action and α and β are adjustable parameters to balance the weight of each component. In this case, we want to maximize the TKE while penalizing actions with large magnitudes. The significant advantage of differentiability will be demonstrated in the results section, where a policy trained with a gradient-informed reward $r_{\text{with grad}} = \alpha TKE - \beta||a||^2 + \gamma \nabla_a TKE \cdot a$ will be compared to the policy described above.

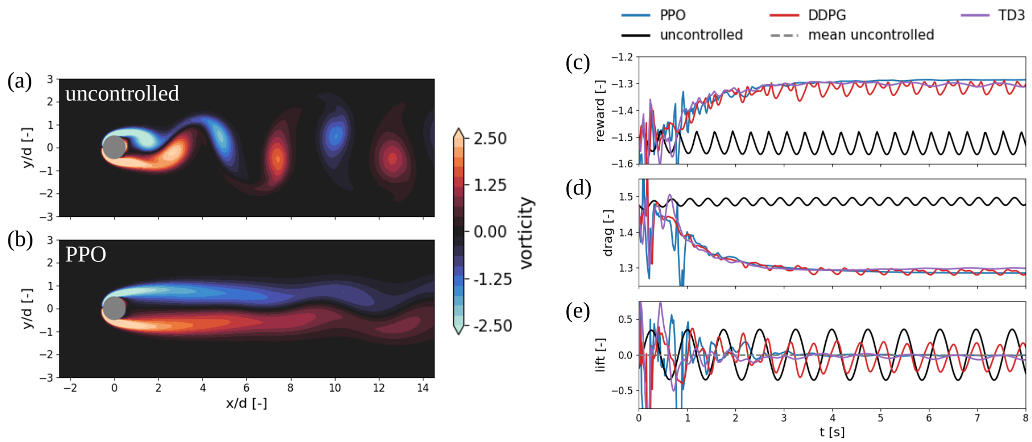

Figure 4: Exemplary training results of the rotational cylinder environment at $Re = 100$. Instantaneous flow fields of an uncontrolled (a) and a controlled (b) environment; (c) - (e) reward, drag and lift forces comparison for all benchmarked algorithms.

4. Results

In the following, we briefly outline benchmark training results to illustrate the working principle and properties of the proposed HydroGym platform. However, due to the immense physical parameter space covered by HydroGym, e.g., various Reynolds numbers, actuation strategies, reward formulations, and flow quantities for observation probing, we can only discuss a few selected test cases in this first paper. These results include on- and off-policy RL methods, e.g., PPO, DDPG, and TD3, trained in the rotational cylinder environment at $Re = 100$, the fluidic pinball environment at $Re = 100$, and the open cavity flow environment at $Re = 7500$ using default implementations of TorchRL (Bou et al., 2024). Moreover, we also present example benchmark results for the fully differentiable Kolmogorov flow environment at $Re = 250$. Here, we use a default PPO implementation of PureJaxRL (Lu et al., 2022) to provide an entire RL training pipeline in JAX, resulting in significant training speedups and support for multi-environment training.

4.1. Rotational Cylinder Environment at $\mathrm{Re} = 100$

In the present study, all agents achieve similar results and learn a competitive control policy that stabilizes the wake and accomplishes almost identical episodic rewards (see Fig. 4 (c)). An example of a controlled flow field is given in Fig. 4 (b) illustrating that the RL based control causes a substantial increase of the recirculation area in the wake of the bluff body. As a result, a total drag reduction of approximately 12% can be observed across all agents (see Fig. 4 (d)) which is in line with previous studies reported in the literature. Likewise, the lift fluctuations are damped compared to the non-actuated reference case (see Fig. 4 (e)), with a moderate reduction for DDPG-based methods and a total oscillation attenuation in the case of PPO and TD3 based controllers.

4.2. Pinball Environment at $\mathrm{Re} = 100$

We now briefly discuss the effectiveness of RL-based methods to control the quasi-periodic flow regime of the fluidic pinball at $Re = 100$. As highlighted in Fig. 5, all agents learn a similar control policy to suppress vortex shedding and stabilize the flow field. Precisely, the agents learn a "boat tailing" actuation strategy (Liu et al., 2024) in which the last two cylinders are rotated inwards while the leading cylinder is kept still. This actuation strategy reduces the adverse pressure gradients along the two downstream cylinders. As a result, the streamlines of the flow do not separate from the two trailing cylinders, as visible in Fig. 5 (a), and closely follow the cylinder contour (see

