A perspective on fluid mechanical environments for challenges in reinforcement learning

Shruti Mishra[1] **Michael Chang**[2] **Vamsi Spandan**[1] **Shmuel M. Rubinstein**[3]
[1]**Sony AI** [2]**Cohere Labs Community** [3]**The Hebrew University of Jerusalem**

Abstract

We consider the challenge of developing agents that efficiently interact with high-dimensional, evolving environments, towards a view of practical reinforcement learning (RL) agents interacting with open worlds, of which they witness and affect only a small part. We argue that canonical fluid mechanics problems, and their simulations, present a compelling testbed for the development of such methods. These problems arise in nonlinear instabilities, where small disturbances can grow to transform the dynamics of a system. Nonlinear instabilities represent several open scientific challenges with industrial applications — the droplet breakup of a liquid jet, mixing at an interface between two fluids, and the appearance of unusually tall rogue waves in the ocean. In these settings, agents may leverage preserved representations across the changing dynamics to learn efficiently.

We present two problem descriptions of agents interacting with a fluid mechanical environment, and describe the state and action spaces, and reward functions, for these agents. For these examples, we specify the aspects of the environment which are nonstationary and the preserved invariances. We note Dedalus and JAX-CFD as open-source simulators that can be used for the development of reinforcement learning methods (Burns et al., 2016; Kochkov et al., 2021). We demonstrate the use of Dedalus for environment generation by creating RL agents that learn to navigate in a stationary environment that is simulated using Dedalus. This sets the stage for future development of RL agents that learn to meaningfully interact with simulated environments that represent scientific challenges in natural and industrial flows.

1 Introduction

Over the last decade, artificial environments such as games have catalyzed the development of some of the most advanced artificial methods for reinforcement learning (Mnih et al., 2015; Silver et al., 2017; 2018; Vinyals et al., 2019; Wurman et al., 2022; Perolat et al., 2022). In advancing reinforcement learning methods towards real-world settings that include the understanding of natural phenomena, a number of challenges remain (Dulac-Arnold et al., 2021). A key challenge is that of agents efficiently interacting with an evolving, high-dimensional, environment, beyond the paradigm of statistically stationary environment interactions, towards the setting of a finite agent learning from interactions in a big world (Javed and Sutton, 2024). Intelligent agents operating in the real world navigate changing environmental dynamics. To do so, agents can leverage knowledge of how the environment is structured to inform meaningful behavior. For instance, animals in urban environments adapt their behaviors to respond to the changes created by human activity — birds have changed the pitch of their song, reducing disruption by the sounds of an urban environment (e.g. Slabbekoorn and den Boer-Visser (2006)), and squirrels have learned to seek ways to obtain food from humans.

In nonlinear instabilities in fluid flows, small disturbances can grow and fundamentally alter the dynamics of a system — a small ripple on a water surface can grow into an unusually large rogue wave and a small fluctuation in an engine can trigger destructive oscillations. The growth of

Author for correspondence: Shruti Mishra; email: shrutimishraphd@gmail.com

small disturbances to a flow can transform pre-existing flow patterns. When disturbances to a flow evolve, flows can transition towards turbulence, a class of canonical multiscale phenomena. With a wide variety of fundamental scientific questions and relevance for industrial applications, hydrodynamic instabilities have naturally been associated with deep scientific activity (Chandrasekhar, 2013). However, nonlinear instabilities are associated with yet unsolved scientific mysteries such as a discrepancy between computational predictions and experimental observations on the growth of a mixing layer at an interface between two fluids. Industrial applications include predicting thermoacoustic oscillations in combustors for energy production and heat flow during boiling in power plants.

Across the dynamics associated with small waves and destructive rogue waves in an ocean, even as energy is transferred across the spatial scales, mathematical representations associated with conservation of mass, momentum and energy are preserved to a good approximation. Agents operating in highly changing fluid flow environments can thus leverage such preserved representations while acting meaningfully across changes in flow structures. Specifically, fluid flows are well modeled by the Navier–Stokes equations. These equations represent conservation of mass and conservation of momentum. With such known parsimonious models, flows have representations that are preserved across critical transitions that are characteristic of instabilities in fluid flows (Chandrasekhar, 2013). With known and learnable representations that can be leveraged, flows with nonlinear instabilities can form a compelling testbed for the development and demonstration of reinforcement learning agents efficiently interacting with evolving, high-dimensional, environments.

