The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

DATA-DRIVEN ACCELERATION OF STATISTICAL CONVERGENCE IN TURBULENT FLOWS

N. MASCLANS[1] AND L. JOFRE[1]

[1] Department of Fluid Mechanics
Universitat Politècnica de Catalunya · BarcelonaTech (UPC), Barcelona 08019, Spain
e-mails: nuria.masclans@upc.edu, lluis.jofre@upc.edu

Key words: Reinforcement learning, Statistics convergence, Turbulent flows

Summary.

Direct numerical simulations are essential for understanding and modeling turbulent flow phenomena. However, achieving converged statistics of high-Reynolds-number turbulent flows typically requires high-resolution DNS experiments over extended time periods, resulting in significant computational costs. This work, therefore, introduces an innovative data-driven methodology tailored to accelerate statistical convergence of turbulent flows, potentially transforming the computational study and optimization of turbulent flows by largely reducing compute times. The strategy involves hastening the convergence of turbulent statistics through controlled "on the fly" perturbations to the Reynolds stresses within a physics-constrained framework. Reinforcement learning is employed to determine the perturbations across the six degrees of freedom of the tensor, encompassing its magnitude, shape and orientation. To that end, this novel framework will be comprehensively described and its performance carefully assessed. The tests will consider the canonical one-dimensional turbulence channel flow at $Re_\tau = 180$, analyzing convergence behavior and speedup factors of first-order turbulent flow statistics.

1 INTRODUCTION

The study, design and optimization of turbulent flow systems by means of direct numerical simulation (DNS) approaches faces significant challenges due to the long computational times typically required to converge statistics. The significant computational demands of such high-fidelity simulations arise from resolving the full range of spatial and temporal scales present in turbulence, which require high-resolution grids and fine temporal integration to fully capture the inherent multiscale non-linear flow dynamics. Therefore, although DNS results provide detailed insight for computational fluid dynamics (CFD) analysis, it is often impractical for design purposes, leading to the use of less accurate methods, such as Reynolds-averaged Navier-Stokes and large-eddy simulation (LES). Moreover, the carbon footprint associated with DNS computations is a growing concern as it scales as Re^4 for wall-bounded flows [1]. This environmental cost is compounded by the extended simulation times required to account for the full spectrum of turbulent scales and achieve statistical convergence. For instance, the time integration requirements for reporting statistical errors of incompressible turbulent flows is on the order of $T/T_{\text{eddy}} \sim 10^4$ with T_{eddy} the characteristic time of the largest eddies [2]. This is significantly longer than what is typically feasible even with substantial computational resources, making the attainment of converged statistical data a considerable computational, temporal and environmental burden [3].

These challenges underscore the need for innovative approaches to reduce the costs and impacts of high-fidelity CFD simulations. In this regard, deep learning (DL) has made significant advances in accelerating numerical simulations through diverse strategies, including solver acceleration for the Poisson equation [4], domain reduction [5], improved accuracy of finite-difference and finite-volume discretization schemes [6], time parallelization [7], fitting closures to classical turbulence models [8], pure machine learning (ML) to replace the Navier-Stokes equations with deep neural network (DNN) approximations [9], and hybrid ML coupled with traditional numerical methods to replace or accelerate iterative solvers [10] and correct errors in under-resolved simulations [11]. However, these methods still require long simulation times for statistical convergence and, to some degree, need to be adapted to each problem under study. Thus, to the best of the authors' knowledge, no strategy has effectively reduced the actual simulation time required to achieve converged statistics.

Simultaneously, the integration of DNNs with reinforcement learning (RL) techniques - referred to as deep reinforcement learning (DRL) - has significantly enhanced decision-making capabilities. By leveraging the advanced feature extraction and high-dimensional data processing capabilities of DNN, DRL has garnered significant interest within the fluid mechanics community. DRL excels at capturing long-term dynamics and learning policies for decision-making, which is essential for flow control and optimization tasks. Despite its great potential, DRL applications in computational and experimental fluid dynamics remains relatively under-explored. Since the pioneering work by Novati and Verma et al. [12, 13], there has been a notable increase in DRL contributions in turbulence modeling, flow control and optimization problems. Among these, drag reduction applications have received the most attention, with numerous studies dedicated to this field [14]. Other significant applications include heat transfer [15], shape optimization [16], and microfluidics [17]. Among the various algorithms available, proximal policy optimization (PPO) [18] is widely recognized for its superior learning stability and resilience to hyperparameter changes, making it a widely used method in CFD [19].