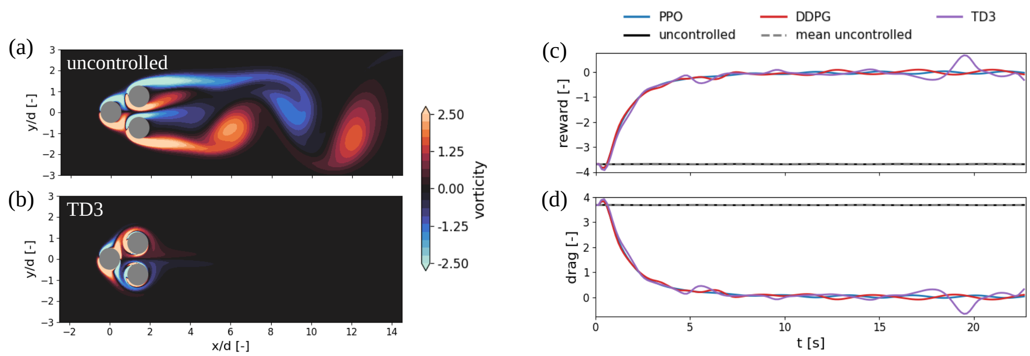

Figure 5: Exemplary training results of the fluidic pinball environment at $Re = 100$. Instantaneous flow fields of an uncontrolled (a) and a controlled (b) environment; (c) - (d) reward and drag force comparison for all benchmarked algorithms.

Fig. 5 (b)) causing a substantially reduced wake region downstream of the cylinders. This new flow topology yields an immense drag reduction of approximately 85% while the lift forces of the individual cylinders balance each other and exhibit barely any oscillations (see Fig. 5 (c) & (d)).

4.3. Shear-Driven Cavity Flow at $\mathrm{Re} = 7500$

In the cavity environment, all agents can stabilize the Kelvin-Helmholtz instability as evidenced by the decreased TKE in Fig. 6 (d) and yield a largely stable and smooth shear layer covering most of the cavity (see Fig. 6 (b) & (c)). However, on-policy methods, specifically PPO agents, are more effective in suppressing the perturbations of the second feedback cycle which recirculates within the cavity. Most likely, this difference originates from the different training approaches underlying on- and off-policy methods: the off-policy methods tested samples from a large, prioritized replay buffer, while the PPO agents trained on samples from an individual episode roll-out. This difference poses significant challenges in the context of the cavity environment since the different time-scales of the physical mechanisms and the spatially disjoint observation locations and actuator positions introduce time-delay effects. As a result, off-policy methods are less effective in this setting as reflected by the lower episodic rewards (see Fig. 6 (d) & (e)).

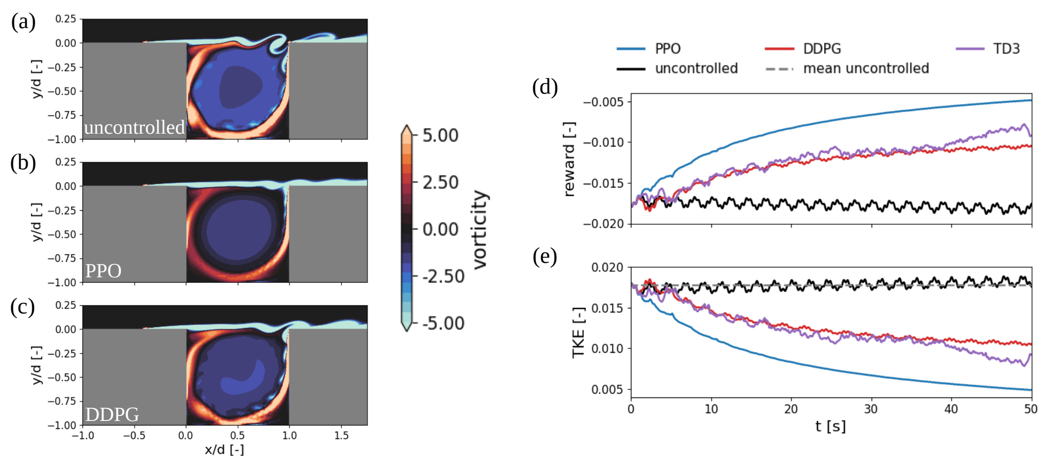

Figure 6: Exemplary training results of the open cavity environment at $Re = 7500$. Instantaneous flow fields of an uncontrolled (a) and a controlled (b) environment; (c) - (d) reward and turbulent kinetic energy (TKE) comparison for all benchmarked algorithms.

