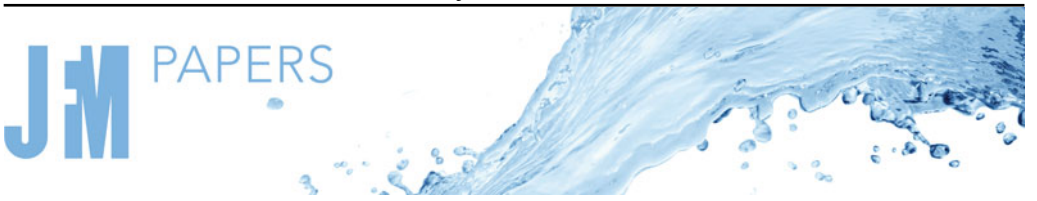
J. Fluid Mech. (2026), *vol.* 1040, A3, doi:10.1017/jfm.2026.11837



Flow separation control of an infinite wing section via multi-agent reinforcement learning

Ricard Montalà[1], Bernat Font[2], Pol Suárez[3], Jean Rabault[4], Oriol Lehmkuhl[5], Ricardo Vinuesa[6,3] and Ivette Rodriguez[1]

[1]TUAREG, Universitat Politècnica de Catalunya - BarcelonaTech (UPC), Terrassa, Spain

[2]Mechanical Engineering, Delft University of Technology (TU Delft), Delft, Netherlands

[3]FLOW, Engineering Mechanics, KTH Royal Institute of Technology, Stockholm, Sweden

[4]Independent Researcher, Oslo, Norway

[5]CASE, Barcelona Supercomputing Center (BSC), Barcelona, Spain

[6]Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA

Corresponding author: Ricard Montalà, ricard.montala@upc.edu

(Received 8 November 2025 UTC; revised 14 June 2026 UTC; accepted 21 June 2026 UTC)

This study investigates the application of deep reinforcement learning (DRL) for active flow control in a three-dimensional NACA0012 wing section with periodic spanwise boundary conditions at low Reynolds number ($Re_c = 1000$) and high angle of attack (AoA $= 20°$), where the flow exhibits massive separation, strong vortex shedding and a chaotic three-dimensional wake. The baseline configuration is validated against previous numerical studies, showing the accuracy of the computational set-up. Two DRL control policies are trained with different reward functions, leading to fundamentally different control strategies. The DRL set-up uses a multi-agent reinforcement learning framework to coordinate distributed actuators along the wing span. In the first case study, when the reward prioritises drag reduction, agents delay the leading-edge separation and stabilise the wake, reducing drag by 21.1 % and lift fluctuations by 54.1 % while maintaining baseline lift. In the second case, when the reward targets lift enhancement instead, agents synchronise actuations with vortex shedding, forming leading-edge instabilities that increase lift by 42.3 %, albeit at the cost of higher fluctuations and slightly increased drag. A comparison with classical open-loop control further demonstrates that DRL autonomously determines physically meaningful and interpretable strategies consistent with established mechanisms in the literature. To the best of the authors' knowledge, this is the first study in which a fully three-dimensional computational fluid dynamics simulation

© The Author(s), 2026. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution-ShareAlike licence (https://creativecommons.org/licenses/by-sa/4.0/), which permits re-use, distribution, and reproduction in any medium, provided the same Creative Commons licence is used to distribute the re-used or adapted article and the original article is properly cited.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

has been employed to train a DRL model for active flow control in wings, suggesting promising directions for extending DRL-based strategies to higher Reynolds numbers and more complex wing configurations where prior physical knowledge may be limited.

Key words: flow control, aerodynamics, machine learning

1. Introduction

In recent years, deep reinforcement learning (DRL) has emerged as a promising data-driven technique for handling the chaotic dynamics of fluid flow systems. Deep reinforcement learning has shown potential across a wide range of applications in fluid mechanics, including autonomous navigation (Tonti *et al.* 2025; Xiong *et al.* 2025), wall-modelled large-eddy simulations (Bae & Koumoutsakos 2022; Zhou & Bae 2024), subgrid-scale modelling (Novati *et al.* 2021; Kurz *et al.* 2023), Reynolds-averaged Navier–Stokes modelling (Fuchs *et al.* 2024), flow control (Verma *et al.* 2018; Guastoni *et al.* 2023) and shape optimisation (Li *et al.* 2021; Viquerat *et al.* 2021).

Owing to its intrinsic decision-making capabilities, DRL is particularly well suited for control problems, especially for active flow-control (AFC) applications. Deep reinforcement learning can discover and apply effective closed-loop control laws in dynamically evolving environments, learning sophisticated strategies that often outperform traditional methods relying on fixed actuation frequencies or predefined laws (Rodriguez *et al.* 2020; Sun *et al.* 2025). Therefore, DRL-AFC systems offer a promising approach for safer and more sustainable transport technologies.

In § 1.1, we first review the traditional AFC methods commonly applied to wings. Section 1.2 then outlines the DRL-based AFC studies present in the literature. Section 1.3 examines the complexities of the baseline problem considered in this study. Finally, the contributions of the present work are summarised in § 1.4.

1.1. *Active flow control in wings*

Among AFC techniques, steady uniform suction and blowing, as well as periodic suction and blowing, have been widely employed. In these control strategies, the actuation intensity is often characterised by the momentum coefficient $C_\mu = (\rho S_{jet} U_{jet}^2)/(\rho S_{ref} U_\infty^2)$; where ρ is the fluid density, S_{jet} the jet control surface, S_{ref} the characteristic reference area, U_{jet} the jet velocity and U_∞ the free-stream velocity. Here, the reference area is taken as the airfoil chord c times the corresponding span width L_z.

In steady suction, low-momentum fluid is removed from the boundary layer while higher-momentum fluid is drawn toward the wall, thereby delaying flow separation and promoting flow attachment. Zhang *et al.* (2017) found that optimal control is achieved when the suction region is located just downstream of the separation point. Zhi-yong *et al.* (2018) demonstrated that suction momentum has a more significant influence on aerodynamic performance than the suction angle. Both studies found that minimum drag and maximum aerodynamic efficiency occur at intermediate values of the momentum coefficient.

Conversely, steady uniform blowing injects momentum into the boundary layer, re-energising the flow and enhancing its ability to overcome adverse pressure gradients. This approach is commonly used in circulation control applications (Chen *et al.* 2012; Radespiel *et al.* 2016). Blowing has also been shown to reduce skin-friction

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

drag in turbulent boundary layers when low-intensity wall-normal injection is applied (Kametani & Fukagata 2011; O'Connor *et al.* 2023). However, this can also lead to increased pressure drag in the presence of adverse pressure gradients, potentially resulting in higher overall drag in aerodynamic configurations such as wings (Atzori *et al.* 2020).

The periodic combination of suction and blowing with zero mean mass flux over a complete actuation cycle is widely used for separation control and stall delay. Typically referred to as synthetic jets or zero-net-mass-flux (ZNMF) actuation (Glezer & Amitay 2002), this approach exploits natural flow instabilities by utilising nonlinear resonance and frequency lock-in to reorganise the disordered and chaotic flow into concentrated lifting vortices (Wu *et al.* 1998). Therefore, unlike steady methods, their effectiveness relies not only on momentum intensity but also on the unsteady generation of vortical structures that enhance mixing and momentum transfer. Consequently, synthetic jets can achieve comparable lift enhancements with substantially lower momentum input than steady actuation (Seifert *et al.* 1996).

The effectiveness of ZNMF actuation is then primarily governed by the non-dimensional forcing frequency $F^+ = fc/U_\infty$, in addition to the momentum coefficient. Previous studies have consistently shown that forcing at $F^+ \approx 1$, corresponding to the order of the natural wake shedding frequency, is the most effective for large-scale separation control (Gilarranz *et al.* 2005; You & Moin 2008; Raju *et al.* 2008; Buchmann *et al.* 2013; Zhang & Samtaney 2015).

For such excitation frequencies, optimal performance is typically achieved within a moderate range of momentum coefficients, with reported values around $0.01 < C_\mu < 0.015$ (McCormick 2000; Rodriguez *et al.* 2020). Furthermore, the required momentum coefficient decreases when the forcing location is placed close to the separation point (Amitay *et al.* 2001; Feero *et al.* 2017).

1.2. *Deep reinforcement learning for active flow control*

Since the early work of Rabault *et al.* (2019) on the stabilisation of a two-dimensional (2-D) cylinder wake at a Reynolds number of $Re_D = 100$, DRL has evolved from controlling low-*Re* laminar flows (Tang *et al.* 2020; Varela *et al.* 2022) to managing weakly turbulent regimes (Ren *et al.* 2021; Varela *et al.* 2022) up to $Re_D = 2000$; where $Re_D = \rho U_\infty D/\mu$ is defined using the cylinder diameter D as the characteristic length and μ denotes the fluid viscosity. These early studies demonstrated that DRL agents can outperform periodic forcing by identifying non-trivial control strategies that suppress vortex shedding in 2-D cylinders with greater energy efficiency.

More recently, the extension to three-dimensional (3-D) and fully turbulent flows has been enabled by two key developments: (i) multi-environment parallelisation, where multiple computational fluid dynamics (CFD) simulations are run concurrently to accelerate the training (Rabault & Kuhnle 2019); and (ii) multi-agent reinforcement learning (MARL), which overcomes the 'curse of dimensionality' arising when controlling multiple actuators simultaneously (Belus *et al.* 2019). This framework has facilitated drag reduction in turbulent channel flows at around $Re_\tau = 180$ (Guastoni *et al.* 2023; Sonoda *et al.* 2023) and the control of 3-D cylinder wakes up to $Re_D = 3900$ (Suárez *et al.* 2025*a*,*b*). Here, the friction Reynolds number is defined as $Re_\tau = \rho\, u_\tau\, \delta/\mu$; with $u_\tau = \sqrt{\tau_w/\rho}$ the friction velocity, τ_w the wall shear stress and δ the boundary layer thickness. Furthermore, strategies like coarse-mesh training (Font *et al.* 2025) and transfer learning from two to three dimensions (Chatzimanolakis *et al.* 2024; Liu *et al.* 2025) have significantly mitigated the high computational cost of 3-D DRL-AFC.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

A critical challenge for practical AFC is achieving high performance with minimal sensor data. Paris *et al.* (2021) demonstrated that sensor placement can be optimised by selecting only the most informative observations for a 2-D cylinder. Similarly, Wang *et al.* (2024) proposed a dynamic feature space lifting to achieve effective control with a single surface-mounted sensor. This is an important step towards practical DRL applications, which will necessarily rely on limited wall-based sensor input data. In related work, Yan *et al.* (2023) investigated wake control behind a square cylinder using surface pressure sensors.

Some authors have also investigated the use of alternative deep-learning architectures. While all of the aforementioned studies employed relatively simple multilayer perceptron (MLP) architectures, Jeon *et al.* (2025) demonstrated that using either group convolutional neural networks or positional encoding leads to faster and better learning. Following a similar approach, Kurz *et al.* (2025) examined the benefits of graph neural networks (GNN). Although both MLP and GNN achieved similar average drag reductions, the GNN showed improved generalisation due to their inherent invariance to input ordering. Other researchers have focused on enhancing the controller's perception and interpretability. Cavallazzi *et al.* (2025) employed a convolutional neural network for feature extraction before passing the state to the agent, and Beneitez *et al.* (2025) integrated explainable deep learning for targeting specific turbulence-sustaining structures in an open channel, leading to more physically consistent control strategies. Separately, Weissenbacher *et al.* (2025) examined transformer models, leveraging attention mechanisms to address partial observability with sparse sensing.

In the context of airfoils and wings, the combination of DRL with AFC remains limited, with most studies focusing on 2-D geometries at low Reynolds numbers. At $Re_c = 3000$ (based on the airfoil chord c), Wang *et al.* (2022) and Garcia *et al.* (2025) reported substantial aerodynamic performance improvements using DRL on a 2-D airfoil, outperforming periodic control in both effectiveness and energy efficiency. More recently, Liu *et al.* (2025) demonstrated that a DRL policy trained on 2-D flow around an airfoil at $Re_c = 20\,000$ can be successfully transferred to the 3-D wing. In their study, the agent controlled the amplitude of a prescribed spanwise sinusoidal jet velocity profile. Other wing-related studies have also primarily focused on optimising static actuation parameters, such as actuator placement (Paris *et al.* 2023), rather than learning temporal control laws.

1.3. *Flow past the NACA0012 wing at $Re_c = 1000$*

The flow past the NACA0012 airfoil at a Reynolds number of $Re_c = 1000$ and angle of attack $\mathrm{AoA} = 20°$ is studied. At this Reynolds number, the wake topology and flow dynamics undergo significant evolution as the angle of attack increases. Numerical and experimental investigations by Kurtulus (2015) and Gupta *et al.* (2023) identified a primary transition at $\mathrm{AoA} > 8°$, where the continuous vortex sheet in the wake evolves into an alternating vortex-shedding mode, resembling a von Kármán vortex street. This transition introduces periodic oscillations in the aerodynamic coefficients, a phenomenon further confirmed by Kouser *et al.* (2021).

As the angle of attack exceeds $\mathrm{AoA} > 13°$, 3-D flow structures begin to develop, resulting in loss of the spanwise homogeneity of the vortex-shedding structures and the formation of parallel vortex rows with a characteristic spanwise wavelength. While 2-D simulations capture the initial shedding frequency, they fail to account for the spanwise instabilities that emerge in this regime (Kurtulus 2015; Kouser *et al.* 2021). According to Kokash & Agbaglah (2024), this transition is attributed to the strong shear stress on the airfoil upper surface, which amplifies surface disturbances that evolve into characteristic

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

streamwise rib-like vortices known as mode B (Williamson 1988). For configurations near AoA = 16°, the resulting spanwise velocity exhibits a sinusoidal pattern with a characteristic wavelength between $\lambda_z/c = 0.6$ (Kokash & Agbaglah 2024) and 0.66 (Kouser *et al.* 2021). These values align closely with the mode B wavelengths reported by Barkley & Henderson (1996) for circular cylinders at an equivalent effective Reynolds number ($Re_D = Re_c \sin \text{AoA} \approx 280$).

Finally, Kouser *et al.* (2021) demonstrated that the wake remains relatively stable in the range of AoA = 14°−17°, but progressively transitions to a chaotic and unstable wake as the angle of attack increases beyond AoA > 18°. This transition leads to the breakdown of the ordered pattern of spanwise-aligned vortex rows, reflecting the onset of a complex 3-D dynamics. For the 2-D case at AoA = 20°, Hoarau *et al.* (2003) identified that the aperiodicity of the von Kármán mode at higher angles of attack is lost via a period-doubling scenario. Moreover, they also studied the history of the 3-D transition mechanisms. In particular, they showed that the amplification of longitudinal and vertical vorticity components perturbs the rectilinear von Kármán rows, causing them to undulate into large-scale 3-D structures. Therefore, at AoA = 20°, the present case lies within this regime, where increased three-dimensionality and weakly turbulent behaviour make the flow particularly challenging for DRL.

1.4. *Contribution*

The present AFC-DRL framework, initially tested on a separated turbulent boundary layer at $Re_\tau = 750$ (Font *et al.* 2025) and further validated with a 3-D cylinder at $Re_D = 100$ (Montalà *et al.* 2024), advances the application of DRL to AFC in realistic aerodynamic configurations.

In contrast to previous studies (see table 1), the present work performs training directly in a fully 3-D flow configuration at a relatively high Reynolds number within the context of DRL, where sample inefficiency remains one of the major limitations. As a result, most existing studies are restricted to low Reynolds number regimes, while those addressing higher Reynolds numbers typically rely on simplified approaches, such as 2-D configurations (Wang *et al.* 2022; Garcia *et al.* 2025), transfer learning from reduced environments (Chatzimanolakis *et al.* 2024; Liu *et al.* 2025) or simplified geometries (Suárez *et al.* 2025*a*; Font *et al.* 2025). By leveraging a solver accelerated through graphics processing units (GPUs), the present work overcomes part of this limitation and enables training in a fully 3-D setting. This allows the agent to interact with the inherently 3-D and chaotic dynamics characteristic of realistic turbulent flows. To the best of the authors' knowledge, this represents the first application of DRL to control a 3-D wing CFD simulation.

Furthermore, the study aims to highlight the flexibility of DRL as an adaptive control framework. By varying the relative weighting of the reward function terms, the controller can prioritise different objectives, such as lift enhancement or drag reduction. Unlike classical periodic forcing, which is inherently fixed, the learned controller adjusts its actuation in response to instantaneous flow conditions and varying optimisation goals, pointing toward multi-objective and time-dependent flow control in realistic operating environments characterised by unsteady and uncertain conditions.