This work, thus, proposes a DRL-based approach to reduce simulation times for achieving converged statistics in turbulent flows. A PPO algorithm is employed to train an agent that introduces physically realizable perturbations within the Reynolds stresses of the flow. The aim is to slightly perturb the turbulent flow to undergo a different but quicker instantaneous trajectory, so that the same converged statistics are reached in less simulation time than the unperturbed simulation. The paper is organized as follows. First, Section 2 details the DRL framework, including the one dimensional turbulence (ODT) simulation environment, the perturbation-based DRL framework, the PPO algorithm employed, and the numerical and training details. Next, Section 3 evaluates the effectiveness of the proposed framework in accelerating statistical convergence in an ODT channel flow at $Re_\tau = 180$. Finally, Section 4 presents the conclusions and further work.

2 METHODOLOGY

This section outlines the approach to accelerate statistics convergence in a computational fluid dynamics environment. The different subsections cover the ODT environment, the formulation of the statistics convergence optimization problem, an introduction to DRL, and the PPO algorithm. Additionally, computation and training details are provided.

2.1 Computational One-Dimensional Turbulence environment

The ODT [20] is a computationally efficient method for simulating turbulent flows, specially boundary-layer problems with a dominant shear direction [21, 22]. It reduces the complexity of the three-dimensional turbulence to a one-dimensional framework, enabling faster simulations while retaining essential turbulent characteristics. The choice of ODT as the simulation environment sets the stage for addressing the statistics convergence problem in a computationally affordable manner.

Figure 1 schematically illustrates the channel flow setup for DNS and the ODT model. Unlike traditional Navier-Stokes-based approaches, which capture large to small flow scales, ODT resolves fine scales and models large-scale advection through two concurrent processes:

- *1-D unsteady diffusion/reaction flow equations*: excludes the advective term, reducing the Navier-Stokes equations to diffusion equations with source terms.
- *Stochastic advection*: introduces turbulent advection via stochastic eddy events using a triple map (Figure 2). This feedback mechanism emulates the effect of a 3-D eddy on 1-D property profiles by capturing compressive strain and rotational folding effects, and conserves all quantities without causing properties discontinuities.

The two-way coupling between the diffusion mechanism and the advection phenomenological modeling by eddy events in ODT temporal flow is represented as

$$\frac{\partial \rho u_i}{\partial t} + E_i(y) = \frac{\partial}{\partial y}\left(\mu \frac{\partial u_i}{\partial y}\right) + S_{u_i}, \tag{1}$$

$$\frac{\partial \rho \phi}{\partial t} + E_\phi(y) = \frac{\partial}{\partial y}\left(\kappa_\phi \frac{\partial \phi}{\partial y}\right) + S_\phi, \tag{2}$$

where y represents the wall-normal coordinate of the simulated ODT 1-D line, E is the eddy function, ϕ is an arbitrary scalar with diffusion coefficient κ_ϕ and source term S_ϕ, and S_{u_i} is the source term of the velocity component u_i. For the incompressible Navier-Stokes equations with a pressure gradient source term

$$\frac{\partial u_i}{\partial t} + E_i(y) = \nu \frac{\partial^2 u_i}{\partial y^2} - \frac{1}{\rho}\frac{\partial p}{\partial x_i}, \tag{3}$$

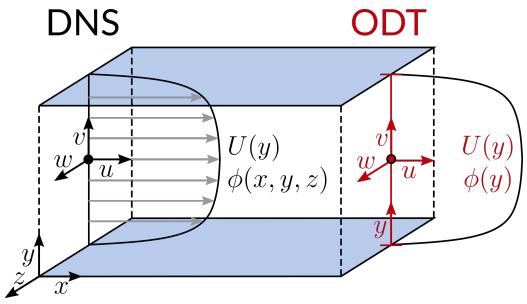

Figure 1: Schematic formulation of DNS (left) and ODT (right) channel flow simulation.

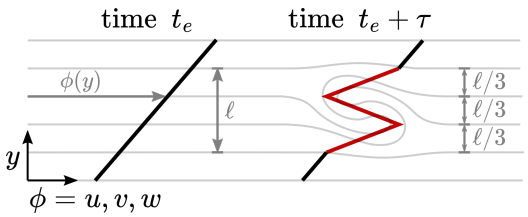

Figure 2: Schematic implementation of a triple map on an ODT profile segment.