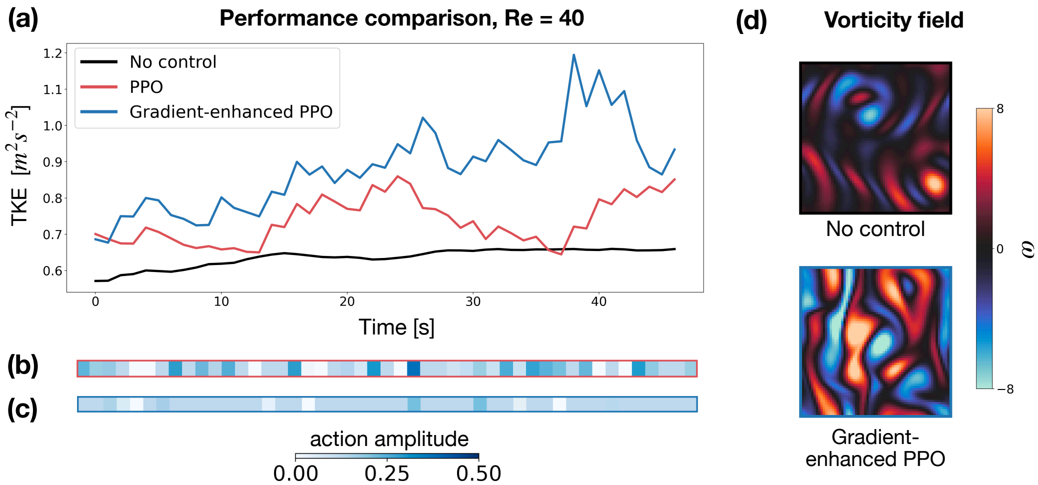

Figure 7: Exemplary training results of the Kolmogorov environment for increasing mixing (TKE). (a) Performance comparison of gradient-enhanced PPO, (b) - (c) efficiency by magnitude of actions, (d) snapshot of the vorticity field at $t = 38$ s for the uncontrolled and gradient-enhanced PPO case.

4.4. Differentiable Kolmogorov Flow at $\mathrm{Re} = 40$

Despite minimal observations, the gradient-enhanced agent was able to learn a control policy which increases mixing as indicated by the consistently increased TKE over time (see Fig. 7 (a)). The resulting flow field exhibits a strong similarity with higher Re-number counterparts, which, by nature, feature stronger turbulent characteristics. Importantly, the agent was also able to learn a control policy that uses sparse actions, in which most amplitudes were close to zero with only one or two amplitudes activated at a time, resulting in an energy efficient control strategy. The inclusion of the gradient term $\nabla_a TKE \cdot a$ was extremely beneficial as it substantially reduces the control energy (Fig. 7 (c)) compared to traditional PPO (Fig. 7 (b)). Considering the entire environment is differentiable, the inclusion of the gradient term steers the policy towards actions that lead to increased TKE while driving less important actions towards zero. It is important to note that several combinations of α and β parameters were tested for traditional PPO, yet none of them achieved the performance of one trial of gradient-enhanced PPO demonstrating its superior performance.

5. Conclusion

In this paper, we introduce HydroGym, a versatile and scalable platform designed to advance the field of reinforcement learning (RL) control in fluid dynamics. HydroGym addresses the significant challenges in fluid flow control by providing a standardized, solver-independent interface that integrates sophisticated flow control benchmarks with state-of-the-art RL algorithms. Our platform supports both non-differentiable and differentiable environments, enabling comprehensive evaluations across various flow phenomena. Through benchmark training results, we demonstrate HydroGym's capability to facilitate the development and testing of RL-based control policies. The results highlight the platform's effectiveness in stabilizing complex flow environments, such as the rotational cylinder, fluidic pinball, shear-driven cavity, and chaotic Kolmogorov flow. HydroGym's scalable design allows for seamless computation from laptops to high-performance computing resources, making it accessible for a wide range of research applications. By bridging the gap between RL and fluid dynamics, HydroGym paves the way for significant advancements in scientific machine learning and practical flow control solutions. Future work will focus on expanding the range of environments to support more complex and realistic flow control scenarios.

Acknowledgments

This research was funded by the German Research Foundation within the Walter Benjamin fellowships MA 10764/1-1 (EL) and LA 5508/1-1 (CL). NA acknowledges funding from ERC Advanced Grant Project No. 101094463. We also acknowledge support from the National Science Foundation AI Institute in Dynamic Systems (grant no. 2112085) and the Boeing Company. Furthermore, the authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. for funding this project by providing computing time on the GCS Supercomputers.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Beyond tabula rasa: Reincarnating reinforcement learning. *arXiv preprint arXiv:2206.01626*, 2022.