Objectives for reinforcement learning agents in flows with nonlinear instabilities can include predicting, meaningfully affecting, and controlling flows with apparently changing dynamics. The proposed paradigm of reinforcement learning in flows with nonlinear instabilities is one of an interplay of apparently changing dynamics and some preserved representations.

2 Stationarity in representation: Invariances in fluid flows

Physical systems are studied using theoretical, computational and experimental approaches. This section presents some representations in fluid flows that remain preserved, even across transient dynamics in the observable field variables such as velocity. The text, “Fluid Mechanics” (Kundu et al., 2012), provides a mathematical introduction to the subject, and “Elementary Fluid Mechanics” (Acheson, 1990) presents a more concise description.

When represented in terms of field variables such as velocity, pressure and density, varying over spatial domain, and evolving over time, fluid flows are naturally high-dimensional. As an example, flows can require thousands of grid points over even millimetre-sized spatial domains with uniform resolution, evolving over milliseconds, for adequate numerical resolution e.g. (Mishra et al., 2022). At the same time, such flows are represented mathematically by a small set of governing equations which determine the evolution of the corresponding flows.

Mathematical models are a powerful tool for capturing relevant physics while abstracting processes that happen at resolutions different from the problem in consideration. Fluid mechanics itself is an example of such an abstraction; interactions and collisions at a molecular level are coarse-grained and expressed in terms of equations for the conservation of mass, momentum and energy for a continuum – the fluid.

The Navier–Stokes equations, indicated for incompressible flows by Equation 1,

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \qquad \text{conservation of mass,} \tag{1a}$$

$$\boldsymbol{u}_t + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\, \boldsymbol{u} = -\frac{\boldsymbol{\nabla} p}{\rho} + \nu \nabla^2 \boldsymbol{u}, \qquad \text{conservation of momentum,} \tag{1b}$$

are a powerful representation of fluid mechanical systems that represent the conservation of mass and momentum, where $\boldsymbol{u}$ is the fluid velocity, p is pressure and ρ is density. An ubiquitous abstraction within the framework of the Navier–Stokes equations is the quantity *viscosity*, often expressed as a constant for Newtonian fluids i.e. ν in Equation 1b. Viscosity represents the macroscopic effect of mixing through movement of particles at a molecular level in terms of the smoothing of gradients of flow-field variables such as velocity. Across several flows with transient flow structures, such as the rapidly changing flow structures around a racecar, the Navier–Stokes equations with a constant viscosity of air remain a good approximation of the flow dynamics.

Abstractions of various types have been used to further enable the solution of these equations for particular problems. For instance, using knowledge of the relevant physical processes to remove terms that are deemed to be small compared to other terms, is done to enable tractability of the problem. These conservation relationships can remain preserved across the relevant dynamics, and arguably provide a world model for intelligent agents.

Computational methods to solve the Navier–Stokes equations rely on further abstractions of flow-field quantities. For instance, finite-difference methods assume that the continuous quantities can be represented by discretized counterparts that vary on a grid. Spectral methods represent the underlying field variables in terms of periodic functions over a domain of interest. Finite-difference methods and spectral methods are two broad categories of a rich set of scientifically validated representations of the Navier–Stokes equations.

3 Nonstationary environments with stationary representations

Having noted a variety of stationary representations that arise in fluid flows in Section 2, this section describes examples of reinforcement learning agents interacting with nonstationary fluid environments which also have known stationary representations.

3.1 An agent in a transitioning pipe flow

This section describes the setup of a reinforcement learning agent interacting with a flow in a pipe that transitions towards turbulence through the growth and nonlinear interactions of perturbations to a mean flow profile. Turbulence is a canonical multiscale phenomenon; interactions in a turbulent flow range from the largest length scale in the flow, dominated by inertial interactions, to the smallest scale, dominated by viscous interactions, e.g. (McKeown et al., 2020). An agent that takes the form of an active particle in a turbulent flow, buffeted by flow structures, experiences interactions with flow structures across these length scales, typically spanning multiple orders of magnitude. Its world is big (Javed and Sutton, 2024).

The flow of water in a pipe is well described by the Navier–Stokes equations for conservation of mass and momentum (Equation 1). The geometry of a three-dimensional pipe flow comprises a circular cross section of radius R, uniform along the length of the pipe, and the flow is driven by a pressure gradient along the length of the pipe. With boundary conditions $\boldsymbol{u} = \boldsymbol{0}$ at the pipe walls, $r = R$, the mean flow profile for a fully developed pipe flow, downstream of the inlet, is

$$\boldsymbol{u} = u\boldsymbol{e}_z = U_0 \left(1 - (r/R)^2\right) \boldsymbol{e}_z, \tag{2}$$

where $\boldsymbol{e}_z$ is the unit vector along the length of the pipe and U_0 is a constant.