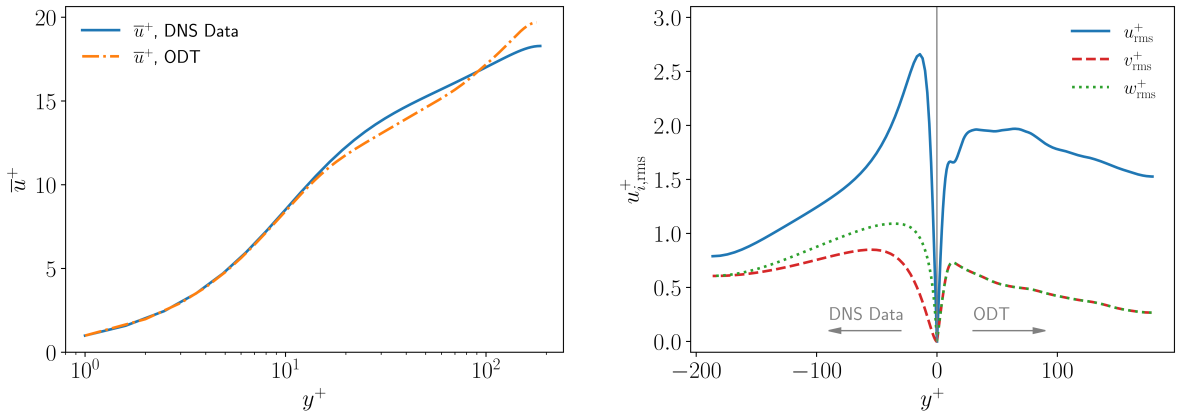

Figure 3: Comparison of mean streamwise velocity (left) and fluctuating velocity (right) profiles between DNS and ODT for channel flow at $Re_\tau = 180$.

where $P_x = \partial p/\partial x$ is the applied pressure gradient in the streamwise direction.

The performance of ODT is compared against DNS results for a channel flow at $Re_\tau = 180$. Figure 3 shows a reasonable ODT reconstruction of the mean streamwise velocity profile, while the fluctuating velocity profiles are less accurate. Consequently, this work focuses on accelerating the convergence of first-order statistics, specifically the mean streamwise velocity $\overline{u}$.

2.2 Formulation of the statistics convergence problem

Building on the ODT model, the optimization problem for accelerating statistics convergence is formulated with the aim of reducing the extensive computational resources and time traditionally required. This framework differs from typical optimization problems like those aimed at drag reduction, as it involves complex iterative processes to enhance temporal convergence without compromising reliability and accuracy.

The general derivation of the statistics convergence optimization problem for the three-dimensional incompressible Reynolds-Averaged Navier-Stokes (RANS) equations is presented

$$\frac{\partial \overline{u_i}}{\partial x_i} = 0, \tag{4}$$

$$\frac{\partial \overline{u_i}}{\partial t} + \overline{u_j}\frac{\partial \overline{u_i}}{\partial x_j} = \nu \frac{\partial^2 \overline{u_i}}{\partial x_j^2} - \frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} - \frac{\partial R_{ij}}{\partial x_j}, \tag{5}$$

where $u_i = \overline{u_i} + u_i'$ denotes the Reynolds decomposition, $R_{ij} = \overline{u_i' u_j'}$ is the Reynolds stress tensor, and $k = \frac{1}{2}R_{ii}$ is the turbulent kinetic energy. It is well established that the Reynolds stress tensor, R_{ij}, must satisfy the *realizability conditions* to remain symmetric and positive semi-definite, ensuring k is non-negative and real [23]. Satisfying these conditions allows the eigen-decomposition of the symmetric and trace-free normalized anisotropy tensor, A_{ij}, as

$$A_{ij} = \left(\frac{R_{ij}}{R_{kk}} - \frac{1}{3}\delta_{ij}\right) = Q_{in}\Lambda_{nl}Q_{jl}, \tag{6}$$

where R_{kk} is the trace of R_{ij}, Q_{in} is the orthonormal matrix of eigenvectors, and Λ_{nl} is the diagonal matrix of eigenvalues, ordered such that $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Thus, the Reynolds stress tensor R_{ij} can be represented in terms of its magnitude, R_{kk}, shape, Λ_{nl}, and orientation, Q_{in}, as

$$R_{ij} = R_{kk}\left(Q_{in}\Lambda_{nl}Q_{jl} + \frac{1}{3}\delta_{ij}\right). \tag{7}$$

Consequently, R_{ij} is expressed in terms of six degrees of freedom (d.o.f.) $z = (R_{kk}, \theta, \psi, \phi, x_1, x_2)$: one for the tensor magnitude (R_{kk}), three for the shape defined by the eigenvectors Euler angles (θ, ψ, ϕ), and two for the orientation specified by the eigenvalues barycentric coordinates (x_1, x_2) [24]. This parametrization aids in understanding the contributions of different components of R_{ij} to the statistical properties of the flow.

To accelerate statistics convergence, perturbations are introduced to the d.o.f. of the non-converged and realizable R_{ij} as $\Delta z_{pert} = (\Delta R_{kk}, \Delta\theta, \Delta\psi, \Delta\phi, \Delta x_1, \Delta x_2)$, leading to a non-converged, perturbed and non-realizable tensor $\hat{R}_{ij} = R_{ij}(z + \Delta z_{pert}) = R_{ij} + \Delta R_{ij}$. This tensor is then modified to satisfy the realizability conditions, yielding $\widetilde{R}_{ij} = R_{ij}(z + \Delta z_{pert} + \Delta z_{rc}) = R_{ij} + \Delta\widetilde{R}_{ij}$, where Δz_{rc} is the minimum perturbation to ensure realizability. The perturbed and realizable $\widetilde{R}_{ij}$ is introduced into the RANS momentum equation (5) as