2. Methodology

In this work, numerical simulations are performed to investigate the flow past a 3-D NACA0012 wing section at a Reynolds number of $Re_c = 1000$ and an angle of attack of AoA = 20°, with spanwise-periodic boundary conditions. Additionally, the interaction of

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

References	Case study	Flow regime	Algorithm
Rabault *et al.* (2019)	2-D cylinder	$Re_D = 100$	PPO
Tang *et al.* (2020)	2-D cylinder	$Re_D = 100{-}400$	PPO
Paris *et al.* (2021)	2-D cylinder	$Re_D = 120$	PPO
Ren *et al.* (2021)	2-D cylinder	$Re_D = 1000$	PPO
Varela *et al.* (2022)	2-D cylinder	$Re_D = 100{-}2000$	PPO
Wang *et al.* (2022)	2-D wing	$Re_c = 3000$	PPO
Yan *et al.* (2023)	2-D square cylinder	$Re_D = 500{-}2000$	SAC
Sonoda *et al.* (2023)	Channel flow	$Re_\tau = 150$	DDPG
Guastoni *et al.* (2023)	Channel flow	$Re_\tau = 180$	DDPG
Vignon *et al.* (2023)	2-D Rayleigh–Bénard	$Ra = 10\,000$	PPO
Vasanth *et al.* (2024)	3-D Rayleigh–Bénard	$Ra = 500{-}750$	PPO
Chatzimanolakis *et al.* (2024)	3-D cylin. (2-D train)	$Re_D = 1000{-}4000$	V-Racer
Wang *et al.* (2024)	2-D cylinder	$Re_D = 100{-}1000$	SAC
Cavallazzi *et al.* (2025)	Channel flow	$R_\tau = 200$	PPO
Suárez *et al.* (2025*b*)	3-D cylinder	$Re_D = 100{-}400$	PPO
Suárez *et al.* (2025*a*)	3-D cylinder	$Re_D = 3900$	PPO
Garcia *et al.* (2025)	2-D wing	$Re_c = 3000$	PPO
Font *et al.* (2025)	Separation control	$Re_\tau = 750$	PPO
Liu *et al.* (2025)	3-D wing (2-D train)	$Re_c = 20\,000$	—
Kurz *et al.* (2025)	2-D cylinder	$Re_D = 100$	PPO
Beneitez *et al.* (2025)	Channel flow	$Re_\tau = 180$	TD3
Jeon *et al.* (2025)	2-D Rayleigh–Bénard	$Ra = 10\,000$	PPO

Table 1. A summary of the existing computational studies on DRL and its applications in AFC.

jets located along the wing surface is included in the simulations, with their mass-flow rates optimised through DRL.

The DRL-CFD framework is illustrated in figure 1. It consists of two main components: the CFD environment and the DRL agent. At the core of the agent lies a neural network (NN) that is responsible for mapping flow states to corresponding control actions. In the present work, the NN processes pressure data collected from sensors located within the flow domain (state) and determines the appropriate jet velocity to be applied back into the system (action). To refine its control policy, learning is achieved through an iterative process between the agent and the environment driven by trial and error. Therefore, the CFD solver and the agent exchange not only states and actions but also rewards. The reward function quantifies the effectiveness of a control action, typically incorporating aerodynamic performance metrics such as lift enhancement or drag reduction.

The environment is implemented using the SOD2D solver (Gasparino *et al.* 2024), developed in Fortran 90 language, whereas the DRL agent is built using the TF-Agents library (Guadarrama *et al.* 2018) in Python. To overcome the 'two-language' set-up, both components are coupled via a Redis in-memory database managed through the SmartSim API (Partee *et al.* 2022). This enables efficient and low-overhead data exchange between the Fortran-based CFD solver and the Python-based DRL agent. The resulting framework is called SmartSOD2D (Font *et al.* 2025) and was originally created as a fork of the RELEXI project (Kurz *et al.* 2022).

Further details for each component are provided in the following sections: § 2.1 describes the numerical set-up of the CFD simulations, while § 2.2 explains the DRL configuration.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

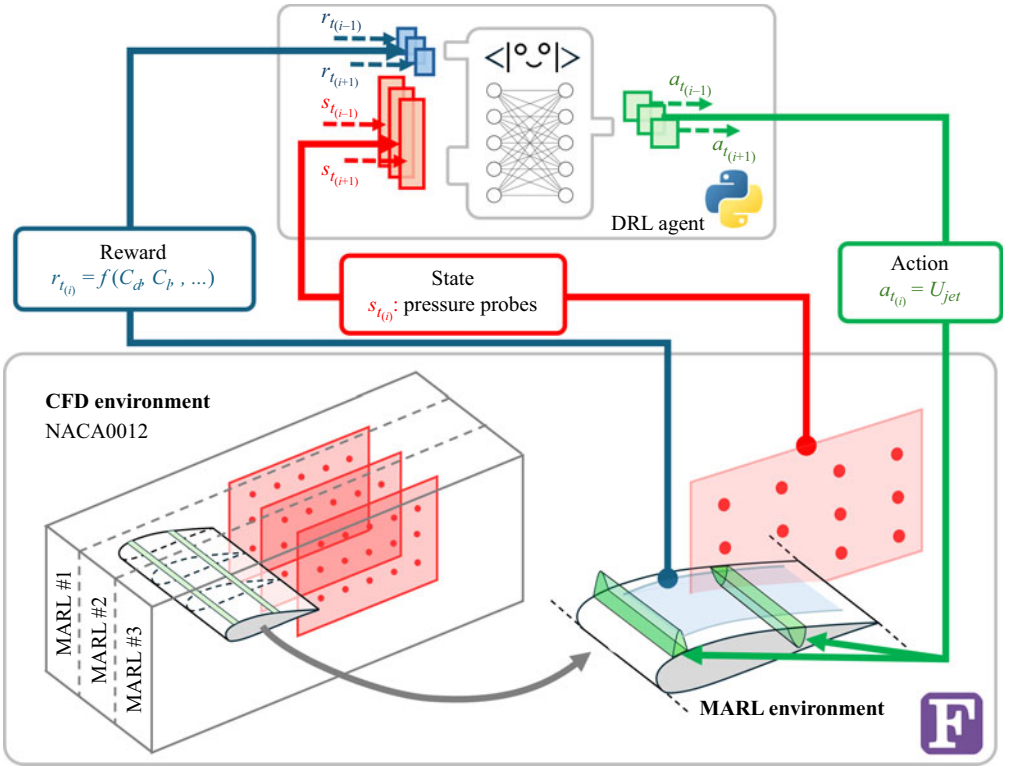


Figure 1. Schematic of the DRL framework showing the MARL strategy.

2.1. *Computational fluid dynamics configuration*

To predict the dynamic evolution of the system, the incompressible Navier–Stokes equations are integrated numerically. These equations are written as

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{2.1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) - \frac{\mu}{\rho}\frac{\partial^2 u_i}{\partial x_j \partial x_j} + \frac{1}{\rho}\frac{\partial p}{\partial x_i} = 0, \tag{2.2}$$

where $u_i = (u, v, w)$ and p denote the velocity and pressure fields, respectively.

To solve these equations, the SOD2D solver is used (Gasparino *et al.* 2024). This is a GPU-enabled spectral-element method (SEM; Patera 1984) code developed at the Barcelona Supercomputing Center (BSC). The flow solver employs projection stabilisation to prevent numerical oscillations arising from dominant convection while introducing minimal numerical dissipation. Aliasing effects caused by the reduced-order integration of convective terms in the SEM are mitigated using a skew-symmetric splitting approach (Kennedy & Gruber 2008). Temporal discretisation is performed with a third-order Backward Differentiation Formula with third-order explicit extrapolation operator-splitting scheme to solve the velocity–pressure coupling, enabling equal-order interpolation of both velocity and pressure (Karniadakis *et al.* 1991). This solver is designed for high-fidelity simulations, and its scalability and computational efficiency on distributed accelerator-based architectures make it well suited for DRL applications, where rapid CFD data generation is crucial for the training.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

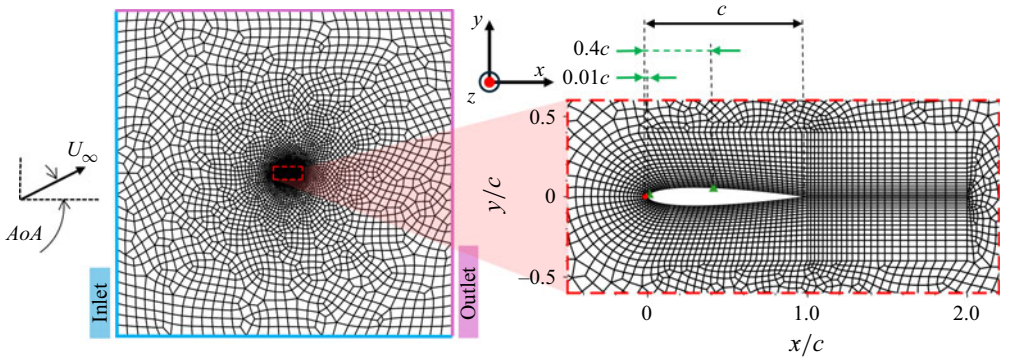


Figure 2. Visualisation of the computational mesh. The full domain is shown on the left, while a zoomed-in view around the wing is displayed on the right. Only the spectral elements are shown, whereas Gauss–Lobatto integration points are not displayed. The red dot indicates the origin of the coordinate system, while the green triangles denote the locations of the jets.

The computational domain extends over $L_x/c = [-15, 16]$, $L_y/c = [-15, 15]$ and $L_z/c = [0, 3]$ in the streamwise, cross-stream and spanwise directions, respectively, with the wing leading edge located at the origin, i.e. $(x, y)/c = (0.0, 0.0)$. A 2-D slice of the mesh in the $x-y$ plane is illustrated in figure 2. An unstructured hexahedral mesh is employed throughout the domain, with a more regular and wall-aligned element arrangement in the vicinity of the wing. Since a SEM is used, each element shown in the figure contains several integration points distributed according to the Gauss–Lobatto quadrature. In this work, fourth-order polynomial functions are employed ($p = 4$). The 2-D mesh contains around $N_{xy} = 1.05 \times 10^5$ degrees of freedom (DoF). This mesh is then extruded along the spanwise (or z) direction with a total of eighteen spectral elements, corresponding to $N_z = 73$ integration points. This yields a 3-D mesh containing approximately $N_{xyz} = 7.6 \times 10^6$ DoF. To determine the appropriate mesh resolution, a mesh sensitivity analysis is presented in Appendix A.

At the inlet, a constant free-stream velocity U_∞ is imposed, corresponding to the target Reynolds number. This velocity is decomposed into the x and y directions according to the defined angle of attack. At the outlet, zero-gradient conditions with constant pressure are applied. The no-slip boundary condition is enforced on the wing surface, while periodic boundary conditions are imposed in the spanwise direction. Finally, on the jet surfaces, Dirichlet boundary conditions are used to specify the target jet velocity when AFC is active. This velocity is uniform across the entire jet outlet and oriented towards the y-axis of the global coordinate system.

To explore 3-D actuations without causing excessive computational cost during DRL training, a MARL strategy is adopted (Belus *et al.* 2019). Leveraging the spanwise invariance of the flow, the computational domain is partitioned into several subdomains referred to as MARL pseudo-environments, each featuring an independent jet actuator spanning the entire width of the subdomain. A single, shared policy is used for all the MARL pseudo-environments at the same time, which processes local state information to output a different individual jet velocity to each pseudo-environment. This set-up is schematised in figure 1 and has been successfully applied in related studies (Suárez *et al.* 2025*a*,*b*; Font *et al.* 2025).

In this study, three MARL pseudo-environments are defined ($n_{MARL} = 3$). Within every subdomain, two jets are present: one near the leading edge of the wing at approximately $x/c = 0.01$, and the other further downstream at $x/c = 0.4$ (as indicated in figure 2).

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

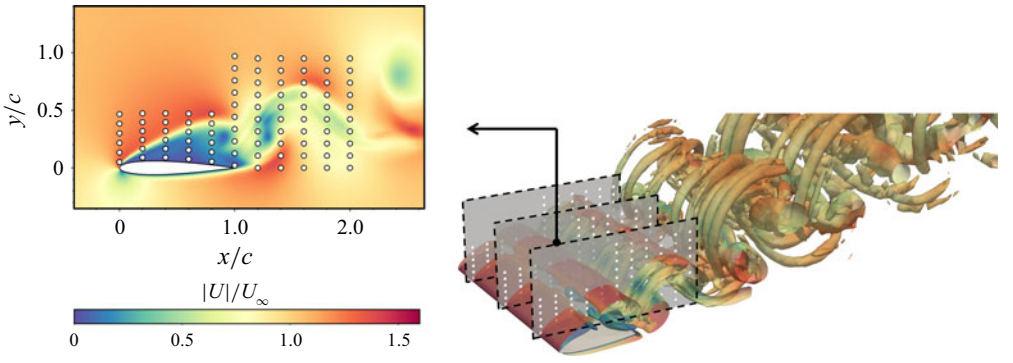


Figure 3. Sensor locations used as the DRL agent state, superimposed on a snapshot of the baseline flow field: zoomed view shows instantaneous velocity magnitude, general 3-D view shows Q-criterion (the second invariant of the velocity gradient tensor) iso-contours $Qc^2/U_\infty^2 = 0.1$ coloured by velocity magnitude.

Both jets span the entire width of their respective subdomains, i.e. $L_{z,jet} = L_z/3 = 1.0c$, and extend an arc length of $L_{xy,jet} = 0.03c$ along the chord-wise direction. Thus, the corresponding individual jet surface can be calculated as $S_{jet} = L_{z,jet} \times L_{xy,jet}$. The agent directly controls the upstream jet, while the velocity of the downstream jet is set to have the opposite mass-flow rate, ensuring ZNMF jets and hence mass conservation at every time step. The agent is restricted to exploring values within the range $U_{jet}/U_\infty = [-1.125, 1.125]$.

For the state representation s_t, each pseudo-environment uses pressure data from 90 sensors distributed across its mid-span plane. As shown in figure 3, the sensor layout is designed based on the flow structures observed in the uncontrolled case. The sensors are positioned above the wing, within the region where the wake is most prominent, aiming to capture both the main separation bubble and the downstream wake dynamics. Moreover, to account for 3-D flow characteristics, the DRL agent input combines the data from the 90 probes in its own MARL environment with those adjacent on either side, resulting in an input size of $90 \times 3 = 270$ probes.

Finally, to further accelerate the data-collection process, ten independent CFD simulations are executed in parallel during training ($n_{CFD} = 10$), following the approach proposed by Rabault & Kuhnle (2019). As a result, each training episode yields $n_{MARL} \times n_{CFD} = 30$ distinct state–action–reward trajectories τ used to perform a single training step of the agent. The resulting speed-up achieved by employing $n_{env} = n_{MARL} \times n_{CFD}$ parallel environments is further analysed in Appendix B.

The total episode duration is set to $T_{eps} = 10.52U_\infty/c$, corresponding to six vortex-shedding cycles of the baseline flow. Each pseudo-environment performs 120 discrete control actions per episode, resulting in an action duration of $T_{act} = T_{eps}/120 = 0.088U_\infty/c$. This duration corresponds to roughly 5 % of the baseline vortex-shedding period, which has been found sufficient in the literature to provide control authority over the system (Rabault & Kuhnle 2019).

In order to avoid spurious, non-physical oscillations in the numerical simulation, the newly predicted action is not applied directly to the environment. Instead, exponential smoothing is used to interpolate between successive discrete actions $a_{t-1} \to a_t$, following the approach of Suárez *et al.* (2025*a*, *b*). As a result, the control signal evolves continuously within the CFD environment, reaching the target value a_t at the end of the action period T_{act}, when a new action is provided by the agent.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

Parameter	Value	Parameter	Value
Number of DoF (N_{xyz})	7.6M	Restart baseline/previous eps.	50/50 %
Domain size ($[\Delta L_x, \Delta L_y, \Delta L_z]$)	[31c, 30c, 3c]	Jet surface ($L_{z,jet} \times L_{xy,jet}$)	$1.0c \times 0.03c$
Episode duration ($T_{eps}\, c/U_\infty$)	10.52	Jet velocity range (U_{jet}/U_∞)	$[-1.125, 1.125]$
Actions/episode	120	Front jet location (x/c)	0.01
Parallel CFD environments (n_{CFD})	10	Rear jet location (x/c)	0.4
MARL pseudo-environments (n_{MARL})	3	State size (variable)	90×3 (pressure)

Table 2. Summary of the CFD environment parameters utilised in this study.