Pipe flow is a canonical flow that is stable to infinitesimal disturbances, but can transition to turbulence via the growth and nonlinear interactions of disturbances to the parabolic mean flow profile (Reynolds, 1883). This is illustrated in Figure 1. In this environment, the observations for the agent are a three-dimensional representation for the velocity field and the pressure field; $\mathcal{S} \equiv [\boldsymbol{u}, p]$, over a distance between $z = 0$ and $z = L$. The action chosen by the agent is equivalent to injecting and moving obstacles in the flow that disturb the uniform cross section of the flow. For simplicity, the agent can move the obstacle at a fixed speed $|\boldsymbol{u}_{\text{obstacle}}|$ and can change the direction of movement of the obstacle at a rate of $\dot{\theta}$, chosen by the agent i.e. $\mathcal{A} \equiv \dot{\theta}$. At the locations $\boldsymbol{x}$ occupied by the obstacle, the flow velocity $\boldsymbol{u} = \boldsymbol{0}$. This creates a perturbation to the mean profile specified in Equation 2. The obstacle can be specified as a small, rigid sphere of diameter $\delta < R$. The agent's goal is to

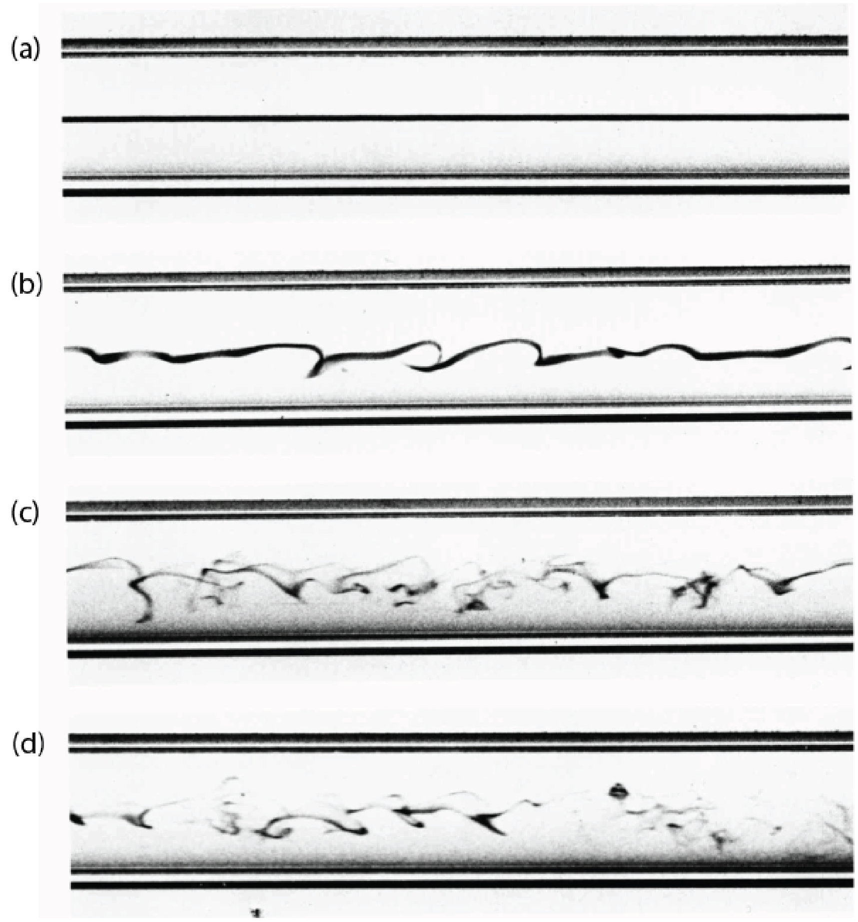

Figure 1: Four snapshots, from top to bottom, illustrating the evolution of a filament of colored water in a pipe flow (Van Dyke, 1982). In all panels, the colored water is indicated in the darker color. (a) The colored water is undisturbed, and the filament occupies a region in the middle of the pipe. (b–d) As the speed of the flow is increased, the filament becomes increasingly unstable, and transitions towards a disordered state. The transition towards such disordered states occurs via the nonlinear interactions of disturbances with each other and with the background flow in the pipe. The book, "An Album of Fluid Motion" by Van Dyke (1982) is a pictographic guide to the diversity of flow patterns observed in nature and in engineering applications.

create the maximum possible disturbance to the mean profile. Accordingly, the reward r can be $\|\boldsymbol{u} - U_0\left(1-(r/R)^2\right)\boldsymbol{e}_z\|_2$ in a specified time horizon T. Through being incentivized to create disturbances to the flow, the RL agent can create an environment that is highly nonstationary at the level of the state description $\mathcal{S}$ e.g. (Avila et al., 2011). Even as the agent can aim to do so, the description of the flow remains invariant at the level of the governing Navier–Stokes equations (1).