$$0 = -\overline{u_j}\frac{\partial\,\overline{u_i}}{\partial x_j} + \nu\frac{\partial^2\,\overline{u_i}}{\partial x_j^2} - \frac{1}{\rho}\frac{\partial\,\overline{p}}{\partial x_i} - \frac{\partial\,R_{ij}}{\partial x_j} + F_{i,pert}, \quad \text{with } F_{i,pert} = -\frac{\partial\,\Delta\widetilde{R}_{ij}}{\partial x_j}, \tag{8}$$

using $\frac{\partial\,\widetilde{R}_{ij}}{\partial x_j} = \frac{\partial\,R_{ij}}{\partial x_j} + \frac{\partial\,\Delta\widetilde{R}_{ij}}{\partial x_j}$. Therefore, the goal is to determine the perturbation load $F_{pert}(\Delta R_{ij})$ along time that accelerates the convergence of statistical measurements relative to the non-perturbed scenario.

For the ODT model, the control problem is derived from the corresponding equation (3)

$$0 = E_i(y) + \nu\frac{\partial^2\,\overline{u_i}}{\partial y^2} - \frac{1}{\rho}\frac{\partial\,\overline{p}}{\partial x_i} - \frac{\partial\,R_{iy}}{\partial y} + F_{i,pert}, \quad \text{with } F_{i,pert} = -\frac{\partial\,\Delta\widetilde{R}_{iy}}{\partial y}. \tag{9}$$

Due to the one-dimensional nature of ODT formulation, characterized by $\partial(\cdot)/\partial x_j = 0$ for $x_j \neq y$, only three out of the nine perturbation terms are introduced, $\Delta\widetilde{R}_{iy}$, leading to differences in the perturbed $\widetilde{R}_{ij}$ compared to the three-dimensional framework.

This work does not impose additional constraints on ΔR_{iy} for statistics convergence. Instead, a DRL framework introduces the non-converged Reynolds stress tensor perturbation ΔR_{ij} through a single perturbing agent, adding an instantaneous load $F_{pert}(\Delta R_{ij})$ within an actuation domain. The following subsections 2.3 and 2.4 detail the methodology for learning optimal actions to achieve statistics convergence acceleration.

2.3 Deep Reinforcement Learning

To address the control problem, DRL is employed to determine the optimal perturbation load. DRL combines reinforcement learning with deep learning, enabling agents to learn optimal policies through interactions with the environment. Its capability to handle high-dimensional state spaces and learn complex behaviour makes DRL an ideal candidate for optimizing statistics convergence in turbulent flow simulations.

The DRL framework describes an agent that iteratively interacts with a dynamic environment to achieve a target, namely accelerating statistics convergence. The interaction cycle proceeds as follows: the environment provides a state s_t, the agent observes the state and selects an action a_t, and the environment introduces that action and transitions to the next state s_{t+1}, returning this state and a reward r_t to the agent. This cycle, known as an experience (s_t, a_t, r_t), occurs over one time-step t, and is repeated until termination, either upon reaching a maximum time-step $t = T$ or achieving a predefined convergence criterion for the statistics. The sequence from $t = 0$ to termination constitutes an episode, during which the collection of experiences forms a trajectory $\tau = (s_0, a_0, r_0), (s_1, a_1, r_1), \ldots$. This process is illustrated as a control loop diagram in Figure 4, particularly within the context of ODT statistics acceleration framework.

DNNs are widely used to approximate primary functions governing agent actions, including the policy $\pi(a_t, s_t)$ for mapping states to actions, the value $V^\pi(s)$ or $Q^\pi(s, a)$ for estimating expected return, and the environment model $P(s'|s, a)$ for predicting state transition probabilities. Accordingly, DRL algorithms fall into three principal categories: policy-based, value-based, and model-based methods, each focusing on learning the respective functions. Additionally, DRL algorithms are also classified into on-policy and off-policy. On-policy algorithms use data generated by the current policy π_t for training and discard the training data after each policy update, while off-policy algorithms retain and reuse data generated by previous policies, enhancing sample efficiency at the cost of increased memory requirements.