At the end of each episode, the trajectories τ gathered across the ten parallel CFD simulations are combined into a batch, which the agent then uses to perform a training step. Once this is completed, simulations are re-launched to generate data for the next episode. Each simulation has an equal probability of restarting from either the baseline scenario or the final state of the preceding episode. This ensures that the DRL agent observes long-term controlled flow features while still effectively learning the highly dynamic initial transition phase from the uncontrolled to controlled regime. An overview of all parameters employed in the CFD environment is provided in table 2.

2.2. *Agent configuration*

The interaction between the agent and the environment is modelled as a Markov decision process. At each discrete time t, the agent receives a state s_t and selects an action a_t according to a policy $\pi_\theta(a_t \mid s_t)$. In DRL, a NN is used to approximate this policy. Following the execution of action a_t the environment evolves from state s_t to the next state s_{t+1} according to the incompressible Navier-Stokes equations. It then returns a scalar reward R_{t+1}, which quantifies the quality of the selected action a_t.

This process continues sequentially until the end of the episode, resulting in a trajectory of the form $\tau = \{(s_0, a_0, R_1, s_1),\ (s_1, a_1, R_2, s_2),\ \ldots,\ (s_t, a_t, R_{t+1}, s_{t+1})\}$, which is then used to refine the policy during the training. In this work, a batch of trajectories collected from n_{env} parallel environments is used to perform each training step. Each individual rollout generated by a single environment is referred to as a MARL episode. The objective throughout this learning process is to maximise the expected return, defined as the cumulative discounted reward

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \ldots = \sum_{l=0}^{\infty} \gamma^l R_{t+1+l}, \tag{2.3}$$

where $\gamma \in [0, 1)$ is the discount factor that balances the importance of immediate versus future rewards. For the present computations, the discount factor is set to $\gamma = 0.99$.

The proximal policy optimisation (PPO) algorithm is employed to iteratively improve the agent's policy over the course of training (Schulman *et al.* 2017). Proximal policy optimisation is a model-free, on-policy and actor-critic method, where the actor learns the policy $\pi_\theta(a_t|s_t)$ and the critic estimates the value function $V_\theta(s_t)$, representing the expected return under the current policy. This is then used to compute the advantage function A_t that guides the training of the actor.

Both the actor and critic networks consist of two separate MLPs, each featuring two hidden layers of 512 neurons. Training is performed using the Adam optimiser (Kingma & Ba 2014) with a learning rate of $\mathrm{LR} = 5 \times 10^{-4}$. The critic is trained by minimising the mean squared error between the predicted and the observed returns, while the actor

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

Parameter	Value	Parameter	Value
Discount factor (γ)	0.99	MLPs architecture (neurons × layers)	512 × 2
GAE lambda (λ)	0.95	Reward local balance (η)	0.8
Value loss coefficient (c_1)	0.5	Lift-fluctuation penalty (α)	0.3
Entropy coefficient (c_2)	0.0	Lift improvement bonus (β)	0.0 and 0.5
Clipping ratio (ϵ)	0.3	Optimiser type	Adam
Learning rate (LR)	5×10^{-4}	Epochs per optimisation	30

Table 3. Summary of PPO hyperparameters used during training.

updates its policy parameters guided by the estimated advantages, which in this work are computed using the generalised advantage estimation (GAE; Schulman *et al.* 2015). This enables the actor to increase the likelihood of actions that yield higher-than-expected returns during training. Detailed mathematical formulations of the PPO agent are provided in Appendix C.

In this work, the training is conducted using the reward function R_i written in (2.4), where the index i denotes the corresponding MARL pseudo-environment. Thus, the reward accounts for both local r_i and global performance, represented by the mean reward computed across all pseudo-environments, with the weighting factor $\eta \in [0, 1]$ modulating the relative importance of each term. In this study, this is set to $\eta = 0.8$

$$R_i = \eta\, r_i + (1-\eta)/n_{MARL} \sum_{j=1}^{n_{MARL}} r_j, \tag{2.4}$$

$$r_i = (C_{d,base} - C_d) - \alpha\,|C_l - C_{l,avg}| + \beta(C_l - C_{l,base}). \tag{2.5}$$

The local reward r_i is calculated as defined in (2.5). This comprises three contributions: The first term encourages a reduction in drag coefficient C_d relative to the baseline $C_{d,base}$; the second penalises fluctuations in the lift coefficient, where $C_{l,avg}$ corresponds to the accumulated mean lift coefficient over the course of the episode; and the third promotes an increase in lift coefficient C_l compared with the baseline $C_{l,base}$.

The lift and drag coefficients are calculated as $C_l = l/(0.5\rho U_\infty S_{ref})$ and $C_d = d/(0.5\rho U_\infty S_{ref})$, where l and d denote the aerodynamic force components in the cross-stream and streamwise directions relative to the free-stream velocity U_∞. These coefficients are averaged over the duration of each action period T_{eps} before computing the reward.

Two separate training runs are conducted to assess the sensitivity to the reward hyperparameters. In both cases, $\alpha = 0.3$, while β is set to 0.0 in the first training and 0.5 in the second. These configurations are hereafter referred to as NLT-OFF and NLT-ON, corresponding to the net lift term being deactivated ($\beta = 0.0$) and activated ($\beta = 0.5$), respectively. All the hyperparameters used for training the PPO agent are listed in table 3.

3. Results

3.1. *Baseline validation*

Before introducing AFC strategies, the uncontrolled scenario is validated to ensure the accuracy and reliability of the numerical set-up. This case serves as both a baseline for evaluating performance improvements achieved through DRL-based control and the initial condition upon which control inputs will be applied.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

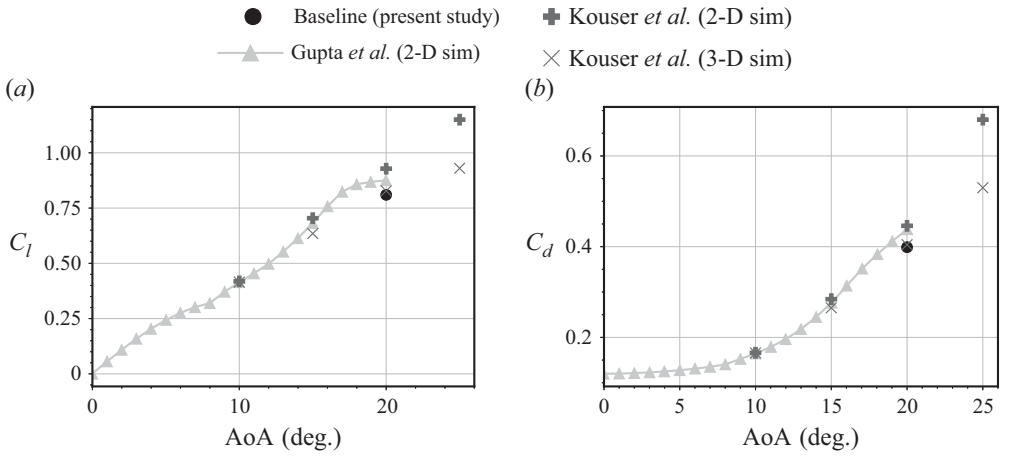


Figure 4. Variation of the mean (*a*) lift coefficient and (*b*) drag coefficient with the angle of attack, comparing the present results with the literature.

Figures 4(*a*) and 4(*b*) compare the mean lift and drag coefficients reported in the literature for different angles of attack. As can be observed, the discrepancies between the 2-D and 3-D simulations become increasingly pronounced as 3-D effects emerge beyond AoA $> 13°$, with differences growing larger at higher angles. Superimposed on these plots are the mean lift and drag coefficients obtained in this study, which closely match the 3-D results reported by Kouser *et al.* (2021) for the same angle of attack.

As discussed in § 1.3, the flow lies in a regime characterised by strong separation and chaotic three-dimensionalities in the wake. Previous studies report periodic spanwise wavelengths of $\lambda_z/c \approx 0.60 - 0.66$ at AoA $= 16°$ (Kouser *et al.* 2021; Kokash & Agbaglah 2024). Anticipating similar values at the current angle of attack, the chosen spanwise domain length of $L_z/c = 3.0$ is sufficient to capture this dynamics.

The statistical analysis of the baseline flow was performed over 40 vortex-shedding cycles after reaching the statistically steady state, corresponding to approximately $70c/U_\infty$. After this sampling period, the running averages of C_l and C_d, as well as their standard deviations, reached a plateau, indicating statistical convergence of the results. For more details, see Appendix A.

To quantitatively assess the present results, table 4 reports the statistics of the aerodynamic forces, together with the 2-D and 3-D results from Kouser *et al.* (2021). The table includes the mean lift coefficient, the standard deviation of the lift coefficient, the drag coefficient and the vortex-shedding Strouhal number $St = fc/U_\infty$, where f denotes the frequency. Here, f is obtained from the peak of the power-spectral density of the lift coefficient signal. Except for the lift coefficient fluctuations $C'_{l,rms}$, which are not reported in previous studies, the differences with respect to the literature are minimal: $|\Delta C_l| = 2.2\,\%$, $|\Delta C_d| = 1.2\,\%$ and $|\Delta St| = 3.7\,\%$. Hence, it can be concluded that these results show good agreement with other 3-D simulations and follow the expected trends reported in the literature, confirming the validity of the numerical set-up used in this study.

3.2. *Deep reinforcement learning control: training*

In the present work, two different DRL trainings are considered by varying the weighting factors that modulate the contribution of each term in the local reward function (see (2.5)). In this section, the results obtained using NLT-OFF ($\alpha = 0.3$, $\beta = 0.0$) and NLT-ON

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

Case ID	C_l	$C'_{l,rms}$	C_d	St
Kouser *et al.* (2021) – 2-D sim.	0.928	—	0.446	0.53
Kouser *et al.* (2021) – 3-D sim.	0.828	—	0.404	0.54
Baseline	0.810	0.085	0.399	0.56
DRL NLT-OFF	0.834 (+2.96 %)	0.039 (–54.1 %)	0.315 (–21.1 %)	0.64 (+14.3 %)
DRL NLT-ON	1.153 (+42.3 %)	0.141 (+65.9 %)	0.413 (+3.51 %)	0.57 (+1.79 %)

Table 4. Summary of key aerodynamic coefficients for the baseline case and the DRL-controlled scenarios, including the mean lift and drag, the r.m.s. of the lift fluctuations and the Strouhal number associated with vortex shedding. The baseline case is compared with other simulations reported in the literature.

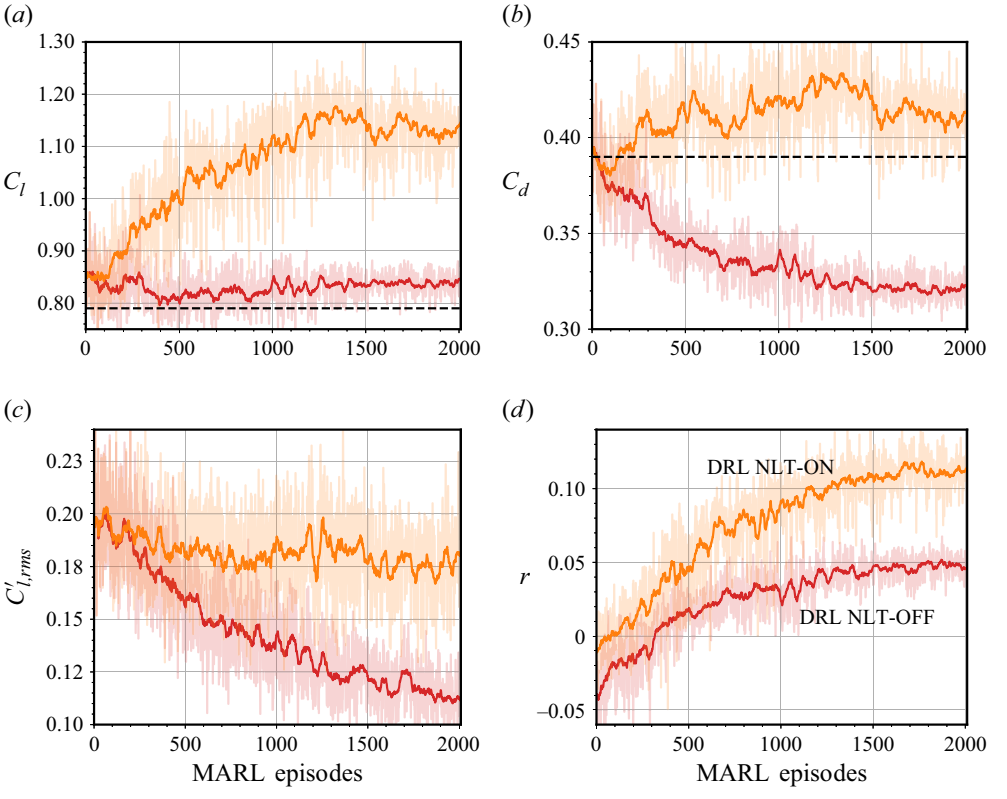


Figure 5. Evolution of the mean (*a*) lift coefficient, (*b*) drag coefficient, (*c*) lift coefficient fluctuations and (*d*) local reward over the last $5U_\infty/c$ of each episode during training. In each plot, a moving average is applied to smooth signals, while the individual data points are shown in a lighter, more transparent colour. The dashed black lines in (*a*) and (*b*) represent the baseline average value from the non-actuated case.

($\alpha = 0.3$, $\beta = 0.5$) are analysed, where the main difference lies in the greater importance given to the lift coefficient ($\beta > 0$) in NLT-ON. The remaining parameters are kept constant.

Figure 5 shows the evolution of the key aerodynamic parameters for both DRL policies during training, namely the lift coefficient C_l, drag coefficient C_d and the root mean square

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

(r.m.s.) of the lift coefficient fluctuations $C'_{l,rms}$, along with the local reward r, which reflects a weighted combination of the previous parameters. For each MARL episode, these values are averaged over the last $5U_\infty/c$ and appended sequentially to the plot. A moving average with a window size equal to the number of parallel environments n_{env} is then applied to smooth the resulting signals.

Training is conducted until a certain level of convergence is observed. In this case, the reward increases over the course of the training and reaches a plateau at around 2010 MARL episodes. At this point, training is stopped for both cases. In the present investigations, each training step combines the trajectories from every 30 MARL episodes, corresponding to the $n_{env} = n_{MARL} \times n_{CFD}$ environments running in parallel, resulting in a total of 67 training steps over the course of the training.

Each training step required approximately 10 minutes of wall-clock time on ten nodes of MareNostrum V, with each node equipped with four NVIDIA H100 GPUs and two Intel Xeon processors (80 CPU cores in total), leading to a total wall-clock time of around 11 hours per training case. This corresponds to a total computational cost of 440 GPU-hours. The reported time includes both experience collection, during which the agents interact with the multiple environments, and training time, when the NN parameters θ are updated. Notably, the NN optimisation step accounts for less than 0.1 % of the total computational time, highlighting the computational bottleneck of CFD environments.

From the values of C_l, C_d and $C'_{l,rms}$ shown in figure 5, it is evident that each training run adopts a completely different strategy. Under NLT-OFF, the lift coefficient remains relatively flat and slightly higher than the baseline. However, both the drag coefficient and the lift fluctuations exhibit a notable decreasing trend, significantly contributing to the increase in total reward. In contrast, for NLT-ON, the reward improvement is primarily driven by the lift coefficient, which increases substantially above the baseline as training progresses. Nevertheless, the drag coefficient remains close to the baseline and even performs slightly worse, while the reduction in lift fluctuations is less pronounced compared with NLT-OFF.

These results confirm that adding a term favouring lift enhancement effectively promotes its increase. However, the method appears highly sensitive to the weighting coefficients α and β, as even a change in β alone leads to markedly different optimisation strategies.

In principle, the weighting factor of $\beta = 0.5$ was selected to balance the lift and drag terms in the reward formulation of (2.5), given that $C_{l,base}$ is approximately twice $C_{d,base}$. Therefore, compared with the NLT-OFF scenario, one would intuitively expect a balance between drag reduction and lift enhancement under NLT-ON reward. However, in practice, it is observed that the agent predominantly optimises lift, highlighting one of the challenges associated with DRL-based control: identifying optimal parameters in the reward function for a given objective.