Martin S. Alnaes, Anders Logg, Kristian B. Oelgaard, Marie E. Rognes, and Garth N. Wells. Unified form language: A domain-specific language for weak formulations of partial differential equations, 2013.

Daniel B. Araya, Tim Colonius, and John O. Dabiri. Transition to bluff-body dynamics in the wake of vertical-axis wind turbines. *Journal of Fluid Mechanics*, 813:346–381, 2017. doi: 10.1017/jfm.2016.862.

H Jane Bae and Petros Koumoutsakos. Scientific multi-agent reinforcement learning for wall-models of turbulent flows. *Nature Communications*, 13(1):1443, 2022.

P.W. Bearman. Circular cylinder wakes and vortex-induced vibrations. *Journal of Fluids and Structures*, 27(5):648–658, 2011. ISSN 0889-9746. doi: https://doi.org/10.1016/j.jfluidstructs.2011.03.021. URL https://www.sciencedirect.com/science/article/pii/S0889974611000600. IUTAM Symposium on Bluff Body Wakes and Vortex-Induced Vibrations (BBVIV-6).

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Luke Bhan, Yuexin Bian, Miroslav Krstic, and Yuanyuan Shi. Pde control gym: A benchmark for data-driven boundary control of partial differential equations. In *6th Annual Learning for Dynamics & Control Conference*, pages 1083–1095. PMLR, 2024.

Katharina Bieker, Sebastian Peitz, Steven L. Brunton, J. Nathan Kutz, and Michael Dellnitz. Deep model predictive flow control with limited sensor data and online learning. *Theoretical and Computational Fluid Dynamics*, 884, 2020. doi: 10.1007/s00162-020-00520-4.

Antoine Blanchard and Themistoklis P. Sapsis. Stabilization of unsteady flows by reduced-order control with optimally time-dependent modes. *Phys. Rev. Fluids*, 4:053902, May 2019. doi: 10.1103/PhysRevFluids.4.053902. URL https://link.aps.org/doi/10.1103/PhysRevFluids.4.053902.

Albert Bou, Matteo Bettini, Sebastian Dittert, Vikash Kumar, Shagun Sodhani, Xiaomeng Yang, Gianni De Fabritiis, and Vincent Moens. Torchrl: A data-driven decision-making library for pytorch. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=QxItoEAVMb.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Michael P Brenner and Petros Koumoutsakos. Machine learning and physical review fluids: An editorial perspective. *Physical Review Fluids*, 6(7):070001, 2021.

MP Brenner, JD Eldredge, and JB Freund. Perspective on machine learning for advancing fluid mechanics. *Physical Review Fluids*, 4(10):100501, 2019.

Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

S. L. Brunton and J. N. Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2nd edition, 2022.

S. L. Brunton and B. R. Noack. Closed-loop turbulence control: Progress and challenges. *Applied Mechanics Reviews*, 67:050801–1–050801–48, 2015.

Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.

Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jake Varley, Alex Irpan, Benjamin Eysenbach, Ryan Julian, Chelsea Finn, et al. Actionable models: Unsupervised offline reinforcement learning of robotic skills. *arXiv preprint arXiv:2104.07749*, 2021.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Gideon Dresdner, Dmitrii Kochkov, Peter Christian Norgaard, Leonardo Zepeda-Nunez, Jamie Smith, Michael Brenner, and Stephan Hoyer. Learning to correct spectral methods for simulating turbulent flows. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=wNBARGxoJn.

Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual Reviews of Fluid Mechanics*, 51:357–377, 2019.

Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Yiqi Feng, Felix S Schranner, Josef Winter, and Nikolaus A Adams. A deep reinforcement learning framework for dynamic optimization of numerical schemes for compressible flow simulations. *Journal of Computational Physics*, 493:112436, 2023.

Davide Forti and Luca Dedè. Semi-implicit bdf time discretization of the navier–stokes equations with vms-les modeling in a high performance computing framework. *Computers & Fluids*, 117: 168–182, 2015.

C Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax–a differentiable physics engine for large scale rigid body simulation. *arXiv preprint arXiv:2106.13281*, 2021.

Christophe Geuzaine and Jean-Francois Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 0:1–24, may 2009. doi: 10.1002/nme.2579. URL https://doi.org/10.1002/nme.2579.