Formally, the return $R(\tau)$ is the discounted sum of rewards in a trajectory

$$R(\tau) = \sum_{t=0}^{T} \gamma^t r_t, \tag{10}$$

where $\gamma \in [0, 1]$ is the discount factor, and the objective $J(\tau)$ is the expected return over many trajectories

$$J(\tau) = \mathbb{E}_{\tau\sim\pi}\big[\sum_{t=0}^{T} \gamma^t r_t\big]. \tag{11}$$

The agent aims to maximize this objective by selecting optimal actions, learning through interaction with the environment over several epochs. Policy Gradient (PG) method aims at solving the optimization problem

$$\max_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\big[R(\tau)\big], \tag{12}$$

where π_θ is the policy network. Policy parameters θ are updated by gradient ascent as

$$\theta \leftarrow \theta + \alpha\nabla_\theta J(\pi_\theta), \tag{13}$$

with α the learning rate, and $g = \nabla_\theta J(\pi_\theta)$ the policy gradient. PG method estimates g as [18]

$$\hat{g} = \hat{\mathbb{E}}_t\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t\right], \tag{14}$$

where $\hat{\mathbb{E}}_t$ denotes the average over a finite batch of samples, and $\hat{A}_t = \hat{A}^\pi(s_t, a_t)$ is the advantage function estimator. Finally, gradient descent methods compute estimator $\hat{g}$ by differentiating an objective or loss function

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t\big[\log \pi_\theta(a_t|s_t)\hat{A}_t\big]. \tag{15}$$

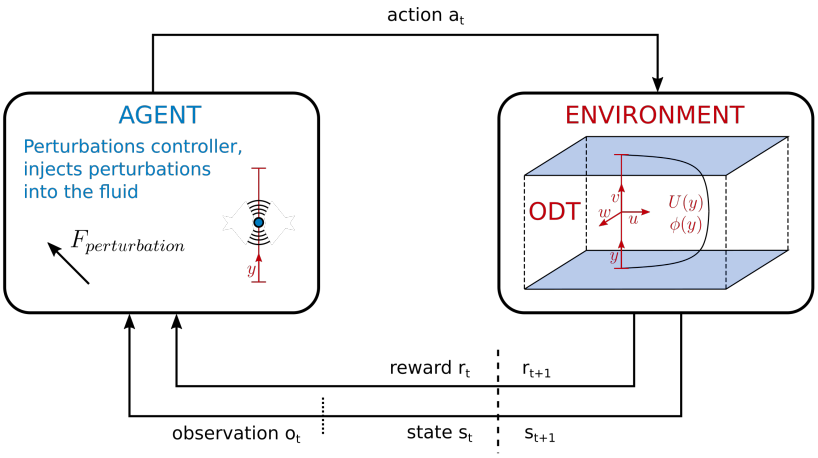

Figure 4: Reinforcement learning control loop for ODT statistics acceleration framework

2.4 Proximal Policy Optimization

PG algorithms often suffer from performance collapse, where the performance of an agent deteriorates rapidly due to the compounding effects of training on suboptimal trajectories. Additionally, on-policy algorithms are typically sample inefficient as they do not reuse data from previous experiences. To address these issues, PPO [18] was developed, building upon earlier PG algorithms like REINFORCE [25], Actor-Critic [26] and Trust Region Policy Optimization (TRPO) [27]. PPO is a state-of-the-art on-policy algorithm that optimizes both policy and value network parameters with respect to the expected return. Unlike traditional PG approaches that update policies based on individual samples, PPO employs a surrogate objective function to facilitate robust policy updates through minibatch training. This approach helps prevent performance collapse by ensuring monotonic policy improvement, and allows for the reuse of off-policy data during training, thereby enhancing sample efficiency. Consequently, PPO is particularly well-suited for environments where sample collection is costly or limited, such as turbulent flow simulations.

In detail, PPO builds upon the concept of a *surrogate objective* function [28]

$$L^{\mathrm{CPI}}(\theta) = \hat{E}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)}\hat{A}_t^{\pi_{\theta_{\mathrm{old}}}}\right] = \hat{E}_t\Big[r_t(\theta)\hat{A}_t\Big], \tag{16}$$

where CPI stands for conservative policy iteration, and $r_t(\theta)$ is the probability ratio. To maintain stability, PPO introduces a trust-region constrain to the optimization problem. One variant, the *KL-penalized surrogate objective*, incorporates a KL divergence penalty:

$$J^{\mathrm{KLPEN}}(\theta) = \max_\theta \hat{E}_t\Big[r_t(\theta)\hat{A}_t - \beta\mathrm{KL}\big(\pi_\theta(a_t|s_t)\,||\,\pi_{\theta_{\mathrm{old}}}(a_t|s_t)\big)\Big], \tag{17}$$

where β is the penalty coefficient. The *clipped surrogate objective* employs a clipping mechanism

$$J^{\mathrm{CLIP}}(\theta) = \hat{E}_t\Big[\min\Big(r_t(\theta)\hat{A}_t,\ \mathrm{clip}\big(r_t(\theta), 1-\epsilon, 1+\epsilon\big)\hat{A}_t\Big)\Big], \tag{18}$$

where ϵ defines the clipping neighborhood. The clipping method is favored for its simplicity, computational efficiency, and robust performance, making it the choice for this implementation.