In the training curves shown in figure 5, the baseline values of C_l and C_d are included as black dashed lines for comparison, whereas no baseline line is plotted for $C'_{l,rms}$. While the averaged coefficients observed during training are generally representative of those expected during evaluation, the r.m.s. lift-fluctuation coefficient remains substantially higher than the baseline value of $C'_{l,rms} = 0.085$. This is an expected consequence of the agent's exploration process, which artificially introduces variance into the C_l signal, thereby inflating the r.m.s. fluctuations. In this context, the evolution of $C'_{l,rms}$ during training should be interpreted more as an indicator that the model is learning and adapting, rather than as a representation of the final expected magnitudes.

Additionally, due to the stochastic nature of the DRL algorithm and the additional variability introduced by GPU operations in the CFD solver, slight differences may arise

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

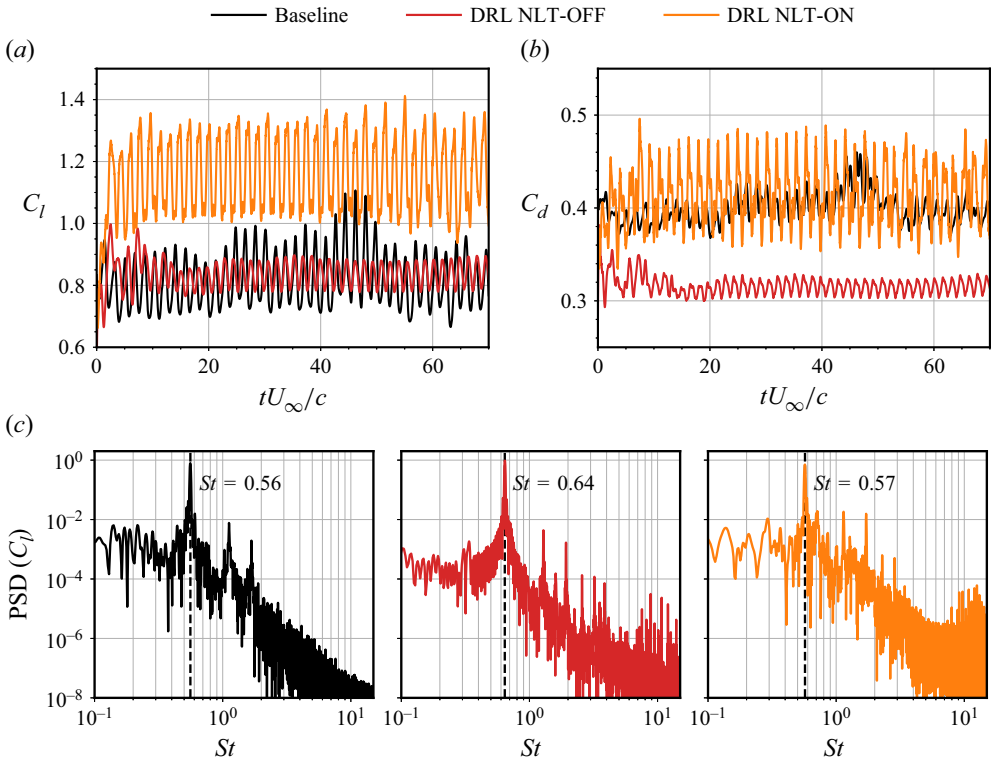


Figure 6. Temporal signals of the (*a*) lift and (*b*) drag coefficients when the DRL control is activated during exploitation compared with the baseline signals. (*c*) Power-spectral densities (PSDs) of the lift coefficient signals.

between different training runs of the same model. However, as shown in Appendix D, the results obtained across different runs remain well bounded and fall within a reasonable confidence interval.

3.3. *Deep reinforcement learning control: evaluation*

After the training phase, where the agent samples actions to discover those that may further increase the accumulated return (exploration), the evaluation mode is applied. In this mode, the most likely action produced by the policy is selected for each observed state (exploitation).

In figures 6(*a*) and 6(*b*), the lift coefficient and drag coefficient signals are shown against the baseline results once the control is activated ($tU_\infty/c > 0$). Initially, during approximately $tU_\infty/c \approx 15$, the flow transitions from the baseline to a new regime. Following this brief transient, the flow reaches a new statistically steady state. The averaged results are then integrated over 50 U_∞/c, which has been shown to be sufficient for convergence (see Appendix A). The resulting aerodynamic coefficients are summarised in table 4.

The performance differences between the two learned policies can be clearly visualised in the lift and drag signals in figures 6(*a*) and 6(*b*). When NLT-OFF is used, the agent primarily focuses on wake stabilisation, as evidenced by the substantial reduction in flow fluctuations and a more regular and periodic pattern of the lift signal, which ultimately leads to drag reduction. Although the drag term numerically dominates the reward

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

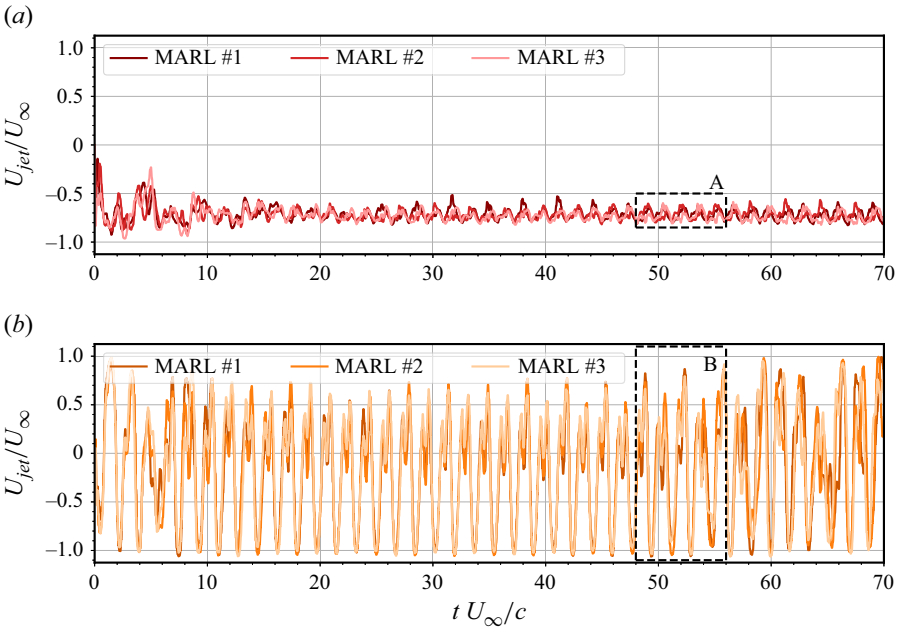


Figure 7. Control actions for (*a*) DRL NLT-OFF and (*b*) DRL NLT-ON from the onset of control activation. Only the front jets are shown, with each MARL pseudo-environment represented by a distinct colour.

function due to its larger weighting factor, the lift-fluctuation term consistently decreases throughout training (figure 5*c*). In contrast, the drag contribution plateaus early (figure 5*b*). This behaviour indicates that the agent has identified the wake stabilisation as the main mechanism to increase the reward, with drag reduction emerging as a secondary effect of the continuous suppression of lift fluctuations. This mechanism resembles the results reported by Rabault *et al.* (2019) for a 2-D cylinder case, and later observed in the 3-D case by Suárez *et al.* (2025*b*), where equivalent reward functions were employed. As indicated in table 4, this policy achieves a mean drag reduction of 21.1 % and a 54.1 % decrease in lift fluctuations, while maintaining the mean lift close to the baseline value, with only a slight increase of approximately 3.0 %.

By contrast, the performance under NLT-ON is markedly different. Here, the agent primarily focuses on increasing the lift coefficient, but this comes with notable drawbacks. The fluctuations in aerodynamic forces are substantially larger than in the baseline, and the drag shows no reduction. With this policy, the mean lift increases by 42.3 %, while the lift fluctuations and drag increase by 65.9 % and 3.5 %, respectively, as summarised in table 4. These performance trends for both rewards were already anticipated during training, as indicated by the evolution of the reward signals.

In both the C_l and C_d time signals, periodic patterns remain clearly visible after DRL control is applied, similar to the baseline flow, where strong vortex shedding dominates the dynamics. This is further supported by the lift coefficient spectra shown in figure 6(*c*), where distinct spectral peaks are present, indicating that the main vortex-shedding frequency is still active. This analysis was performed after discarding the transient period in the DRL cases. The corresponding Strouhal numbers are reported in table 4. While NLT-ON does not show an important variation of the Strouhal number compared with the baseline, under NLT-OFF, this is increased by approximately 14 %.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

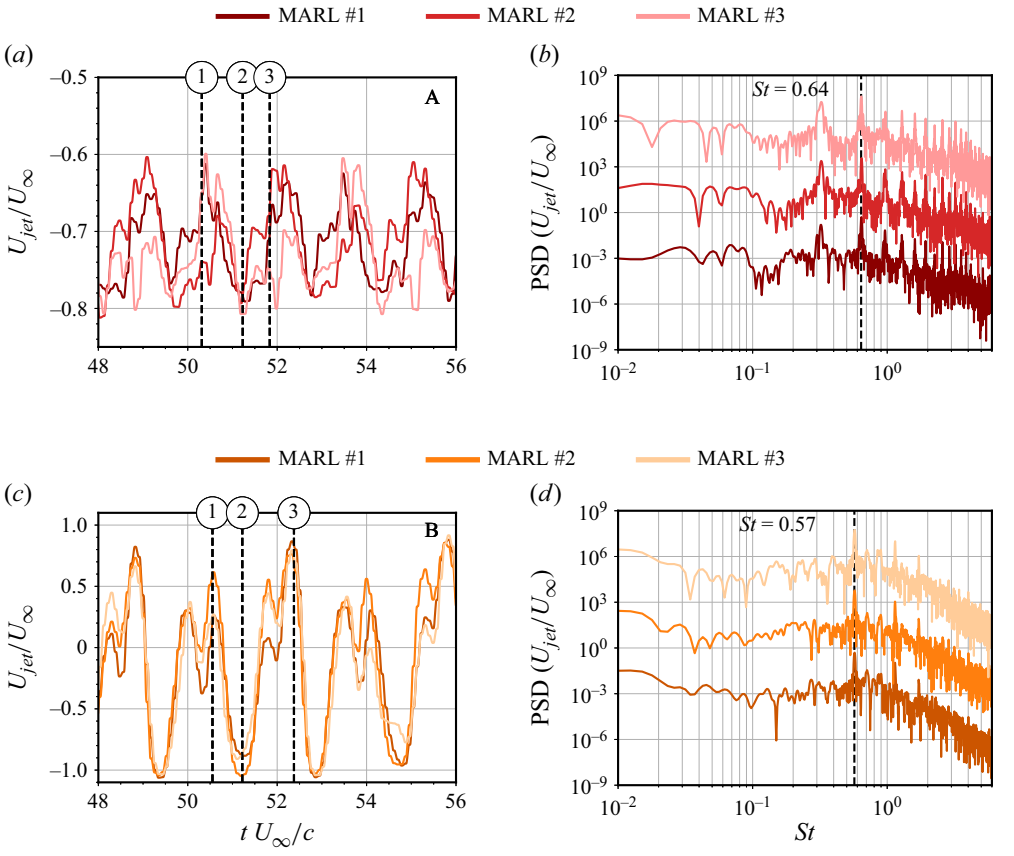


Figure 8. Zoomed view of the control actions from figure 7 in the statistically steady state for (*a*) DRL NLT-OFF and (*b*) DRL NLT-ON. Only the front jets are displayed, and each MARL pseudo-environment is represented by a distinct colour. The corresponding PSDs are shown on (*b*) and (*d*), respectively, with offsets applied across the MARL pseudo-environments for clarity.

The instantaneous control actions U_{jet}/U_∞ applied by the DRL are shown in figure 7 for the two policies. Only the actions of the front jet are displayed, with each colour corresponding to one MARL pseudo-environment, thereby illustrating the variation of the action along the spanwise direction as well. As described in § 2.1, the rear jet simultaneously applies the opposite mass-flow rate to ensure mass conservation.

As observed in these plots, the agent initially destabilises the baseline flow before reaching a new statistically steady state. This behaviour was also reflected in the C_l and C_d signals in figure 6. Once the statistically steady state is reached ($tU_\infty/c > 15$), it becomes evident that the control strategy strongly depends on the selected reward. A zoomed view of the jet actions in the stationary regime is provided in figures 8(*a*) and figure 8(*c*). Additionally, the corresponding PSDs for the NLT-OFF and NLT-ON actions after discarding the initial transient are presented in figures 8(*b*) and 8(*d*), respectively.

Under NLT-OFF, the control actions are consistently negative, indicating suction by the front jet and corresponding blowing by the rear jet. The signals have a mean offset of around $U_{jet}/U_\infty = -0.72$, with minor fluctuations around this value. This corresponds to a momentum coefficient of $C_\mu = 0.016$, based on the r.m.s. of the jet velocity. A periodic pattern can be identified, with peaks and valleys located at approximately the

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

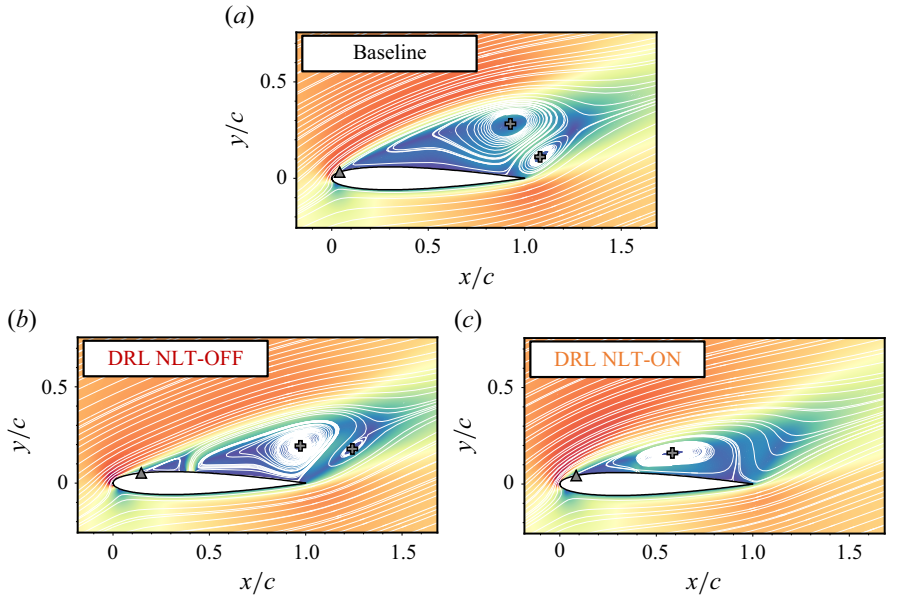


Figure 9. Time-averaged velocity streamlines with the velocity magnitude shown in the background. Results are presented for the (*a*) baseline, (*b*) DRL NLT-OFF and (*c*) DRL NLT-ON scenarios. Crosses indicate the locations of the vortex cores, and triangles denote the flow separation point at the leading edge.

same location, although some chaotic variability persists across different MARL pseudo-environments. In this case, two dominant frequencies are detected. One at $St = 0.64$, corresponding to the natural vortex-shedding frequency of the flow, as confirmed by the lift coefficient spectra in figure 6(*c*), and a sub-harmonic at $St = 0.32$.

This strategy resembles the suction mechanism observed in previous AFC studies, where stabilising the wake is achieved by applying suction at the leading edge. By drawing extra momentum into the boundary layer at this critical location, the flow better overcomes the high curvature near the leading edge, thereby delaying shear layer separation. As demonstrated in figure 9(*b*), this shifts the shear layer closer to the suction surface, physically limiting the entrainment and growth of the trailing-edge vortex. Consequently, the cross-stream extent of the wake structures is reduced, leading to a narrower wake with smaller vertical spacing between vortex cores and an increased shedding frequency. The observed vortex-shedding frequency in this case is therefore driven primarily by flow stabilisation rather than by the DRL controller.

In contrast, NLT-ON results in much more correlated signals across environments. Here, the actions follow a more 2-D and synchronised control strategy, with overlapping periodic signals. This strategy involves both suction and blowing, as the signals exhibit negative and positive values, although the mean value remains negative at approximately $U_{jet}/U_\infty = -0.17$. The amplitude of fluctuations is clearly higher than in the NLT-OFF case, which explains the increased values of $C'_{l,rms}$ observed earlier for this scenario. In this case, the r.m.s. of the jet velocity is $U_{jet,\,rms}/U_\infty = 0.58$, corresponding to a momentum coefficient of $C_\mu = 0.011$. Furthermore, a single dominant frequency emerges in the actions, closely matching the natural vortex-shedding frequency of the uncontrolled flow, i.e. $St = 0.57$.