Luca Guastoni, Jean Rabault, Philipp Schlatter, Hossein Azizpour, and Ricardo Vinuesa. Deep reinforcement learning for turbulent drag reduction in channel flows. *The European Physical Journal E*, 46(4):27, 2023.

Peter Gunnarson, Ioannis Mandralis, Guido Novati, Petros Koumoutsakos, and John Dabiri. Deep reinforcement learning for efficient navigation in vortical flow fields. In *APS Division of Fluid Dynamics Meeting Abstracts*, pages R01–018, 2020.

Jayesh K Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized pde modeling. *arXiv preprint arXiv:2209.15616*, 2022.

David A. Ham, Paul H. J. Kelly, Lawrence Mitchell, Colin J. Cotter, Robert C. Kirby, Koki Sagiyama, Nacime Bouziani, Sophia Vorderwuelbecke, Thomas J. Gregory, Jack Betteridge, Daniel R. Shapero, Reuben W. Nixon-Hill, Connor J. Ward, Patrick E. Farrell, Pablo D. Brubeck, India Marsden, Thomas H. Gibson, Miklós Homolya, Tianjiao Sun, Andrew T. T. McRae, Fabio Luporini, Alastair Gregory, Michael Lange, Simon W. Funke, Florian Rathgeber, Gheorghe-Teodor Bercea, and Graham R. Markall. *Firedrake User Manual*. Imperial College London and University of Oxford and Baylor University and University of Washington, first edition edition, 5 2023.

Taylor A Howell, Simon Le Cleac'h, Jan Brüdigam, J Zico Kolter, Mac Schwager, and Zachary Manchester. Dojo: A differentiable physics engine for robotics. *arXiv preprint arXiv:2203.00806*, 2022.

Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.

Esther Lagemann, Marian Albers, Christian Lagemann, and Wolfgang Schröder. Impact of Reynolds number on the drag reduction mechanism of spanwise travelling surface waves. *Flow, Turbulence and Combustion*, pages 1–14, 2023a.

Esther Lagemann, Steven L Brunton, and Christian Lagemann. Uncovering wall-shear stress dynamics from neural-network enhanced fluid flow measurements. *Proceedings of the Royal Society A*, 480(2292):20230798, 2024a.

Esther Lagemann, Steven L Brunton, Wolfgang Schröder, and Christian Lagemann. Towards extending the aircraft flight envelope by mitigating transonic airfoil buffet. *Nature Communications*, 15(1):5020, 2024b.

Kai Lagemann, Christian Lagemann, and Sach Mukherjee. Invariance-based learning of latent dynamics. In *The Twelfth International Conference on Learning Representations*, 2023b.

Robert Tjarko Lange. gymnax: A JAX-based reinforcement learning environment library, 2022. URL http://github.com/RobertTLange/gymnax.

Yanting Liu, Nan Deng, Bernd R Noack, and Xin Wang. The fluidic pinball with symmetric forcing displays steady, periodic, quasi-periodic, and chaotic dynamics. *Theoretical and Computational Fluid Dynamics*, pages 1–21, 2024.

Chris Lu, Jakub Kuba, Alistair Letcher, Luke Metz, Christian Schroeder de Witt, and Jakob Foerster. Discovered policy optimisation. *Advances in Neural Information Processing Systems*, 35:16455–16468, 2022.

Isaac D Lutz, Shunzhi Wang, Christoffer Norn, Alexis Courbet, Andrew J Borst, Yan Ting Zhao, Annie Dosey, Longxing Cao, Jinwei Xu, Elizabeth M Leaf, et al. Top-down design of protein architectures with reinforcement learning. *Science*, 380(6642):266–273, 2023.

Guy Y. Cornejo Maceda, Yiqing Li, Francois Lusseyran, Marek Morzyński, and Bernd R. Noack. Stabilization of the fluidic pinball with gradient-enriched machine learning control. *Journal of Fluid Mechanics*, 917, apr 2021. doi: 10.1017/jfm.2021.301. URL https://doi.org/10.1017%2Fjfm.2021.301.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

B. R. Noack, M. Morzynski, and G. Tadmor. *Reduced-order modelling for flow control*, volume 528. Springer Science & Business Media, 2011.

Bernd R. Noack and Marek Morzyński. The fluidic pinball - a toolkit for multiple-input multiple-output flow control (version 1.0). *Technical Report 02/2016*, 2017.