3.2 An agent in an evolving cellular flow

Reinforcement learning studies have considered particle swimmers in steady fluid flows i.e. with velocity and pressure constant in time, with a limited state space (Colabrese et al., 2017; Gustavsson et al., 2017). The strength of the background flow field affects the optimal strategy for a swimmer (Colabrese et al., 2017). The setup of a reinforcement learning agent in a nonstationary environment can consist of a particle in an evolving cellular flow, such as that illustrated in Figure 2. An example set

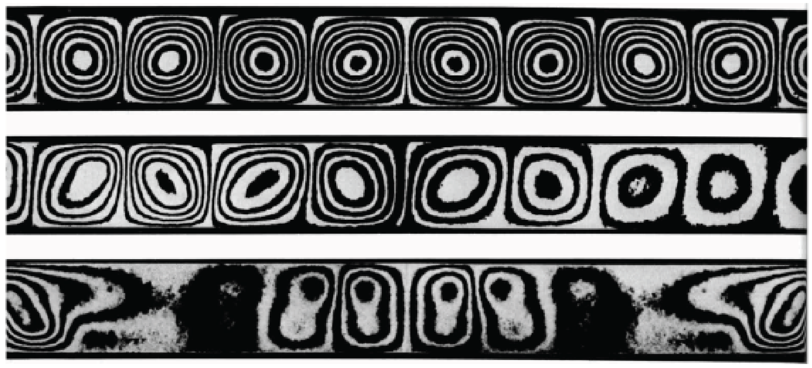

Figure 2: Three snapshots, (a–c), of fluid flow between two plates (Van Dyke, 1982). In each panel, the bottom plate is heated. Under small disturbances to a static fluid between the plates, a flow evolves into convection rolls. In panel (a) the plates are maintained at fixed temperatures. In panel (b), there is a temperature gradient in the temperature of the bottom plate, along the length of the plate. In panel (c), the plate and fluid is subject to an external rotational motion.

up for a two dimensional flow is described by

$$\omega = -2U_0 \cos(x) \cos(y) \exp\left(-2\nu t\right), \tag{3}$$

where $\omega = \boldsymbol{\nabla} \times \boldsymbol{u}$. This flow is a solution to Equation 1 for the initial condition $\omega(x, y, t = 0) \equiv -2U_0 \cos(x) \cos(y)$ in an infinite two-dimensional space. In this flow, the state space $\mathcal{S} \equiv [\omega, \theta]$, where θ is the orientation of a particle in the flow. The action space $\mathcal{A} \equiv \dot{\theta}$ is the change in orientation of the particle. The reward $R \equiv y(s') - y(s)$ defines the change in position between subsequent states, in the preferred direction of movement along y. This setup is similar to that of Colabrese et al. (2017), who used reinforcement learning to train particle swimmers moving through a stationary flow pattern described by $\omega = -2U_0 \cos(x) \cos(y)$.

The time dependence in the flow described by Equation 3, through the term $e^{-\nu t}$, means that the flow, and thus the environment experienced by a particle swimmer through the flow, is inherently nonstationary. In this flow, preserved invariances include the governing Navier–Stokes equations (1), a periodic structure in the flow, and the material property viscosity. As indicated in Figure 3, the learned strategy for a swimmer navigating through this structure changes as the strength of a stationary flow relative to the swimmer strength changes. As such, a swimmer navigating an evolving flow can leverage the evolving structure to determine new policies for traversal.

3.3 Flow control in the literature

Flow control using RL has been considered in prior literature, largely towards achieving a statistically stationary regime. Beintema et al. (2020) consider flow control in Rayleigh–Bénard convection, a classical system for studying thermal convection, a model system for cloud dynamics (Kundu et al., 2012; Drazin and Reid, 2004; Bodenschatz et al., 2000) and the chosen system for the development of several supervised learning algorithms. In the Kuramoto–Sivashinsky equations, a classical system for spatiotemporal chaos, Weissenbacher et al. (2025) consider flow control towards mitigating chaos, and creating a statistically stationary regime.