This behaviour resembles periodic forcing strategies employed in previous AFC studies to delay stall. A frequency lock-in occurs, with the DRL agent effectively exploiting resonance. As shown in figure 9(*c*), the trailing-edge vortex is suppressed, while a

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

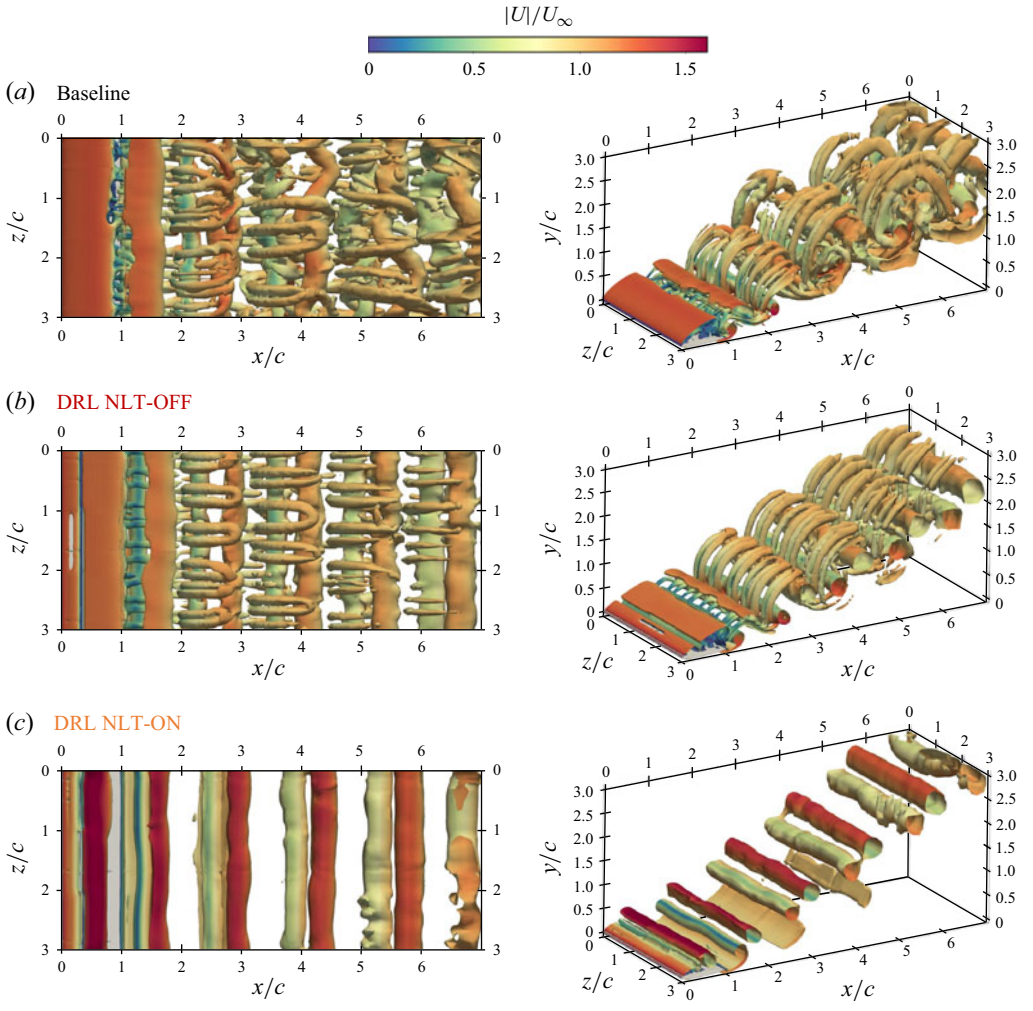


Figure 10. Visualisation of vortical structures using Q-criterion iso-contours $Qc^2/U_\infty^2 = 0.1$ coloured by velocity magnitude. (*a*) Baseline, (*b*) DRL-control under NLT-OFF and (*c*) DRL-control under NLT-ON. On the left, the top view; and on the right, the perspective view.

persistent vortex remains near the suction surface. As later confirmed in the dynamical analysis, this resonant excitation drives lift enhancement through maximisation of the lifting vortex, indicating that the frequency is actively targeted by the DRL agent.

The fact that the DRL agent explicitly targets well-established control mechanisms reported in the literature indicates that it can identify and exploit physically meaningful strategies despite its data-driven nature. Moreover, the resulting momentum coefficients are consistent with those reported in prior studies (see § 3.4), although they are slightly higher, likely due to increased viscous effects associated with the lower Reynolds numbers considered here.

In figure 10, the vortical structures of the controlled scenarios in the statistically steady state are compared against the baseline case using the Q-criterion iso-contours. In the uncontrolled scenario (figure 10*a*), the periodic pattern of the rib-like structures of mode B is broken, and no clear periodicity is observed along the z-direction. The high angle

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

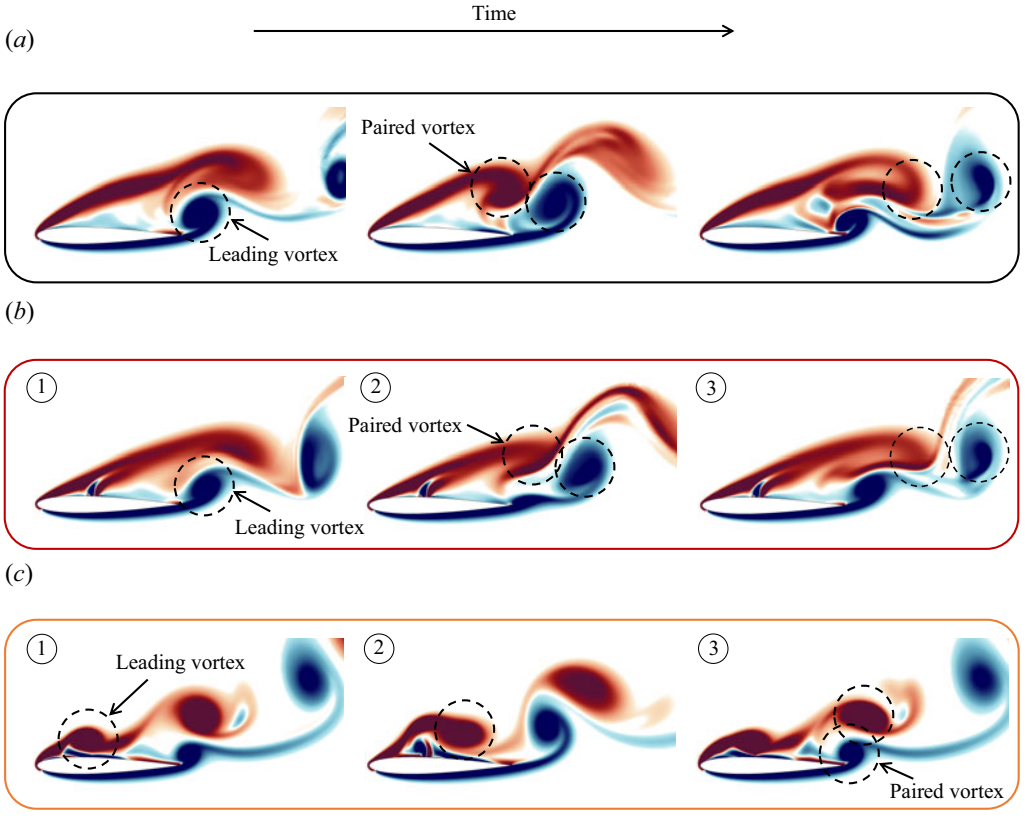


Figure 11. z-Vorticity contours $\omega_z c/U_\infty \in [-10, 10]$ in the x–y plane over a vortex-shedding cycle for the (a) baseline uncontrolled scenario and the DRL-controlled scenarios under (b) NLT-OFF and (c) NLT-ON. Frame numbers 1, 2 and 3 in (b) and (c) correspond to the times highlighted in figures 8(a) and 8(c), respectively.

of attack leads to a significant separation, creating a low-pressure region that induces the formation of a vortex near the trailing edge (leading vortex), which partially entrains fluid above the suction side of the wing. This vortex interacts with the shear layer originating at the leading edge, leading to the generation of a new vortex (paired vortex). This process is illustrated in figure 11(a), which shows the temporal evolution of the flow over a shedding cycle. This figure also presents the same analysis for the two controlled cases, with each snapshot corresponding to the instants indicated as black dashed lines in figure 8.

Under NLT-OFF, the wake exhibits a more organised pattern of rib-like structures, with a well-defined periodicity along the invariant direction compared with the baseline, as observed in figure 10(b). This transition from a massive, separated wake to a more confined, organised structure explains the emergence of the regular periodic patterns in the lift signal and is analogous to reducing the angle of attack, where homogeneous von Kármán vortex rows are recovered, and chaotic features are no longer observed. In this scenario, the detached leading vortex forms further downstream of the trailing edge due to the deflection of the shear layer towards the wing wall (see figure 11(b), left). Consequently, the wake is elongated along the streamwise direction, and the paired vortices are less separated vertically, which explains the reduction of the vortex-shedding amplitude and its shift to higher frequencies (see figure 11(b), right). By partially

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

mitigating the adverse effects of vortex shedding, this mechanism leads to a reduction in fluctuations and, ultimately, in drag.

For NLT-ON, the flow exhibits a nearly 2-D behaviour, with well-defined vortex tubes along the z-direction, as observed in figure 10(c). This is consistent with the nearly 2-D control actions applied by the agent in figure 8(c). However, the mechanism of formation of these vortices differs from the NLT-OFF and baseline scenarios. Here, the paired vortex is primarily generated by the oscillatory behaviour of the jet actions rather than by the breakdown of the shear layer induced by the primary vortex. When the front jet blows, it lifts the shear layer at the leading edge, creating a small low-pressure region here (figure 11(c), left). As a result, the shear layer begins to roll up, forming a vortex that subsequently breaks down when the front jet reverses to suction and the rear jet to blowing (figure 11(c), centre). This vortex is then convected downstream with the mean flow, generating a counter-rotating vortex as it passes the trailing edge (figure 11(c), right). Therefore, by contrast to what was observed before, in this case, the leading-edge vortex acts as the precursor to the trailing-edge vortex. This shift in the vortex-shedding dynamics, actively enforced by the DRL controller, supports the presence of a resonance mechanism underlying the control strategy, with the controller artificially creating prominent lifting vortices.

3.4. *Classic control*

To further investigate the benefits of DRL control, different classical control actuations are applied to the same wing configuration. Again, two jets are located at the leading edge and mid-chord locations, as defined in § 2.1. In this study, however, a homogeneous jet velocity is prescribed along the spanwise direction. The fixed control law used for the front jet is defined as

$$\frac{U_{jet}}{U_\infty} = \frac{A}{U_\infty} \sin\left(2\pi f \frac{U_\infty}{c} t\right) + \frac{A_{DC}}{U_\infty}, \tag{3.1}$$

while, similar to the DRL scenario, the rear jet instantaneously applies the opposite mass-flow rate. In (3.1), three key parameters need to be specified: the actuation amplitude A, the actuation frequency f and the steady component of the signal A_{DC}. To enable a comparison with the AFC strategy learned by the DRL agents, a range of parameters will be tested, following a traditional parametric study approach. The target frequencies are selected based on the vortex-shedding frequency of the uncontrolled scenario St_0, exploring values around this baseline, specifically $fU_\infty/c = [0.8, 1.0, 1.2]\, St_0$. For the actuation amplitude A and the steady component A_{DC}, values are chosen relative to the maximum jet velocity explored by the DRL agents, i.e. $(U_{jet}/U_\infty)_{max} = 1.125$. Specifically, the amplitude A is tested at 100 %, 50 % and 25 % of this maximum, while the steady component A_{DC} is tested at −50 %, 0 % and 50 %.

Moreover, classical AFC actuations derived from the DRL solutions are also investigated. Specifically, the statistical characteristics of the DRL jet signals shown in figure 8, namely the dominant frequency (f), mean jet velocity (A_{DC}) and standard deviation, are used to define an actuation following (3.1). More precisely, since a sinusoidal law is employed, the amplitude (A) is selected such that the standard deviation of the resulting signal matches that of the DRL jet signal. The same two-jet configuration described previously is used. In addition, a simplified actuation case considering only the front leading-edge (LE) jet is also examined.

The statistics from all these simulations are collected over $50U_\infty/c$ convective time units after the initial transient, as was done previously for the DRL-controlled scenarios, which is sufficient for statistical convergence. Figure 12(a) presents the C_l and C_d

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

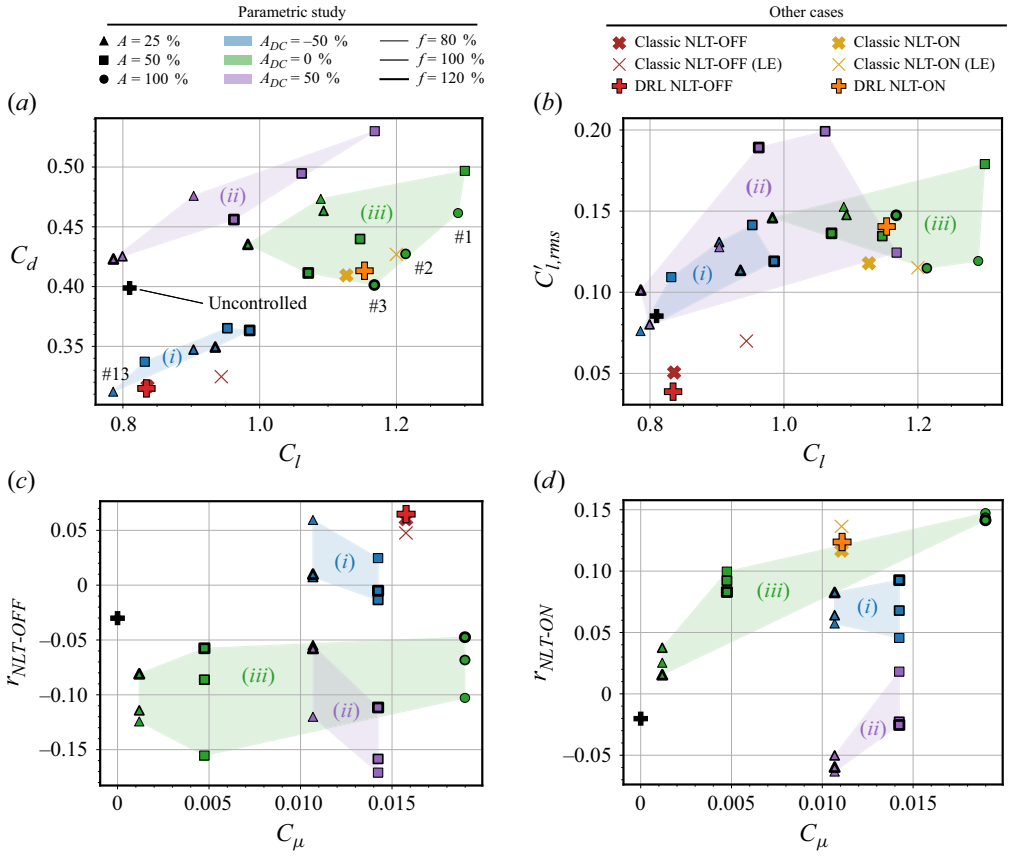


Figure 12. Results of the classic control parametric study compared against the uncontrolled baseline scenario (black '+' markers), the DRL-controlled cases under NLT-OFF (red '+' markers) and NLT-ON (orange '+' markers) and the classic controls derived from the corresponding DRL cases (red and orange '×' markers): (*a*) lift coefficient versus drag coefficient; (*b*) lift coefficient versus r.m.s. of lift coefficient fluctuations; and momentum coefficient versus local reward under (*c*) NLT-OFF and (*d*) NLT-ON.

coefficients obtained for all classic control cases, compared with the results from the DRL strategies and the uncontrolled baseline case, while figure 12(*b*) further illustrates the $C'_{l,rms}$. The obtained results are quantitatively reported in Appendix E.

From the parametric study, three distinct regimes can be identified based on the steady DC component. (i) When only suction is applied ($A_{DC} < 0$), the largest drag reduction is achieved while maintaining lift and fluctuation levels close to the baseline. (ii) In contrast, blowing alone ($A_{DC} > 0$) generally results in higher drag coefficients with intermediate lift values and a broader range of fluctuations depending on the applied amplitude A. (iii) Finally, periodic suction and blowing ($A_{DC} = 0$) yields the highest lift coefficients, although this is typically accompanied by increased drag and fluctuation levels, albeit still generally lower than in the second regime.