2.5 Numerical Implementation and Training Details

The environment consists of an ODT simulation of incompressible channel flow at $Re_\tau = u_\tau \delta/\nu = 180$ with parameters $u_\tau = 1\,\mathrm{m/s}$, $\delta = 1\,\mathrm{m}$, $\nu = 0.0056\,\mathrm{m}^2/\mathrm{s}$. The implementation employs an open-source C++ code with adaptive mesh refinement [29]; detailed code information is available in Stephens & Lignell [30]. The ODT parameters used are documented in Lignell et al. [29], with an additional proportional feedback control loop to enforce $Re_\tau = 180$. The characteristic time of the problem based on viscous units is $t_\tau = \delta/u_\tau$. Statistics computation is initiated once turbulence is fully developed, taking around $t/t_\tau \approx 100$ [31]. Therefore, data collection for statistics calculation starts at this time point, with flow fields used as restart data for each training epoch. Performance is assessed against converged statistics at $t/t_\tau = 10000$.

The simulation's dominant scale is the viscous time scale $t_\tau = \nu/u_\tau^2 = 0.005566\,\mathrm{s}$, noted as δt_{phy}, with agent interactions at $\delta t_{act} = 0.01$. Observations are gathered from $n_{probes} = 360$ sensors at intervals of $\Delta y^+ = 1$ in the wall-normal direction. The observation space consists of (non-converged) R_{ij} d.o.f. data with dimensions $[5, 360]$, omitting the null and constant x_2 due to the one-dimensional nature of the simulation framework. Additionally, the (non-converged) mean streamwise velocity at such uniform grid, u_t, is captured for the reward calculation as

$$r_t = e^{-\frac{\mathrm{NRMSE}(\overline{u}_t)}{2\sigma^2}}, \tag{19}$$

where $\mathrm{NRMSE}(\overline{u}_t) = ||\overline{u}_t - \overline{u}_C||_2 \,/\, ||\overline{u}_C||_2$, and $\sigma = 0.1$. The action space is five-dimensional ($n_{act} = 5$), perturbing R_{ij} as $a_t = (\Delta R_{kk}, \Delta\theta, \Delta\psi, \Delta\phi, \Delta x_1)$, excluding Δx_2. Perturbations are applied across a grid of 260 points centered within the channel at intervals of $\delta y^+ = 1$, and the actions are bounded and filtered to ensure a smooth transition between the non-perturbed (viscous and buffer) and perturbed (logarithmic and outer) layers. Additionally, a regularization factor r_i is included in the ODT momentum equation to prevent numerical instability, by penalizing large perturbation loads suggested by sub-optimal policies as

$$\frac{\partial u_i}{\partial t} + E_i(y) = \nu \frac{\partial^2 u_i}{\partial y^2} - \frac{1}{\rho}\frac{\partial p}{\partial x_i} + r_i\, F_{i,pert}, \tag{20}$$

$$r_i = \min\left(\frac{|\mathrm{rhs}_i|}{|\mathrm{rhs}_i - F_{i,pert} + \epsilon|}, \frac{|\mathrm{rhs}_i|}{|\mathrm{rhs}_i + F_{i,pert} + \epsilon|}\right), \tag{21}$$

$$\mathrm{rhs}_i = \nu \frac{\partial^2 u_i}{\partial y^2} - \frac{1}{\rho}\frac{\partial p}{\partial x_i}. \tag{22}$$

Effective training of the PPO algorithm relies on adequate network architectures and carefully tuned hyperparameters, which must balance exploration and exploitation for stable training and optimal policy convergence. Based on an extensive empirical analysis, key algorithm parameters are summarized in Table 1.

3 PERFORMANCE ANALYSIS

This section evaluates the performance of the DRL framework within the context of stochastic ODT simulations of channel flow. Each training episode, or realization, is initialized from the same developed turbulence state, with varied random seeds for eddy generator and PPO network initialization to ensure robust training outcomes. Performance assessment involves ensemble

learning rate	$5e-4$
number of steps	64
batch size	64
number of epochs	≤ 50
discount factor γ	0.99
gral. adv. est. factor gae_λ	0.95
clipping range	0.2
entropy coefficient	$1e-3$
value coefficient	$1e-1$
initial log standard deviation	-1.9
eposide termination	$t^+ = 20$
training termination	$\text{NRMSE}(\overline{u}) = 7.5e-3$
features extractor network	1-D CNN [6, 64, 64] + flatten + MLP [256]
policy network	MLP [256, 64, 64, 64, 6]
value-function network	MLP [256, 64, 64, 1]

Table 1: PPO training parameters and networks architecture

averaging over 10 randomized training runs, consistently achieving the targeted normalized root mean squared error for the mean streamwise velocity profile, $\text{NRMSE}(\overline{u}) = 7.5e-3$, within 1 to 3 epochs. On average, the DRL approach reaches convergenced statistics in approximately $t_C^+ \approx 49$, compared to $t_C^+ \approx 80$ required by non-perturbed simulations. This represents an approximate 40% reduction in convergence time for the mean streamwise velocity, facilitated by the supervised R_{ij} perturbations-based DRL framework.