4 Simulators for reinforcement learning in fluid flows

Virtual environments created by simulators have enabled the advancement and demonstration of reinforcement learning algorithms (Barth-Maron et al., 2018; Haarnoja et al., 2018; Hafner et al., 2019). Examples include the MuJoCo physics engine and Isaac Gym, which form testbeds for bio-inspired and robotics domains (Todorov et al., 2012; Makoviychuk et al., 2021). In order for

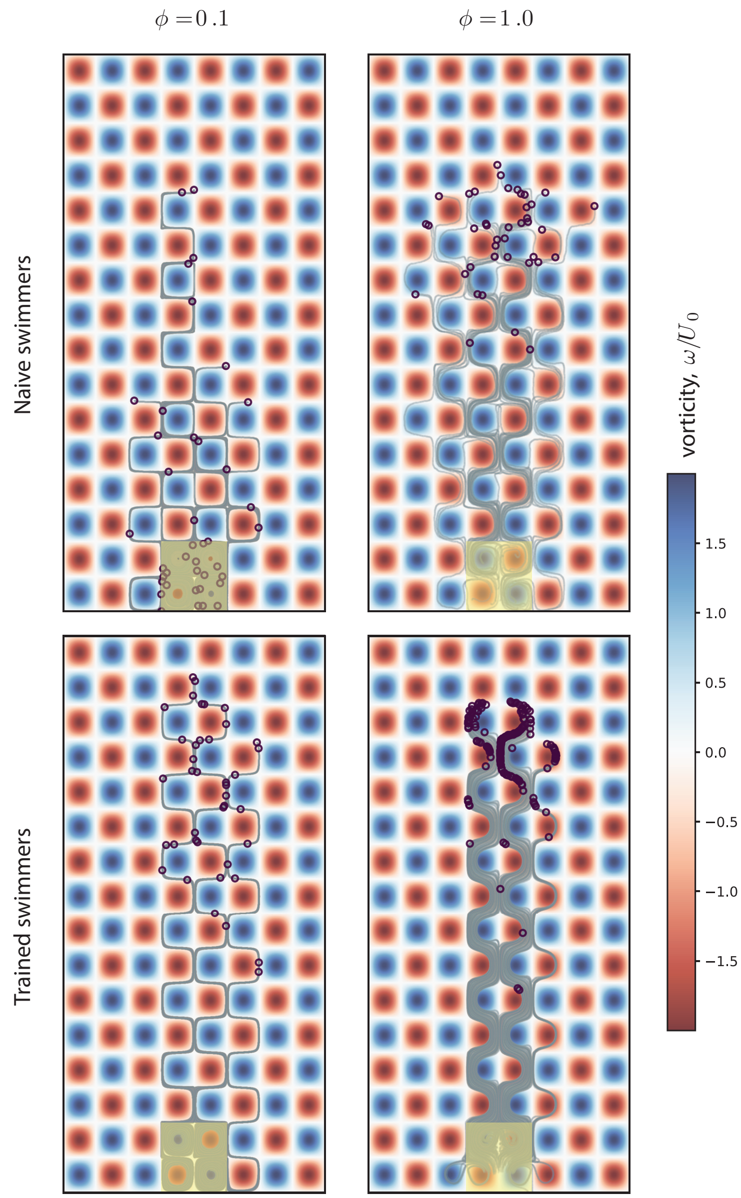

Figure 3: Swimmers navigating through a steady (unchanging) vortex field. The strength of the vorticity field is shown in the red–blue color scheme. The left column shows swimmers that are weaker than the surrounding flow, and thus have to learn to navigate the structure of the background flow. The right column illustrates swimmers that have a strength comparable to that of the background flow. The strength of the swimmers relative to the background flow is specified by the parameter ϕ. The grey lines in the panels indicate the trajectories of 40 swimmers, initialized in the region indicated by the yellow square. The dark purple circles indicate final positions attained by the swimmers in a fixed time duration. The top row shows naïve swimmers which always attempt to orient themselves upwards. The bottom row shows swimmers trained using tabular Q-learning. This demonstrates the training setup of Colabrese et al. (2017), and shows strategies (right panel) for navigating steady flows of different strengths. We set up the flow using Dedalus (Burns et al., 2016). Simulation and training parameters are specified in Table 1.

flows with nonlinear instabilities to be adopted as an interesting domain for advancing reinforcement learning methods, the adoption of a relevant simulator may hold similar significance. This section presents two simulators that may be used for the development of reinforcement learning algorithms in evolving fluid flows, followed by a potential direction for the development of simulators.