Regimes (i) and (iii) correspond to the two mechanisms already described in the literature for wake stabilisation and stall delay, respectively, and are consistent with those identified by the DRL agent under the corresponding reward settings (NLT OFF and NLT ON). On the other hand, regime (ii) does not offer any advantage over the other two.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

Concerning the other parameters studied, higher amplitudes generally lead to increased lift coefficients, most likely due to the increased momentum coefficient, while the effect of frequency depends on the actuation type. For (ii) blowing and (iii) combined suction–blowing cases, increasing the frequency results in lower lift and drag coefficients, with the lift-to-drag ratio remaining approximately constant. As noted in the literature, the frequency in the suction–blowing regime is one of the main features of the actuation, although no significant differences are observed as long as f remains close to the natural vortex-shedding frequency. Conversely, for (i) suction-only actuation, the opposite trend is observed, and increasing the frequency leads to higher lift and drag coefficients, accompanied by larger fluctuations. This suggests that the main benefits arise from the steady component of the signal rather than from the frequency component, as discussed in § 3.4. In this case, the objective is to delay the separation point rather than to periodically interact with the lifting vortex through resonance mechanisms.

When comparing these results with the DRL solutions, it can be observed that both DRL scenarios achieve very promising results. DRL NLT-OFF lies within regime (i), yielding one of the lowest drag coefficients and the lowest lift fluctuations. On the other hand, DRL NLT-ON is located close to regime (iii), where maximum lift coefficients are achieved, with intermediate values of lift fluctuations and drag.

Given the multi-parameter nature of the optimisation, figures 12(c) and 12(d) are also presented. These figures show the time-averaged reward r obtained by each case when using the α and β parameters defined for NLT-OFF and NLT-ON, respectively. These are the quantities that were targeted for optimisation and therefore provide the real balance among the three terms C_l, C_d and $C'_{l,rms}$. These quantities are plotted against the momentum coefficient C_μ to further assess the performance efficiency achieved by each case.

It can be seen that the DRL agents trained under NLT-OFF obtain the highest $r_{NLT\text{-}OFF}$ reward compared with all other classic control cases. Consequently, it is among the cases with the lowest drag coefficient and the smallest lift fluctuations, which aligns with the main objective of this reward, the wake stabilisation.

In comparison with the parametric study, case ID 13 in table 9 shows similar performance as the DRL NLT-OFF scenario. This case exhibits a slightly lower momentum coefficient but also a reduction in C_l and, notably, an increase in $C'_{l,rms}$. Hence, it leads to lower aerodynamic efficiency and significantly higher fluctuations of the aerodynamic forces. This case consistently corresponds to steady suction ($A_{DC} < 0$) at the front jet and small amplitude A of the jet signal, and thus qualitatively resembles the DRL strategy.

When applying the classical control based on the DRL actuation, i.e. classic NLT-OFF and classic NLT-OFF (LE), very similar results are obtained, thereby demonstrating that the DRL is able to learn physically interpretable flow-control strategies that can be reduced to simpler actuation laws. For the same jet configuration, the results are closely matched, with the DRL-based case yielding slightly lower r.m.s. fluctuations. When only the LE jet is considered, the lift fluctuations increase. This indicates that, despite the similar global performance metrics, the rear blowing jet contributes to additional wake stabilisation.

As observed in figure 13(a), the rear blowing jet energises the near-wake region and encourages a more rapid convergence of the external streamlines. This partially suppresses the vertical entrainment of the free stream between the two vortex cores and shifts the trailing-edge vortex upward, thereby reducing the vertical distance between the two vortices. As a result, the dual-jet configuration achieves a significantly narrower wake and smoother pressure recovery compared with the case with LE suction alone.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

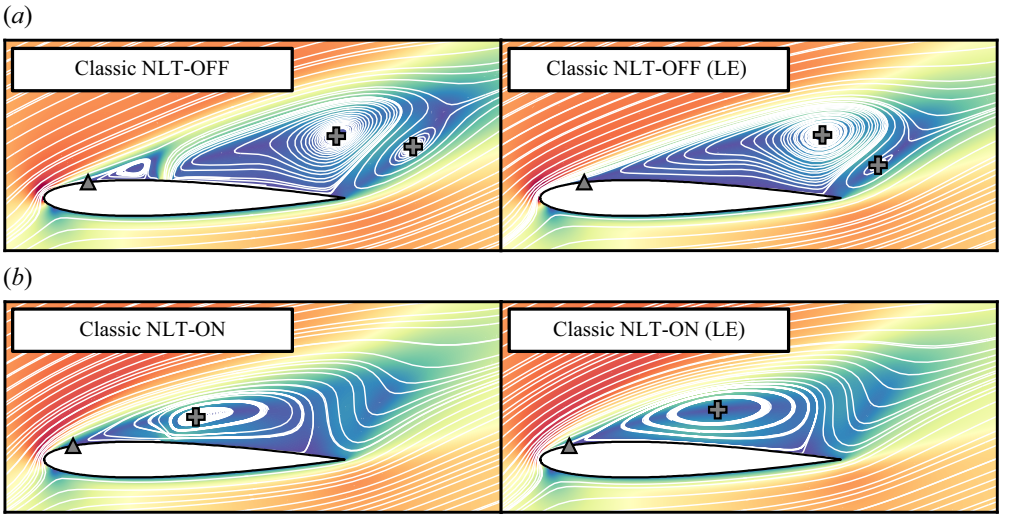


Figure 13. Time-averaged velocity streamlines with the velocity magnitude shown in the background. Classical control cases based on DRL NLT-OFF are shown in (*a*), while those derived from DRL NLT-ON are shown in (*b*). Crosses indicate the locations of the vortex cores, and triangles denote the flow separation point at the LE.

For the DRL strategy trained under NLT-ON, some classic control cases from the parametric study (case IDs 1, 2 and 3 in table 9) show slightly higher reward $r_{NLT\text{-}ON}$ values. In PPO-based methods, this may occur because the agent can converge to local optima, as the clipped policy update may limit exploration and prevent escape from suboptimal regions in the parameter space. However, the differences in reward are very small, and the resulting performance is highly similar.

These parametric cases correspond to those with the highest amplitude and no steady component ($A_{DC} = 0$). Accordingly, they also exhibit higher momentum coefficients. In contrast, the DRL-controlled scenario achieves approximately half the momentum coefficient while maintaining a comparable balance between lift, drag and r.m.s. of lift fluctuations, as reflected by $r_{NLT\text{-}ON}$. Therefore, the steady components present in the DRL jet signals, and absent in these cases, appear to be beneficial in reducing the required momentum coefficient while maintaining similar performance.

When analysing the classical strategies derived from the DRL results, again these yield results very similar to those obtained with the DRL controller. In this case but, the configuration using only the LE jet exhibits a slightly higher reward. As reported in the literature, periodic suction–blowing actuation at the LE is commonly used to increase the lift coefficient and thereby delay stall, which explains the favourable performance of this configuration. As observed in figure 13(*b*), the rear jet in the NLT-OFF regime is located very close to the lifting vortex core, perturbing and preventing its proper development, thereby leading to a slight deterioration in performance.

Nevertheless, the two-jet configuration provides greater flexibility, with only a minor degradation in performance. From a practical point of view, the required mass-flow rate must be sourced from an external supply. While this is inherently satisfied in synthetic jet actuation, it is not the case when steady components are involved. The present configuration addresses this issue, enabling the possibility of switching between DRL-ON and DRL-OFF regimes and changing the operational objective during flight.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

4. Conclusions

This study investigates the use of DRL for AFC in a 3-D NACA0012 wing operating at a low Reynolds number ($Re = 1000$) and a high angle of attack (20°) under spanwise-periodic boundary conditions, a regime characterised by strong separation and complex 3-D wake structures. The baseline flow is first thoroughly validated against existing 3-D simulations, confirming that the numerical set-up accurately captures the key aerodynamic features.

Two distinct DRL reward formulations are considered, which lead to substantially different control outcomes. Under NLT-OFF, which focuses primarily on drag reduction and flow stabilisation, the agent learns to apply quasi-steady suction at the LE and blowing at the rear jet. This control strategy effectively delays the separation point and maintains the shear layer closer to the wing surface, constraining the entrainment of the leading vortex and promoting wake elongation with partial suppression of the vortex-shedding mechanism. The resulting wake exhibits a more coherent spanwise organisation, consistent with enhanced stabilisation of the flow. Additionally, the rear jets are found to further stabilise the wake by providing additional momentum. This control results in a 21.1 % reduction in drag and a 54.1 % decrease in lift fluctuations, while the mean lift remains essentially unchanged, with a minor increase of 2.96 %.

In contrast, NLT-ON places greater emphasis on lift enhancement, producing a markedly different control strategy. Here, the agent applies both suction and blowing at the LE in a more synchronised manner. This alternating jet actuation directly interacts with the shear layer at the LE, generating a travelling vortex moving along the suction side. Unlike the baseline and NLT-OFF cases, where vortex shedding is initiated downstream, here, the shear layer is forced to roll up at the LE, effectively initiating the shedding process upstream and hence resembling the resonance mechanism commonly associated with synthetic jets. As a result, a nearly 2-D wake develops. This strategy increases mean lift by 42.3 %, at the cost of a 65.9 % increase in lift fluctuations and a slight 3.5 % increase in drag. In this case, the rear jet slightly deteriorates the performance of the actuation, although it provides greater flexibility for combination with other actuation mechanisms.

Comparison with classical open-loop strategies shows that DRL achieves comparable performance, which is not unexpected given the simplicity of the present weakly turbulent regime, where control mechanisms are well established. Nevertheless, this provides confidence in extending DRL to more complex flows. Despite being purely data driven, the learned policies converge towards actuation mechanisms consistent with optimal controls reported in the literature, demonstrating that DRL can identify physically meaningful control laws without explicit prior knowledge of the flow physics. This highlights its potential for more complex configurations where such prior knowledge is limited, particularly in regimes where feedback control and multi-frequency actuation are expected to provide additional benefits under varying or more turbulent conditions.

Overall, this study presents the first DRL training on a fully 3-D wing configuration, marking a step forward toward the use of DRL in realistic fluid dynamics scenarios in which the agent can adaptively select control strategies in real time according to the external conditions and prescribed objectives. However, the results also highlight a strong sensitivity of the control strategy to the reward formulation, where even minor variations in reward weighting can lead to fundamentally different behaviours, with limited physical intuition currently available to guide parameter selection. This underscores the necessity

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

Mesh refinement	N_{xy}	N_z	N_{xyz}	C_l	$C'_{l,rms}$	C_d	St
Coarse	67.1×10^4	49	3.29×10^6	0.794	0.078	0.393	0.55
Medium	1.05×10^5	73	7.6×10^6	0.810	0.085	0.399	0.56
Fine	1.66×10^5	109	18.0×10^6	0.802	0.082	0.396	0.56

Table 5. Mesh characteristics and resulting aerodynamic coefficients, including the vortex-shedding Strouhal number, for each refinement level considered.

for a systematic framework in reward function design in order to reliably achieve targeted performance objectives in flow-control problems.

Acknowledgements. The authors acknowledge the Red Española de Supercomputación (IM-2024-2-0004 and IM-2024-3-0005) and the European High Performance Joint Undertaking (EHPC-REG-2024R01-038) for providing the computational resources used in this work, granting access to the MareNostrum V supercomputer at the BSC.

Funding. This research was partially funded by the Ministerio de Ciencia e Innovación of Spain (PID2023-150408OB-C21 and PID2023-150408OB-C22). Work has also been co-funded by the European High Performance Computing Joint Undertaking, together with Germany, Italy, Slovenia, Spain, Sweden and France under grant agreement No 101092621; and under grant PCI2022-135085-2, funded by MICIU/AEI/10.13039/501100011033 and by the 'European Union NextGenerationEU/PRTR'. R. Montalà's work was supported by a contract from the Agència d'Ajuts Universitaris i Recerca of the Generalitat de Catalunya through a FI-SDUR grant (2022 FISDU 00066). O. Lehmkuhl has been partially supported by a Ramón y Cajal postdoctoral contract (RYC2018-025949-I). The authors also acknowledge the support of the Agència d'Ajuts Universitaris i Recerca of the Generalitat de Catalunya to the research groups Large-scale Computational Fluid Dynamics (2021 SGR 00902) and Turbulence and Aerodynamics Research Group (2021 SGR 01051). R. Vinuesa acknowledges financial support from the European Research Council under grant no. 2021-CoG-101043998, DEEPCONTROL. The views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Union or the European Research Council.

Declaration of interests. The authors report no conflict of interest.

Data availability statement. The data that support the findings of this study are available from the corresponding author upon reasonable request.

Appendix A. Mesh sensitivity and temporal convergence analyses

To assess the independence of the numerical results from the mesh resolution, three different mesh refinement levels were analysed. The main characteristics of each mesh, along with the resulting aerodynamic coefficients, are summarised in table 5. As can be observed, the three mesh levels yield very similar aerodynamic statistics, indicating that the results are essentially mesh-independent. Based on this analysis, the medium mesh is selected for the present study, and all results presented hereafter were obtained using this grid.

In addition, a convergence analysis of the aerodynamic time series was performed using progressively larger backward-facing time windows (Δt^*) to evaluate the accumulated average and r.m.s. of C_d and C_l. As shown in figure 14 for all the cases under study, the final aerodynamic quantities are statistically converged.

Appendix B. Impact of the number of parallel environments on DRL training speed

As proposed by Rabault & Kuhnle (2019), training can be accelerated by collecting trajectories from n_{env} parallel environments, thus increasing the volume of training data per episode. In a 3-D geometry with one invariant direction, n_{env} can be increased via two

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

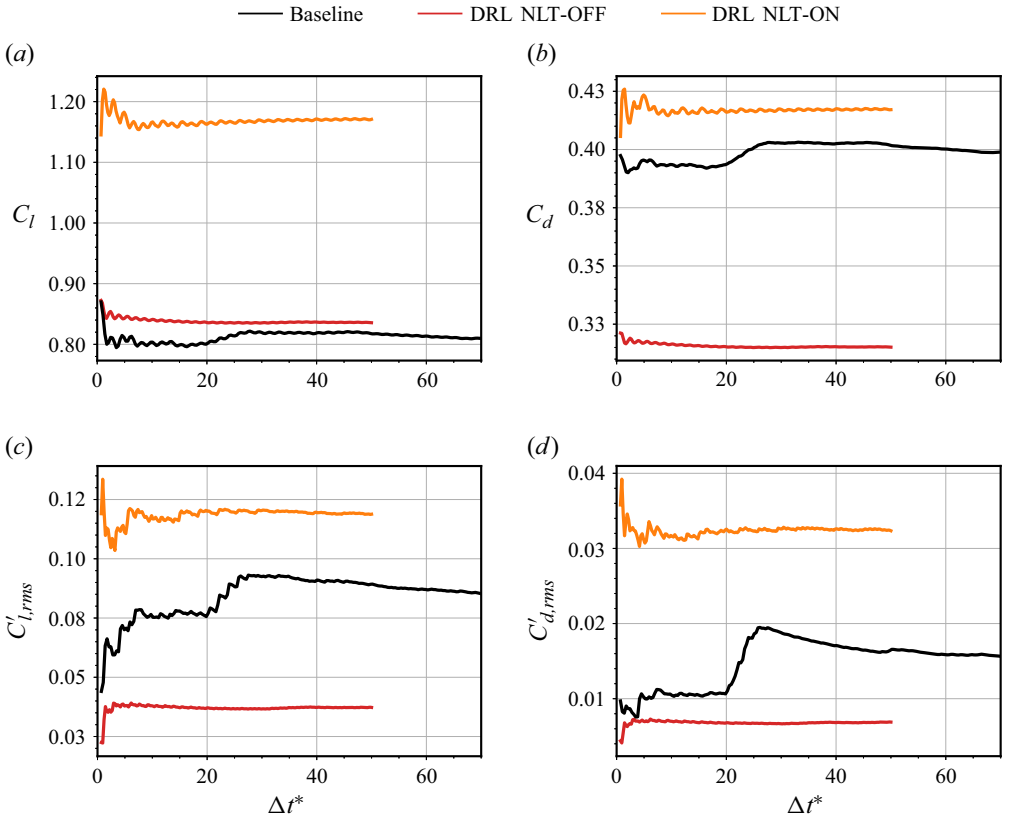


Figure 14. Temporal convergence analysis for the three cases analysed: baseline, DRL NLT-OFF and DRL NLT-ON. Evolution of the (*a*) lift coefficient, (*b*) drag coefficient, (*c*) r.m.s. of the lift coefficient fluctuations and (*d*) r.m.s. of the drag coefficient fluctuations computed using backward-moving windows of length Δt^* from the final simulation time. The window length is $\Delta t^* = (t_{end} - t)\, U_\infty / c$, and the accumulated mean is $\phi(\Delta t^*) = (1/\Delta t^*) \int_0^{\Delta t^*} \phi(t)\, \mathrm{d}t$.

strategies: (i) executing n_{CFD} parallel simulations, or (ii) increasing the number of n_{MARL} environments, resulting in a total of $n_{env} = n_{CFD} \times n_{MARL}$ parallel environments.