Figure 5 illustrates the temporal evolution of $\text{NRMSE}(\overline{u})$, comparing the non-accelerated ensemble-averaged reference (gray and black lines) to an individual DRL-based training process (blue and green lines). The selected training example achieves the targeted convergence tolerance within a total simulation time of $t^+ \approx 30$ over 2 realizations, while the non-perturbed simulation, averaged over the ensemble, requires $t^+ \approx 80$. Initially, the first RL epoch (blue line) experiences temporary convergence degradation compared to the reference ensemble result (black line), which causes episode termination at $t^+ = 20$. Consequently, the simulation is reinitialized into the second realization (green line), which achieves the targeted convergence tolerance at local $t^+ \approx 10$ thereby completing the training process. This behaviour suggests effective learning of an optimal policy by the DRL framework, leading to faster statistical convergence.

Overall, these results underscore the efficacy of the proposed method in accelerating turbulent statistics convergence in ODT channel flow simulations, demonstrating significant potential for enhancing computational efficiency and reducing convergence times for key statistical quantities.

4 CONCLUSIONS

This work presents a novel R_{ij} perturbation-based framework to accelerate the statistics convergence in turbulent flow simulations using deep reinforcement learning. Specifically, the application of a proximal policy optimization algorithm demonstrated accelerated statistics convergence for one-dimensional turbulence channel flow simulations at $Re_\tau = 180$. The framework's general formulation enables straightforward adaptation to diverse complex turbulent flow problems, computational solvers, and DRL algorithms. The novelty of the proposed approach lies

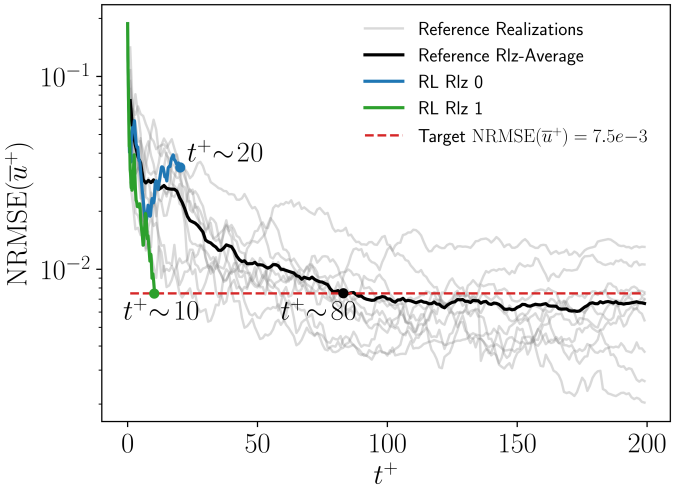

Figure 5: Normalized root mean squared error evolution of non-converged streamwise velocity profile, NRMSE($\overline{u}$), comparing non-perturbed reference and DRL-based perturbed simulations.

in addressing the direct reduction of statistics convergence simulation time through controlled perturbations, a methodology that has not been previously explored compared to the extensive research on enhancing numerical and computational performance. Therefore, this approach offers a promising strategy for addressing computational challenges and achieving high-fidelity turbulence data, potentially decreasing the time and resources required for such analysis.

Future research will focus on several key areas to build upon the initial success of this framework. A primary area of interest is deepening the understanding of the perturbation actions influence on the overall statistics convergence. This investigation will employ cluster-based interpretation methods to extract representative states from observations and explore the relationship between control inputs and corresponding actions. Secondly, the framework will be extended to three-dimensional turbulent channel flow simulations, providing a comprehensive evaluation of the framework's capabilities and identifying potential challenges in higher-dimensional flow simulations. Additionally, further exploration will involve training with low Re_τ simulation data to accelerate statistics convergence at high-Re_τ, extending applicability to a broader range of turbulent flow conditions. Lastly, ongoing work includes developing an unsupervised reward formulation independent of the converged statistics baseline. This advancement would enable the framework to operate effectively without prior information on converged statistics, making it suitable for very computationally expensive simulations.

ACKNOWLEDGMENTS

This work is funded by the European Union (ERC, SCRAMBLE, 101040379). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. The authors also acknowledge support from the SGR (2021-SGR-01045) program of the Generalitat de Catalunya (Spain).

REFERENCES

[1] Xiang I A Yang, Wen Zhang, Mahdi Abkar, and William Anderson. Computational fluid dynamics: its carbon footprint and role in carbon emission reduction, 2024.

[2] Y. Shirian, J. A. K. Horwitz, and A. Mani. On the convergence of statistics in simulations of stationary incompressible turbulent flows. *Comput. Fluids*, 266:106046, 2023.

[3] P. Schlatter and R. Örlü. Assessment of direct numerical simulation data of turbulent boundary layers. *J. Fluid Mech.*, 659:116–126, 2010.