4.1 Open-source simulators: Dedalus and JAX-CFD

Dedalus is a general-purpose simulator for differential equations (Burns et al., 2016). Analogous to MuJoCo leveraging approximations to allow for simulations to be practically feasible, the number of spectral coefficients in Dedalus can be chosen for speed versus fidelity. Dedalus has a pre-existing set of examples of flows with nonlinear instabilities. In machine learning, Dedalus has been used as the simulator for superresolution in Rayleigh–Bénard convection, which is a model system for cloud dynamics (Jiang et al., 2020).

As an example, we replicate the environment of Colabrese et al. (2017), and demonstrates the use of Dedalus to create the environment of a cellular flow pattern described in Section 3.2, and illustrated in Figure 2. Briefly, an agent is in a flow described by a stationary vorticity field,

$$\omega = -2U_0 \cos(x) \cos(y). \tag{4}$$

Similar to the setup of Colabrese et al. (2017), a s wimmer in the two dimensional vorticity field has a fixed speed and can choose to change its direction vector. The goal of the agent is to traverse upwards, with the state and action spaces, and reward function specified in Section 3.2. For the observation, the vorticity is discretized into three values and the direction of the swimmer is discretized into four values, $\{\uparrow, \rightarrow, \leftarrow, \downarrow\}$. The action for the preferred swimming direction is similarly discretized into four values. Figure 3 shows trajectories for swimmers for increasing amounts of training, illustrating results similar to those of Colabrese et al. (2017), with the environment set up in Dedalus.

JAX-CFD is a GPU simulator for fluid mechanics, designed for coupling with machine learning methods (Kochkov et al., 2021). JAX-CFD has largely been used with supervised learning methods, and more recently for problems in flow control (Alhashim et al., 2025).

4.2 Simulators based on learned surrogate models

The speed of a simulator is a limiting factor for the development of reinforcement learning methods. Surrogate models have been an area of development in the machine learning community in recent years. These models aim to learn key physics during training time, to allow for relatively cheap inference compared to traditional simulations of fluid mechanics (Fonda et al., 2019; Jiang et al., 2020; Pfaff et al., 2021; Lusch et al., 2018; Sanchez-Gonzalez et al., 2020; Wiewel et al., 2019; Wu et al., 2020). Surrogate models have the potential to provide for simulators that can allow for fast computation of the environment dynamics in the reinforcement learning setting.

5 Summary and future directions

The perspective in this paper presents scientific challenges associated with nonlinear instabilities in fluid mechanics as a source of problem formulations for reinforcement learning algorithms, particularly for learning efficiently nonstationary environments. Nonlinear instabilities result in highly changing dynamics, with preserved invariances in the form of conservation equations and potentially in the form of material properties such as viscosity. These invariances can be leveraged by agents towards developing agents that can meaningfully interact with changing environments. We described two examples of nonlinear instabilities. Both the examples are modeled by the Navier–Stokes equations, a well established representation for fluid mechanical systems. For the instabilities, we specified the state space, action space and a reward function for a reinforcement learning agent interacting with the instabilities.

Simulated environments have been instrumental in the development of artificial reinforcement learning methods. We described two simulators, Dedalus and JAX-CFD, as avenues for creating environments for RL agents that interact with nonstationary environments. For a stationary environment related to the nonstationary cellular flow described in Section 3.2, we reproduced results from the literature using Dedalus to create the environment. Relatedly, we noted surrogate models as a source of potential environments for RL agents in the future.

The description of nonlinear instabilities with preserved invariances, together with a demonstration of Dedalus to create environments for RL agents, sets the stage for the integration of RL agents that can learn to leverage known invariances to act efficiently in evolving environments. Future directions can incorporate the environment capabilities of a simulator like Dedalus, to integrate it with nonstationary environments such as those created in nonlinear instabilities. While we have described instabilities in fluid flows, with remarkably preserved invariances in the form of the Navier–Stokes equations, as a source of problem descriptions for advancing reinforcement learning methods, instabilities exist beyond fluid flows. For example, there are instabilities in the dynamics of solid objects. Dedalus includes the capability to simulate some of these dynamics, such the bending of an elastic sheet.