This section investigates the impact of n_{env} on training speed-up. For physical consistency, each MARL environment maintains a constant spanwise width $L_{z,MARL}/c = 1.0$; thus, the total domain length L_z/c scales proportionally with n_{MARL}. Mesh resolution is also fixed across all runs. To isolate solver parallelisation effects, the DoF per GPU are held constant at 7.6×10^6 DoF/GPU, meaning larger L_z requires more GPUs per CFD simulation.

The considered cases are listed in table 6, with all training runs reaching 2016 MARL episodes. Trainings are conducted under NLT-OFF reward ($\alpha = 0.3$ and $\beta = 0.0$ in (2.5)). Unlike the main training runs in § 3.2, which used four GPUs, this scaling analysis allocates only one GPU per CFD simulation, resulting in a longer wall-clock time per episode.

The total wall-clock time for each DRL training is shown in figure 15. When varying n_{CFD}, the scaling behaviour is nearly ideal. The episode duration remains approximately constant at around 27 minutes. Consequently, because fewer training steps are needed to reach the target 2016 MARL episodes, the overall training wall-clock time reduces proportionally, exhibiting an almost linear decrease.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

n_{CFD}	n_{MARL}	n_{env}	L_z/c	Mesh N_{xyz} (M)	Total GPUs	Training steps	GPU-hours
4	3	12	3.0	7.6	4	168	302
8	3	24	3.0	7.6	8	84	302
16	3	48	3.0	7.6	16	42	304
32	3	96	3.0	7.6	32	21	307
8	6	48	6.0	15.2	16	42	331
8	12	96	12.0	30.4	32	21	347

Table 6. Training cases performed for the DRL scaling study when the number of CFD simulations or MARL pseudo-environments is varied to increase the total number of environments, while maintaining the same number of MARL episodes. The spanwise domain length for each CFD simulation is given by L_z/c, while N_{xyz} denotes the mesh size. The total amount of GPUs used for each training is also included, along with the number of training steps conducted, and the corresponding computational cost in GPU-hours.

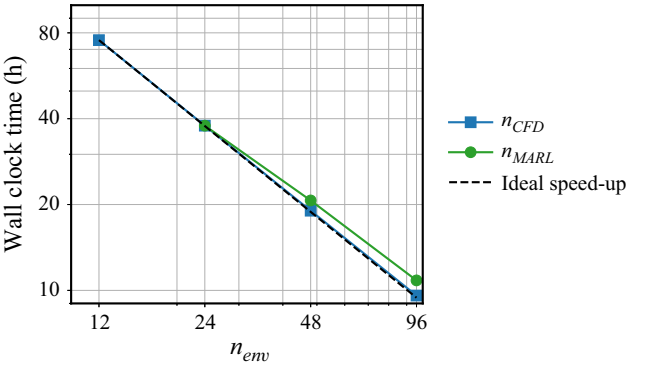


Figure 15. Total wall-clock time (in hours) required for training as a function of the number of CFD simulations and MARL pseudo-environments.

In contrast, while increasing n_{MARL} also accelerates training, the scaling deviates slightly from the ideal case. As discussed earlier, a larger n_{MARL} requires allocating extra GPUs per CFD simulation. This leads to greater communication overhead, which partially offsets the scaling benefits. Consequently, the approximate wall-clock time per episode increases from 27 minutes (using 1 GPU per CFD simulation) to 31 minutes (using 4 GPUs per CFD simulation).

To examine the convergence of reward during training, figure 16 shows the evolution of this quantity. In figure 16(*a*), the total number of MARL episodes is directly proportional to the computational cost of the training (CPU time). On the other hand, figure 16(*b*) presents the evolution of the reward as a function of the number of training steps, obtained by dividing the number of MARL episodes by the number of parallel environments. This quantity is then proportional to the wall-clock time.

All cases reach approximately the same reward by the end of the training. However, this methodology has inherent limitations. Increasing n_{env} enables the agent to make more informed decisions at each training step, thus following a more direct path toward the optimal policy and reducing inefficient exploration. Nevertheless, the step size in the optimisation process (primarily governed by the learning rate) remains fixed. Therefore, beyond a certain point, further increasing the number of experiences collected per training

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

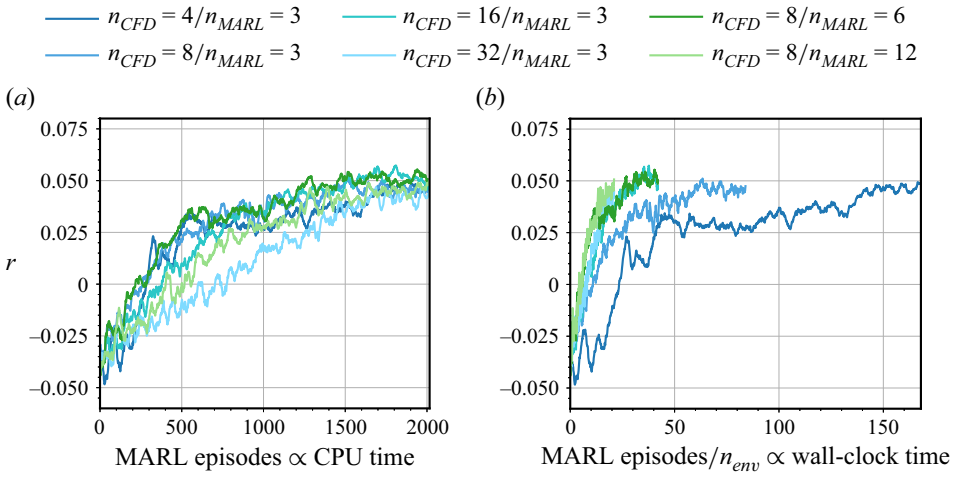


Figure 16. Local reward evolution during training when varying the number of CFD simulations (in blue) and the number of MARL pseudo-environments (in green).

step will not produce additional improvements in training speed, although this limit has not yet been reached in the present study.

Appendix C. Mathematical formulation of the PPO agent

As an actor-critic method, PPO employs an additional network, referred to as the critic, which approximates the value function $V_\theta(s_t)$. This function estimates the expected return under the current policy from time step t onwards

$$V_\theta(s_t) \approx \mathbb{E}_\pi \left[G_t \mid s_t \right]. \tag{C1}$$

This is then used to evaluate the quality of actions taken through the computation of the advantage A_t, defined as

$$A_t = G_t - V_\theta(s_t). \tag{C2}$$

Therefore, by comparing the empirical return G_t with expected return under the current policy $V_\theta(s_t)$, it quantifies how much better (or worse) the action was relative to the mean behaviour of the policy, guiding the policy to reinforce actions with high advantages. In this work, to reduce the high variance associated with the use of the full empirical return G_t, the GAE is adopted. The GAE combines information from multiple-step returns, weighted by a factor $\lambda \in [0, 1]$, thereby balancing bias and variance. For the present computations, $\lambda = 0.95$. This approach yields smoother and more stable advantage estimates

$$A_t \approx \sum_{l=0}^{\infty} (\gamma\lambda)^l \, \delta_{t+l}, \tag{C3}$$

$$\delta_t = R_{t+1} + \gamma V_\theta(s_{t+1}) - V_\theta(s_t). \tag{C4}$$

The total loss function maximised in PPO is

$$L(\theta) = \mathbb{E}_t \left[L^{CLIP}(\theta) - c_1 L^{VF}(\theta) + c_2 S[\pi_\theta](s_t) \right], \tag{C5}$$

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

where

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[\min \left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} A_t,\ \text{clip} \left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)},\ 1-\epsilon,\ 1+\epsilon \right) A_t \right) \right], \quad \text{(C6)}$$

$$L^{VF}(\theta) = \mathbb{E}_t \left[\left(V_\theta(s_t) - V_t^{target} \right)^2 \right]. \quad \text{(C7)}$$

The first term in the total loss defined in (C5) corresponds to the clipped surrogate loss, which is used to train the actor by increasing the probability of actions with positive advantage ($A_t > 0$) and decreasing the probability of those with negative advantage ($A_t < 0$). The PPO algorithm improves on earlier policy-gradient methods by introducing this clipped objective function, which constrains policy updates to prevent large and destabilising changes in behaviour. The second term in (C5) is the squared error between the predicted value and a target value, used to train the critic. This term is weighted by the coefficient c_1, which is set to 0.5 in this work. Finally, the last term, weighted by the coefficient c_2, is an entropy bonus that encourages exploration by preventing premature convergence to deterministic policies and maintaining diversity in action selection during training. Here, $c_2 = 0.0$, which means that no additional entropy regularisation is applied to the loss.

Appendix D. Reproducibility of DRL training under stochastic effects

Due to its policy sampling and exploration mechanisms, the PPO agent is inherently stochastic. In principle, fixing the seed of all random processes during training should yield reproducible results. However, the use of GPUs to perform the CFD simulations introduces an additional source of stochasticity. The GPU-based computations involve non-deterministic operations such as parallel reductions and floating-point arithmetic, which can result in small numerical differences that accumulate over time.

As a result, even when running a baseline configuration, the results may start to diverge, although the overall flow statistics are maintained. This effect is amplified when the DRL agent interacts with the CFD environment because the resulting next state depends on the perturbed CFD solution, which in turn depends on the current state. Consequently, the sequence of observed transitions $(s_t, a_t, R_{t+1}, s_{t+1})$ can differ between training runs.

In this section, a reproducibility analysis is performed by repeating the same training multiple times with identical parameters and evaluating the results obtained. In this case $n_{CFD} = 10$ and $n_{MARL} = 3$. Trainings are performed using the two rewards discussed in this work, with $\alpha = 0.3$, $\beta = 0.0$ for NLT-OFF, and $\alpha = 0.3$, $\beta = 0.5$ for NLT-ON.

The evolution of the reward over five independent training runs for each reward is shown in figure 17. It is observed that after 2010 MARL episodes, the reward for each case converges to similar values. Once the learned policy is evaluated, the aerodynamic coefficients are summarised in table 7. As in § 3, the statistics are computed over 50 vortex-shedding cycles of the baseline flow to ensure convergence.

Using these five samples, the mean values μ and standard deviation σ of the main aerodynamic coefficients across the five runs are reported in table 8. For DRL NLT-OFF, the coefficient of variation (CV $= \sigma/\mu \times 100$) is 3.3 % for C_l, 30.3 % for $C'_{l,rms}$ and 2.6 % for C_d; whereas in DRL NLT-ON, the corresponding values are 6.9 %, 9.3 % and 1.9 %, respectively. These values confirm that the variability across runs is small, with the largest differences appearing in the unsteady lift fluctuations for DRL NLT-OFF.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

Training run	DRL NLT-OFF			DRL NLT-ON		
	C_l	$C'_{l,rms}$	C_d	C_l	$C'_{l,rms}$	C_d
1	0.834	0.048	0.316	1.167	0.144	0.412
2	0.886	0.097	0.333	1.082	0.157	0.393
3	0.810	0.052	0.311	1.163	0.148	0.416
4	0.841	0.063	0.322	1.044	0.122	0.385
5	0.845	0.060	0.321	0.995	0.137	0.374

Table 7. Aerodynamic coefficients obtained by evaluating the NLT-OFF and NLT-ON policies learned from five independent DRL training runs with identical parameters.

	μ_{C_l}	σ_{C_l}	$\mu_{C'_{l,rms}}$	$\sigma_{C'_{l,rms}}$	μ_{C_d}	σ_{C_d}
DRL NLT-OFF	0.843	0.028	0.064	0.019	0.321	0.008
DRL NLT-ON	1.090	0.075	0.142	0.013	0.396	0.008

Table 8. Mean μ and standard deviation σ of aerodynamic coefficients computed across the five independent runs shown in table 7 for the NLT-OFF and NLT-ON policies.

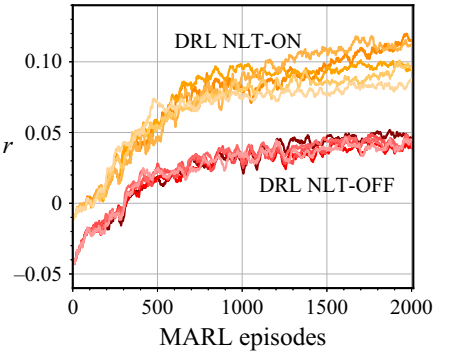


Figure 17. Training curves showing the evolution of the local reward for the NLT-OFF and NLT-ON policies. Five independent training runs are performed for each policy using identical training parameters; NLT-OFF results are shown in red, and NLT-ON in orange.

Appendix E. Results from the classic control study

Table 9 reports the main aerodynamic metrics, namely C_l, $C'_{l,rms}$ and C_d, obtained when applying the classical control strategies discussed in § 3.4 of the manuscript. Additionally, these results are compared against the baseline scenario, showing the relative difference for each quantity, i.e. $\Delta C_x = (C_x - C_{x,\,base})/C_{x,\,base}$ (%). Finally, the momentum coefficient of each actuation strategy, based on the $U_{jet,rms}$, is also provided.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

Case ID	A/U_∞	A_{DC}/U_∞	fU_∞/c	C_l	$C'_{l,rms}$	C_d	ΔC_l (%)	$\Delta C'_{l,rms}$(%)	ΔC_d (%)	C_μ
1	1.125	0	0.45	1.290	0.119	0.462	59.3	39.8	15.7	0.0190
2	1.125	0	0.56	1.213	0.115	0.427	49.8	34.6	7.15	0.0190
3	1.125	0	0.67	1.168	0.147	0.401	44.2	72.8	0.63	0.0190
4	0.563	−0.563	0.45	0.832	0.109	0.337	2.72	28.1	−15.5	0.0142
5	0.563	−0.563	0.56	0.953	0.142	0.365	17.7	65.8	−8.44	0.0142
6	0.563	−0.563	0.67	0.985	0.119	0.363	21.7	39.6	−8.90	0.0142
7	0.563	0	0.45	1.300	0.179	0.497	60.5	110	24.6	0.0047
8	0.563	0	0.56	1.147	0.135	0.440	41.6	57.8	10.3	0.0047
9	0.563	0	0.67	1.071	0.136	0.411	32.3	59.8	3.15	0.0047
10	0.563	0.563	0.45	1.168	0.125	0.530	44.3	45.9	32.9	0.0142
11	0.563	0.563	0.56	1.062	0.199	0.495	31.1	133	24.0	0.0142
12	0.563	0.563	0.67	0.962	0.189	0.456	18.8	121	14.4	0.0142
13	0.281	−0.563	0.45	0.786	0.076	0.312	−2.97	−10.8	−21.7	0.0107
14	0.281	−0.563	0.56	0.903	0.131	0.347	11.5	53.5	−12.9	0.0107
15	0.281	−0.563	0.67	0.935	0.114	0.350	15.5	33.1	−12.4	0.0107
16	0.281	0	0.45	1.089	0.153	0.473	34.5	79.0	18.7	0.0012
17	0.281	0	0.56	1.094	0.148	0.463	35.0	73.1	16.2	0.0012
18	0.281	0	0.67	0.983	0.146	0.435	21.4	71.1	9.16	0.0012
19	0.281	0.563	0.45	0.904	0.128	0.476	11.6	49.8	19.3	0.0107
20	0.281	0.563	0.56	0.799	0.080	0.425	−1.29	−5.97	6.66	0.0107
21	0.281	0.563	0.67	0.786	0.101	0.423	−2.92	18.8	6.09	0.0107
NLT-OFF	0.072	−0.723	0.64	0.836	0.051	0.317	3.23	−40.7	−20.6	0.0158
NLT-OFF (LE)	0.072	−0.723	0.64	0.944	0.070	0.325	16.6	−18.1	−18.6	0.0158
NLT-ON	0.824	−0.171	0.57	1.127	0.118	0.409	39.2	38.1	2.65	0.0111
NLT-ON (LE)	0.824	−0.171	0.57	1.200	0.115	0.427	48.2	35.0	7.04	0.0111

Table 9. Summary of the control parameters and corresponding results for each classic control case. The first twenty-one rows correspond to the parametric study, while the four bottom rows correspond to the classic controls derived from DRL actuations.