[4] A. Chikitkin and F. Noskov. Accelerating explicit method for poisson's equation using machine learning techniques. *AIP Conference Proceedings*, 2312(1):020002, 2020.

[5] K. Fukami, K. Fukagata, and K. Taira. Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid Mech.*, 870:106–120, 2019.

[6] Y. Bar-Sinai, S. Hoyer, J. Hickey, and M. P. Brenner. Learning data-driven discretizations for partial differential equations. *Proc. Natl. Acad. Sci. U.S.A.*, 116(31):15344–15349, 2019.

[7] Q. Wang, S. A. Gomez, P. J. Blonigan, A. L. Gregory, and E. Y. Qian. Towards scalable parallel-in-time turbulent flow simulations. *Phys. Fluids*, 25(11):110818, 2013.

[8] A. Beck, D. Flad, and C. D. Munz. Deep neural networks for data-driven les closure models. *J. Comput. Phys.*, 398:108910, 2019.

[9] Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.*, 9:1–10, 2018.

[10] O. Obiols-Sales, A. Vishnu, N. Malaya, and A. Chandramowliswharan. CFDNet: A deep learning-based accelerator for fluid simulations. 2020.

[11] J. Sirignano, J. F. MacArt, and J. B. Freund. Dpm: A deep learning pde augmentation method with application to large-eddy simulation. *J. Comp. Phys.*, 423:109811, 2020.

[12] Synchronisation through learning for two self-propelled swimmers. *Bioinspir. Biomim.*, 12(3):036001, 2017.

[13] S. Verma, G. Novati, and P. Koumoutsakos. Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proc. Natl. Acad. Sci. U.S.A.*, 115(23):5849–5854, 2018.

[14] P. Garnier, J. Viquerat, J. Rabault, A. Larcher, A. Kuhnle, and E. Hachem. A review on deep reinforcement learning for fluid mechanics. *Computers Fluids*, 225:104973, 2021.

[15] E. Hachem, H. Ghraieb, J. Viquerat, A. Larcher, and P. Meliga. Deep reinforcement learning for the control of conjugate heat transfer. *J. Comput. Phys.*, 436:110317, 2021.

[16] S. Qin, S. Wang, L. Wang, C. Wang, G. Sun, and Y. Zhong. Multi-objective optimization of cascade blade profile based on reinforcement learning. *Appl. Sci.*, 11(1), 2021.

[17] X. Y. Lee, A. Balu, D. Stoecklein, B. Ganapathysubramanian, and S. Sarkar. A case study of deep reinforcement learning for engineering design: Application to microfluidic devices for flow sculpting. *J. Mech. Des.*, 141(11):111401, 2019.

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017. arXiv preprint arXiv:1707.06347.

[19] J. Viquerat, P. Meliga, A. Larcher, and E. Hachem. A review on deep reinforcement learning for fluid mechanics: An update. *Phys. Fluids*, 34(11), 2022.

[20] A. R. Kerstein. One-dimensional turbulence: model formulation and application to homogeneous turbulence, shear flows, and buoyant stratified flows. *J. Fluid Mech.*, 392:277–334, 1999.

[21] A. R. Kerstein, Wm. T. Ashurst, S. Wunsch, and V. Nilsen. One-dimensional turbulence: Vector formulation and application to free shear flows. *J. Fluid Mech.*, 447:85–109, 2001.

[22] Falko Meiselbach. *Application of ODT to turbulent flow problems.* PhD thesis, 06 2015.

[23] U. Schumann. Realizability of Reynolds-stress turbulence models. *Phys. Fluids*, 20(5):721–725, 1977.

[24] Lluís Jofre, Stefan P. Domino, and Gianluca Iaccarino. Eigensensitivity analysis of subgrid-scale stresses in large-eddy simulation of a turbulent axisymmetric jet. *International Journal of Heat and Fluid Flow*, 77:314–335, 2019.

[25] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[26] V. R. Konda and J. N. Tsitsiklis. On actor-critic algorithms. *SIAM J. Control Optim.*, 42(4):1143–1166, 2003.

[27] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1889–1897, 2015.

[28] S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning*, volume 2, pages 267–274, 2002.

[29] D. O. Lignell, A. R. Kerstein, G. Sun, and E. E. Monson. Mesh adaption for efficient multiscale implementation of one- dimensional turbulence. *Theor. Comput. Fluid Dyn.*, 27(3-4):273–295, 2013.

[30] V. B. Stephens and D. O. Lignell. One-dimensional turbulence (ODT): Computationally efficient modeling and simulation of turbulent flows. *SoftwareX*, 13:100641, 2021.

[31] F. Schwertfirm and M. Manhart. Dns of passive scalar transport in turbulent channel flow at high schmidt numbers. *Int. J. Heat Fl. Flow*, 28(6):1204–1214, 2007.