Acknowledgements

S.M. thanks Yannis Hardalupas and Miles Cranmer for discussions of fluid instabilities, Daniel Fortunato for a discussion on Dedalus, Faron Hesse for a discussion of Figure 3, and Jordan Hoffmann for reading drafts of this paper. S.M. acknowledges partial funding support from CQDM under the FACS/Acquité project and a G-Research Grant for Early Career Researchers. We thank the reviewer for feedback on an earlier version of this paper.

References

Acheson, D. J. (1990), *Elementary fluid dynamics*, Oxford University Press.

Alhashim, M. G., Hausknecht, K. and Brenner, M. P. (2025), 'Control of flow behavior in complex fluids using automatic differentiation', *Proceedings of the National Academy of Sciences* **122**(8), e2403644122.

Avila, K., Moxey, D., De Lozar, A., Avila, M., Barkley, D. and Hof, B. (2011), 'The onset of turbulence in pipe flow', *Science* **333**(6039), 192–196.

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., Dhruva, T., Muldal, A., Heess, N. and Lillicrap, T. (2018), Distributed distributional deterministic policy gradients, *in* 'International Conference on Learning Representations'.

Beintema, G., Corbetta, A., Biferale, L. and Toschi, F. (2020), 'Controlling Rayleigh–Bénard convection via reinforcement learning', *Journal of Turbulence* **21**(9-10), 585–605.

Bodenschatz, E., Pesch, W. and Ahlers, G. (2000), 'Recent developments in Rayleigh–Bénard convection', *Annual review of fluid mechanics* **32**(1), 709–778.

Burns, K. J., Vasil, G. M., Oishi, J. S., Lecoanet, D. and Brown, B. (2016), 'Dedalus: Flexible framework for spectrally solving differential equations', *Astrophysics Source Code Library* pp. ascl–1603.

Chandrasekhar, S. (2013), *Hydrodynamic and Hydromagnetic Stability*, Courier Corporation.

Colabrese, S., Gustavsson, K., Celani, A. and Biferale, L. (2017), 'Flow navigation by smart microswimmers via reinforcement learning', *Physical review letters* **118**(15), 158004.

Drazin, P. G. and Reid, W. H. (2004), *Hydrodynamic stability*, 2 edn, Cambridge university press.

Dulac-Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S. and Hester, T. (2021), 'Challenges of real-world reinforcement learning: definitions, benchmarks and analysis', *Machine Learning* **110**(9), 2419–2468.

Fonda, E., Pandey, A., Schumacher, J. and Sreenivasan, K. R. (2019), 'Deep learning in turbulent convection networks', *Proceedings of the National Academy of Sciences* **116**(18), 8667–8672.

Gustavsson, K., Biferale, L., Celani, A. and Colabrese, S. (2017), 'Finding efficient swimming strategies in a three-dimensional chaotic flow by reinforcement learning', *The European Physical Journal E* **40**, 1–6.

Haarnoja, T., Zhou, A., Abbeel, P. and Levine, S. (2018), Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, *in* 'International conference on machine learning', Pmlr, pp. 1861–1870.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H. and Davidson, J. (2019), Learning latent dynamics for planning from pixels, *in* 'International conference on machine learning', PMLR, pp. 2555–2565.

Javed, K. and Sutton, R. S. (2024), The big world hypothesis and its ramifications for artificial intelligence, *in* 'Finding the Frame: An RLC Workshop for Examining Conceptual Frameworks'.

Jiang, C. M., Esmaeilzadeh, S., Azizzadenesheli, K., Kashinath, K., Mustafa, M., Tchelepi, H. A., Marcus, P., Prabhat, Anandkumar, A. et al. (2020), Meshfreeflownet: a physics-constrained deep continuous space-time super-resolution framework, *in* 'SC20: International Conference for High Performance Computing, Networking, Storage and Analysis', IEEE, pp. 1–15.

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P. and Hoyer, S. (2021), 'Machine learning–accelerated computational fluid dynamics', *Proceedings of the National Academy of Sciences* **118**(21), e2101784118.

Kundu, P. K., Cohen, I. M. and Dowling, D. R. (2012), *Fluid Mechanics*, 5 edn, Elsevier.

Lusch, B., Kutz, J. N. and Brunton, S. L. (2018), 'Deep learning for universal linear embeddings of nonlinear dynamics', *Nature communications* **9**(1), 1–10.

Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A. and Slate, G. (2021), 'Isaac gym: High performance gpu-based physics simulation for robot learning', *arXiv preprint arXiv:2108.10470* .