References

Amitay, M., Smith, D.R., Kibens, V., Parekh, D.E. & Glezer, A. 2001 Aerodynamic flow control over an unconventional airfoil using synthetic jet actuators. *AIAA J.* **39** (3), 361–370.

Atzori, M., Vinuesa, R., Fahland, G., Stroh, A., Gatti, D., Frohnapfel, B. & Schlatter, P. 2020 Aerodynamic effects of uniform blowing and suction on a naca4412 airfoil. *Flow, Turbul. Combust.* **105**, 735–759.

Bae, H.J. & Koumoutsakos, P. 2022 Scientific multi-agent reinforcement learning for wall-models of turbulent flows. *Nat. Commun.* **13**, 1443.

Barkley, D. & Henderson, R.D. 1996 Three-dimensional Floquet stability analysis of the wake of a circular cylinder. *J. Fluid Mech.* **322**, 215–241.

Belus, V., Rabault, J., Viquerat, J., Che, Z., Hachem, E. & Reglade, U. 2019 Exploiting locality and translational invariance to design effective deep reinforcement learning control of the 1-dimensional unstable falling liquid film. *AIP Adv.* **9** (12), 125014.

Beneitez, M., Cremades, A., Guastoni, L. & Vinuesa, R. 2025 Improving turbulence control through explainable deep learning. arXiv:2504.02354.

Buchmann, N.A., Atkinson, C. & Soria, J. 2013 Influence of ZNMF jet flow control on the spatio-temporal flow structure over a NACA-0015 airfoil. *Exp. Fluids* **54**, 1485.

Cavallazzi, G.M., Guastoni, L., Vinuesa, R. & Pinelli, A. 2025 Deep reinforcement learning for the management of the wall regeneration cycle in wall-bounded turbulent flows. *Flow Turbul. Combust.* **115**, 1291–1317.

Chatzimanolakis, M., Weber, P. & Koumoutsakos, P. 2024 Learning in two dimensions and controlling in three: generalizable drag reduction strategies for flows past circular cylinders through deep reinforcement learning. *Phys. Rev. Fluids* **9**, 043902.

Chen, C., Seele, R. & Wygnanski, I. 2012 Separation and circulation control on an elliptical airfoil by steady blowing. *AIAA J.* **50** (10), 2235–2247.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

FEERO, A.M., LAVOIE, P. & SULLIVAN, E.P. 2017 Influence of synthetic jet location on active control of an airfoil at low Reynolds number. *Exp. Fluids* **58**, 99.

FONT, B., ALCÁNTARA-ÁVILA, F., RABAULT, J., VINUESA, R., LEHMKUHL, O. 2025 Deep reinforcement learning for active flow control in a turbulent separation bubble. *Nat. Commun.* **16** (1), 1422.

FUCHS, L.M., VON SALDERN, J.G.R., KAISER, T.L. & OBERLEITHNER, K. 2024 Deep reinforcement learning-augmented Spalart–Allmaras turbulence model: application to a turbulent round jet flow. *Fluids* **9** (4), 88.

GARCIA, X., MIRÓ, A., SUÁREZ, P., ALCÁNTARA-ÁVILA, F., RABAULT, J., FONT, B., LEHMKUHL, O., VINUESA, R. 2025 Deep-reinforcement-learning-based separation control in a two-dimensional airfoil. *Intl J. Heat Fluid Flow* **116**, 109913.

GASPARINO, L., SPIGA, F. & LEHMKUHL, O. 2024 SOD2D: a GPU-enabled spectral finite elements method for compressible scale-resolving simulations. *Comput. Phys. Commun.* **297**, 109067.

GILARRANZ, J.L., TRAUB, L.W. & REDINIOTIS, O.K. 2005 A new class of synthetic jet actuators – part II: application to flow separation control. *J. Fluids Engng* **127** (2), 377–387.

GLEZER, A. & AMITAY, M. 2002 Synthetic jets. *Annu. Rev. Fluid Mech.* **34**, 503–529.

GUADARRAMA, S., *et al.* 2018 TF-Agents: a library for reinforcement learning in tensorflow. Available at: https://github.com/tensorflow/agents.

GUASTONI, L., RABAULT, J., SCHLATTER, P., AZIZPOUR, H. & VINUESA, R. 2023 Deep reinforcement learning for turbulent drag reduction in channel flows. *Eur. Phys. J. E* **46**, 27.

GUPTA, S., ZHAO, J., SHARMA, A., AGRAWAL, A., HOURIGAN, K. & THOMPSON, M.C. 2023 Two- and three-dimensional wake transitions of a NACA0012 airfoil. *J. Fluid Mech.* **954**, A26.

HOARAU, Y., BRAZA, M., VENTIKOS, Y., FAGHANI, D. & TZABIRAS, G. 2003 Organized modes and the three-dimensional transition to turbulence in the incompressible flow around a NACA0012 wing. *J. Fluid Mech.* **496**, 63–72.

JEON, J., RABAULT, J., VASANTH, J., ALCÁNTARA-ÁVILA, F., BARAL, S. & VINUESA, R. 2025 Inductive biased-deep reinforcement learning methods for flow control: group-invariant and positional-encoding networks improve learning reproducibility and quality. *Phys. Fluids* **37** (7), 077189.

KAMETANI, Y. & FUKAGATA, K. 2011 Direct numerical simulation of spatially developing turbulent boundary layers with uniform blowing or suction. *J. Fluid Mech.* **681**, 154–172.

KARNIADAKIS, G.E., ISRAELI, M. & ORSZAG, S.A. 1991 High-order splitting methods for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **97** (2), 414–443.

KENNEDY, C.A. & GRUBER, A. 2008 Reduced aliasing formulations of the convective terms within the Navier–Stokes equations for a compressible fluid. *J. Comput. Phys.* **227** (3), 1676–1700.

KINGMA, D.P. & BA, J. 2014 Adam: a method for stochastic optimization. arXiv:1412.6980.

KOKASH, H. & AGBAGLAH, G.G. 2024 A two- to three-dimensional wake transition mechanism induced by the angle of attack of a NACA0012 airfoil. *Metasci. Aerosp.* **1** (3), 329–345.

KOUSER, T., XIONG, Y., YANG, D. & PENG, S. 2021 Direct numerical simulations on the three-dimensional wake transition of flows over NACA0012 airfoil at $Re = 1000$. *Intl J. Micro Air Veh.* **13**, 17568293211055656.

KURTULUS, D.F. 2015 On the unsteady behavior of the flow around NACA, 0012 airfoil with steady external conditions at $Re = 1000$. *Intl J. Micro Air Veh.* **7** (3), 301–326.

KURZ, M., KAUSHIK, R., BLIND, M., KOPPER, P., SCHWARZ, A., RODACH, F. & BECK, A. 2025 Invariant control strategies for active flow control using graph neural networks. *Comput. Fluids* **303**, 106854.

KURZ, M., OFFENHÄUSER, P. & BECK, A. 2023 Deep reinforcement learning for turbulence modeling in large eddy simulations. *Intl J. Heat Fluid Flow* **99**, 109094.

KURZ, M., OFFENHÄUSER, P., VIOLA, D., RESCH, M. & BECK, A. 2022 Relexi – a scalable open source reinforcement learning framework for high-performance computing. *Softw. Impacts* **14**, 100422.

LI, R., ZHANG, Y. & CHEN, H. 2021 Learning the aerodynamic design of supercritical airfoils through deep reinforcement learning. *AIAA J.* **59** (10), 3988–4001.

LIU, Q., TRUJILLO-CORONA, L.J., SHU, F. & GROSS, A. 2025 Reinforcement learning-based closed-loop airfoil flow control. arXiv:2505.04818.

MCCORMICK, D. 2000 Boundary layer separation control with directed synthetic jets. In *38th AIAA Aerospace Sciences Meeting and Exhibit*. *AIAA Paper* 2000-0519.

MONTALÀ, R., FONT, B., SUÁREZ, P., RABAULT, J., LEHMKUHL, O., VINUESA, R. & RODRIGUEZ, I. 2024 Towards active flow control strategies through deep reinforcement learning. In *9th European Congress on Computational Methods in Applied Sciences and Engineering, Lisbon*. CIMNE (International Center for Numerical Methods in Engineering).

NOVATI, G., DE LAROUSSILHE, H.L. & KOUMOUTSAKOS, P. 2021 Automating turbulence modelling by multi-agent reinforcement learning. *Nat. Mach. Intell.* **3**, 87–96.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

O'Connor, J., Diessner, M., Wilson, K., Whalley, R.D., Wynn, A. & Laizer, S. 2023 Optimisation and analysis of streamwise-varying wall-normal blowing in a turbulent boundary layer. *Flow, Turbul. Combust.* **110**, 933–1021.

Paris, R., Beneddine, S. & Dandois, J. 2021 Robust flow control and optimal sensor placement using deep reinforcement learning. *J. Fluid Mech.* **913**, A25.

Paris, R., Beneddine, S. & Dandois, J. 2023 Reinforcement-learning-based actuator selection method for active flow control. *J. Fluid Mech.* **955**, A8.

Partee, S., Ellis, M., Rigazzi, A., Shao, A.E., Bachman, S., Marques, G. & Robbins, B. 2022 Using machine learning at scale in numerical simulations with SmartSim: an application to ocean climate modeling. *J. Comput. Sci.* **62**, 101707.

Patera, A.T. 1984 A spectral element method for fluid dynamics: laminar flow in a channel expansion. *J. Comput. Phys.* **54** (3), 468–488.

Rabault, J., Kuchta, M., Jensen, A., Réglade, U. & Cerardi, N. 2019 Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *J. Fluid Mech.* **865**, 281–302.

Rabault, J. & Kuhnle, A. 2019 Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach. *Phys. Fluids* **31** (9), 094105.

Radespiel, R., Burnazzi, M., Casper, M. & Scholz, P. 2016 Active flow control for high lift with steady blowing. *Aeronaut. J.* **120** (1223), 171–200.

Raju, R., Mittal, R. & Cattafesta, L. 2008 Dynamics of airfoil separation control using zero-net mass-flux forcing. *AIAA J.* **46** (12), 3103–3115.

Ren, F., Rabault, J. & Tang, H. 2021 Applying deep reinforcement learning to active flow control in weakly turbulent conditions. *Phys. Fluids* **33** (3), 037121.

Rodriguez, I., Lehmkuhl, O. & Borrell, R. 2020 Effects of the actuation on the boundary layer of an airfoil at Reynolds number $Re = 60\,000$. *Flow Turbul. Combust.* **105**, 607–626.

Schulman, J., Moritz, P., Levine, S., Jordan, M. & Abbeel, P. 2015 High-dimensional continuous control using generalized advantage estimation. arXiv:1506.02438.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. 2017 Proximal policy optimization algorithms. arXiv:1707.06347.

Seifert, A., Darabi, A. & Wyganski, I. 1996 Delay of airfoil stall by periodic excitation. *J. Aircraft* **33** (4), 691–698.

Sonoda, T., Liu, Z., Itoh, T. & Hasegawa, Y. 2023 Reinforcement learning of control strategies for reducing skin friction drag in a fully developed turbulent channel flow. *J. Fluid Mech.* **960**, A30.

Sun, Q., Bahri, F., Jabbal, M., Stryczniewicz, W., Jefferson-Loveday, R., Stefes, B. & Büscher, A. 2025 Suppression of flow separation of a high-lift wing with active flow control. *Aerosp. Sci. Technol.* **159**, 110017.

Suárez, P., Alcántara-Ávila, F., Miró, A., Rabault, J., Font, B., Lehmkuhl, O., Vinuesa, R. 2025*a* Active flow control for drag reduction through multi-agent reinforcement learning on a turbulent cylinder at $Re_D = 3900$. *Flow Turbul. Combust.* **115**, 3–27.

Suárez, P., Alcántara-Ávila, F., Rabault, J., Miró, A., Font, B., Lehmkuhl, O. & Vinuesa, R. 2025*b* Flow control of three-dimensional cylinders transitioning to turbulence via multi-agent reinforcement learning. *Commun. Engng* **4**, 113.

Tang, H., Rabault, J., Kuhnle, A., Wang, Y. & Wang, T. 2020 Robust active flow control over a range of Reynolds numbers using an artificial neural network trained through deep reinforcement learning. *Phys. Fluids* **32** (5), 053605.

Tonti, F., Rabault, J. & Vinuesa, R. 2025 Navigation in a simplified urban flow through deep reinforcement learning. *J. Comput. Phys.* **538**, 114194.

Varela, P., Suárez, P., Alcántara-Ávila, F., Miró, A., Rabault, J., Font, B., García-Cuevas, L.M., Lehmkuhl, O. & Vinuesa, R. 2022 Deep reinforcement learning for flow control exploits different physics for increasing Reynolds number regimes. *Actuators* **11** (12), 359.

Vasanth, J., Rabault, J., Alcántara-Ávila, F., Morensen, M. & Vinuesa, R. 2024 Multi-agent reinforcement learning for the control of three-dimensional Rayleigh–Bénard convection. *Flow Turbul. Combust.* **115**, 1319–1355.

Verma, S., Novati, G. & Koumoutsakos, P. 2018 Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proc. Natl Acad. Sci.* **115** (23), 5849–5854.

Vignon, C., Rabault, J., Vasanth, J., Alcántara-Ávila, F., Mortensen, M. & Vinuesa, R. 2023 Effective control of two-dimensional Rayleigh–Bénard convection: invariant multi-agent reinforcement learning is all you need. *Phys. Fluids* **35** (6), 065146.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press

VIQUERAT, J., RABAULT, J., KUHNLE, A., GHRAIEB, H., LARCHER, A. & HACHEM, E. 2021 Direct shape optimization through deep reinforcement learning. *J. Comput. Phys.* **428**, 110080.

WANG, Y., MEI, Y., AUBRY, N., CHEN, Z., WU, P. & WU, W. 2022 Deep reinforcement learning based synthetic jet control on disturbed flow over airfoil. *Phys. Fluids* **34** (3), 033606.

WANG, Q., YAN, L., HU, G., CHEN, W., RABAULT, J. & NOACK, B.R. 2024 Dynamic feature-based deep reinforcement learning for flow control of circular cylinder with sparse surface pressure sensing. *J. Fluid Mech.* **988**, A4.

WEISSENBACHER, M., BOROVYKH, A. & RIGAS, G. 2025 Reinforcement learning of chaotic systems control in partially observable environments. *Flow Turbul. Combust.* **115**, 1357–1378.

WILLIAMSON, C.H.K. 1988 The existence of two stages in the transition to three-dimensionality of a cylinder wake. *Phys. Fluids* **31** (11), 3165–3168.

WU, J.-Z., LU, X.-Y., DENNY, A.G., FAN, M. & WU, J.-M. 1998 Post-stall flow control on an airfoil by local unsteady forcing. *J. Fluid Mech.* **371**, 21–58.

XIONG, T., LIU, Z., WANG, Y., ONG, C.J. & ZHU, L. 2025 Chemotactic navigation in robotic swimmers via reset-free hierarchical reinforcement learning. *Nat. Commun.* **16**, 5441.

YAN, L., LI, Y., HU, G., CHEN, W., ZHONG, W. & NOACK, B.R. 2023 Stabilizing the square cylinder wake using deep reinforcement learning for different jet locations. *Phys. Fluids* **35** (11), 115104.

YOU, D. & MOIN, P. 2008 Active control of flow separation over an airfoil using synthetic jets. *J. Fluid. Struct.* **24** (8), 1349–1357. Unsteady Separated Flows and their Control.

ZHANG, W. & SAMTANEY, R. 2015 A direct numerical simulation investigation of the synthetic jet frequency effects on separation control of low-re flow past an airfoil. *Phys. Fluids* **27** (5), 055101.

ZHANG, W., ZHANG, Z., CHEN, Z. & TANG, Q. 2017 Main characteristics of suction control of flow separation of an airfoil at low Reynolds numbers. *Eur. J. Mech. – B/Fluids* **65**, 88–97.

ZHI-YONG, Z., WANG-LONG, Z., ZHI-HUA, C., XIAO-HUI, S. & CHEN-CHAO, X. 2018 Suction control of flow separation of a low-aspect-ratio wing at a low Reynolds number. *Fluid Dyn. Res.* **50** (6), 065504.

ZHOU, D. & BAE, H.J. 2024 Wall modeling of turbulent flows with varying pressure gradients using multi-agent reinforcement learning. *AIAA J.* **62** (10), 3713–3727.

https://doi.org/10.1017/jfm.2026.11837 Published online by Cambridge University Press