Bernd R. Noack, Konstantin Afanasiev, Marek MorzyŃski, Gilead Tadmor, and Frank Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003. doi: 10.1017/S0022112003006694.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Luc R. Pastur, Nan Deng, Marek Morzyński, and Bernd R. Noack. Reduced-order modeling of the fluidic pinball. In *11th Chaotic Modeling and Simulation International Conference*, pages 205–213. Springer International Publishing, 2019. doi: 10.1007/978-3-030-15297-0_19. URL https://doi.org/10.1007%2F978-3-030-15297-0_19.

Sebastian Peitz, Samuel E. Otto, and Clarence W. Rowley. Data-driven model predictive control using interpolated koopman generators. *SIAM Journal on Applied Dynamical Systems*, 19(3): 2162–2193, jan 2020. doi: 10.1137/20m1325678. URL https://doi.org/10.1137%2F20m1325678.

Charles Pivot, Laurent Cordier, and Lionel Mathelin. A continuous reinforcement learning strategy for closed-loop control in fluid dynamics. In *35th AIAA Applied Aerodynamics Conference*, page 3566, 2017.

Fernando Porté-Agel, Yu-Ting Wu, and Chang-Hung Chen. A numerical study of the effects of wind direction on turbine wakes and power losses in a large wind farm. *Energies*, 6(10):5297–5313, 2013. ISSN 1996-1073. doi: 10.3390/en6105297. URL https://www.mdpi.com/1996-1073/6/10/5297.

Jean Rabault, Miroslav Kuchta, Atle Jensen, Ulysse Réglade, and Nicolas Cerardi. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of Fluid Mechanics*, 865:281–302, 2019. doi: 10.1017/jfm.2019.62.

Oliver T. Schmidt and Peter J. Schmid. A conditional space–time pod formalism for intermittent and rare events: example of acoustic bursts in turbulent jets. *Journal of Fluid Mechanics*, 867: R2, 2019. doi: 10.1017/jfm.2019.200.

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

D. Sipp and A. Lebedev. Global stability of base and mean flows: a general approach and its applications to cylinder and open cavity flows. *Journal of Fluid Mechanics*, 593:333–358, 2007.

D. Sipp, O. Marquet, P. Meliga, and A. Barbagallo. Dynamics and control of global instabilities in open-flows: a linearized approach. *Applied Mechanics Reviews*, 63, 2010.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in Neural Information Processing Systems*, 35:1596–1611, 2022.

Russ Tedrake et al. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems, 2014.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

Siddhartha Verma, Guido Novati, and Petros Koumoutsakos. Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proceedings of the National Academy of Sciences*, 115(23):5849–5854, 2018.

Ricardo Vinuesa and Steven L Brunton. Enhancing computational fluid dynamics with machine learning. *Nature Computational Science*, 2(6):358–366, 2022.

Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojtek Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/, 2019.

Qiulei Wang, Lei Yan, Gang Hu, Chao Li, Yiqing Xiao, Hao Xiong, Jean Rabault, and Bernd R Noack. Drlinfluids: An open-source python platform of coupling deep reinforcement learning and openfoam. *Physics of Fluids*, 34(8):081801, 2022.

Zhou Xian, Bo Zhu, Zhenjia Xu, Hsiao-Yu Tung, Antonio Torralba, Katerina Fragkiadaki, and Chuang Gan. Fluidlab: A differentiable environment for benchmarking complex fluid manipulation. *arXiv preprint arXiv:2303.02346*, 2023.

P. K. Yeung, X. M. Zhai, and Katepalli R. Sreenivasan. Extreme events in computational turbulence. *Proceedings of the National Academy of Sciences*, 112(41):12633–12638, 2015. doi: 10.1073/pnas.1517368112. URL https://www.pnas.org/doi/abs/10.1073/pnas.1517368112.

Z. Yin, H.J.H. Clercx, and D.C. Montgomery. An easily implemented task-based parallel scheme for the fourier pseudospectral solver applied to 2d navier–stokes turbulence. *Computers & Fluids*, 33(4):509–520, 2004. ISSN 0045-7930. doi: https://doi.org/10.1016/j.compfluid.2003.06.003. URL https://www.sciencedirect.com/science/article/pii/S004579300300077X.

Xiangyuan Zhang, Weichao Mao, Saviz Mowlavi, Mouhacine Benosman, and Tamer Başar. Controlgym: Large-scale control environments for benchmarking reinforcement learning algorithms. In *6th Annual Learning for Dynamics & Control Conference*, pages 181–196. PMLR, 2024.