McKeown, R., Ostilla-Mónico, R., Pumir, A., Brenner, M. P. and Rubinstein, S. M. (2020), 'Turbulence generation through an iterative cascade of the elliptical instability', *Science advances* **6**(9), eaaz2717.

Mishra, S., Rubinstein, S. M. and Rycroft, C. H. (2022), 'Computing the viscous effect in early-time drop impact dynamics', *Journal of Fluid Mechanics* **945**, A13.

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

Perolat, J., de Vylder, B., Hennes, D., Tarassov, E., Strub, F., de Boer, V., Muller, P., Connor, J. T., Burch, N., Anthony, T., McAleer, S., Elie, R., Cen, S. H., Wang, Z., Gruslys, A., Malysheva, A., Khan, M., Ozair, S., Timbers, F., Pohlen, T., Eccles, T., Rowland, M., Lanctot, M., Lespiau, J.-B., Piot, B., Omidshafiei, S., Lockhart, E., Sifre, L., Beauguerlange, N., Munos, R., Silver, D., Singh, S., Hassabis, D. and Tuyls, K. (2022), 'Mastering the game of Stratego with model-free multiagent reinforcement learning', *Science* **378**(6623), 990–996.

Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A. and Battaglia, P. W. (2021), Learning mesh-based simulation with graph networks, *in* ‘International Conference on Learning Representations’.

Reynolds, O. (1883), ‘Xxix. an experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels’, *Philosophical Transactions of the Royal society of London* (174), 935–982.

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J. and Battaglia, P. (2020), Learning to simulate complex physics with graph networks, *in* ‘International Conference on Machine Learning’, PMLR, pp. 8459–8468.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K. and Hassabis, D. (2018), ‘A general reinforcement learning algorithm that masters chess, shogi, and go through self-play’, *Science* **362**(6419), 1140–1144.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T. and Hassabis, D. (2017), ‘Mastering the game of go without human knowledge’, *Nature* **550**(7676), 354–359.

Slabbekoorn, H. and den Boer-Visser, A. (2006), ‘Cities change the songs of birds’, *Current biology* **16**(23), 2326–2331.

Todorov, E., Erez, T. and Tassa, Y. (2012), Mujoco: A physics engine for model-based control, *in* ‘2012 IEEE/RSJ international conference on intelligent robots and systems’, IEEE, pp. 5026–5033.

Van Dyke, M. (1982), *An album of fluid motion*, Vol. 176, Parabolic Press Stanford.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P. et al. (2019), ‘Grandmaster level in StarCraft II using multi-agent reinforcement learning’, *nature* **575**(7782), 350–354.

Weissenbacher, M., Borovykh, A. and Rigas, G. (2025), ‘Reinforcement learning of chaotic systems control in partially observable environments’, *Flow, Turbulence and Combustion* pp. 1–22.

Wiewel, S., Becher, M. and Thuerey, N. (2019), Latent space physics: Towards learning the temporal evolution of fluid flow, *in* ‘Computer graphics forum’, Vol. 38, Wiley Online Library, pp. 71–82.

Wu, J.-L., Kashinath, K., Albert, A., Chirila, D., Prabhat and Xiao, H. (2020), ‘Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems’, *Journal of Computational Physics* **406**, 109209.

Wurman, P. R., Barrett, S., Kawamoto, K., MacGlashan, J., Subramanian, K., Walsh, T. J., Capobianco, R., Devlic, A., Eckert, F., Fuchs, F., Gilpin, L., Khandelwal, P., Kompella, V., Lin, H., MacAlpine, P., Oller, D., Seno, T., Sherstan, C., Thomure, M. D., Aghabozorgi, H., Barrett, L., Douglas, R., Whitehead, D., Dürr, P., Stone, P., Spranger, M. and Kitano, H. (2022), ‘Outracing champion Gran Turismo drivers with deep reinforcement learning’, *Nature* **602**(7896), 223–228.

6 Appendix: Parameters for simulation and training

Table 1: Parameter settings for simulation and training

Parameter	**Value(s)**
Simulation parameters for Dedalus	
Domain size, $L \equiv L_x \equiv L_y$	4π
Resolution, $N \equiv N_x \equiv N_y$	512
Number of vortices in simulation domain	4
Reference velocity, U_0 (Equation 3)	1
Algorithm parameters for Q-learning	
Learning rate, α	0.1
Discount factor, γ	0.99
Number of episodes	2000
Initial fraction of random actions, $\epsilon_{\text{initial}}$	1
Final fraction of random actions at 1400 episodes, ϵ_{final}	1×10^{-2}