Automating Turbulence Modeling by Multi-Agent Reinforcement Learning

Guido Novati[a], **Hugues Lascombes de Laroussilhe**[a], **and Petros Koumoutsakos**[a,1]

[a]Computational Science and Engineering Laboratory, Clausiusstrasse 33, ETH Zürich, CH-8092, Switzerland

This manuscript was compiled on May 20, 2020

The modeling of turbulent flows is critical to scientific and engineering problems ranging from aircraft design to weather forecasting and climate prediction. Over the last sixty years numerous turbulence models have been proposed, largely based on physical insight and engineering intuition. Recent advances in machine learning and data science have incited new efforts to complement these approaches. To date, all such efforts have focused on supervised learning which, despite demonstrated promise, encounters difficulties in generalizing beyond the distributions of the training data. In this work we introduce multi-agent reinforcement learning (MARL) as an automated discovery tool of turbulence models. We demonstrate the potential of this approach on Large Eddy Simulations of homogeneous and isotropic turbulence using as reward the recovery of the statistical properties of Direct Numerical Simulations. Here, the closure model is formulated as a control policy enacted by cooperating agents, which detect critical spatio-temporal patterns in the flow field to estimate the unresolved sub-grid scale (SGS) physics. The present results are obtained with state-of-the-art algorithms based on experience replay and compare favorably with established dynamic SGS modeling approaches. Moreover, we show that the present turbulence models generalize across grid sizes and flow conditions as expressed by the Reynolds numbers.

Multi-agent Reinforcement Learning | Turbulence Modeling | Large-eddy Simulations

The prediction of turbulent flows is critical for engineering (cars to nuclear reactors), science (ocean dynamics to astrophysics) and policy (climate modeling and weather forecasting). Over the last sixty years we have increasingly relied for such predictions on simulations based on the numerical integration of the Navier-Stokes equations. Today we can perform simulations using trillions of computational elements and resolve flow phenomena at unprecedented detail. However, despite the ever increasing availability of computing resources, most simulations of turbulent flows require the adoption of models to account for the spatio-temporal scales that cannot be resolved. Over the last few decades, the development of Turbulence Models (TM) has been the subject of intense investigations that have relied on physical insight and engineering intuition. Recent advances in machine learning and in the availability of data have offered new perspectives (and hope) in developing data-driven TM. Interestingly, turbulence and statistical learning theories have common roots in the seminal works of Kolmogorov on the analysis of homogeneous and isotropic turbulent flows (see (1, 2). These flows are characterized by vortical structures and their interactions exhibiting a broad spectrum of spatio-temporal scales (3–5) At one end of the spectrum we encounter the integral scales, which depend on the specific forcing, flow geometry, or boundary conditions. At the other end we find the Kolmogorov scales at which turbulent kinetic energy is dissipated. The handling of these turbulent scales provides a classification of turbulence simulations: Direct Numerical Simulations (DNS), which use a sufficient number of computational elements to resolve all scales of the flow field, and simulations using TM where the equations are solved in relatively few computational elements and the non-resolved terms are described by closure models. In DNS (6) most of the computational effort is spent in fully resolving the Kolmogorov scales despite them being statistically homogeneous and largely unaffected by large scale effects. Remarkable DNS (7) have provided us with unique insight into the physics of turbulence that can lead in turn to effective TM. However, it is well understood that in the foreseeable future DNS will not be feasible at resolutions necessary for engineering applications. In TM (8) two techniques have been dominant: Reynolds Averaged Navier-Stokes (RANS) and Large-eddy Simulations (LES) in which only the large scale unsteady physics are explicitly computed whereas the sub grid-scale (SGS), unresolved, physics are modeled. In the context of LES (9), classic approaches to the explicit modeling of SGS stresses include the standard (10) and the dynamic Smagorinsky model (11–13). SGS models have been constructed using physical insight, numerical approximations and often problem-specific intuition. While efforts to develop models for turbulent flows using machine learning and neural networks (NN) in particular date back decades (14, 15), recent advances in hardware and algorithms have made their use feasible for the development of data-driven turbulence closure models (16).

To date, to the best of our knowledge, all data-driven tur-

Significance Statement

Turbulence Modeling (TM) is an essential component of flow simulations routinely used in fields ranging form car design to weather forecasting. Over the last sixty years the development of TM has been founded on physical insight and engineering intuition. Here, we introduce the automated discovery of TM by multi-agent Reinforcement Learning (RL). RL systematically explores spatiotemporal patterns in under-resolved simulations to produce TM that are shown to generalize to previously unseen flow conditions and resolutions. RL provides a potent framework to solve longstanding challenges in TM and can have a major impact in the predictive capabilities of flow simulations used across science and engineering.

G.N. and P.K. designed research; G.N. and H.L. performed research; G.N. and P.K. wrote the paper.

The authors declare no conflict of interest.

[1]To whom correspondence should be addressed. E-mail: petros@ethz.ch

bulence closure models are based on supervised learning (SL). In LES, early approaches (19) trained a NN to emulate and speed-up a conventional, but computationally expensive, SGS model. More recently, data-driven SGS models have been trained by SL to predict the "perfect" SGS terms computed from filtered DNS data (20, 21). Variants include deriving the target SGS term from optimal estimator theory (22) and reconstructing the SGS velocity field as a deconvolution operation, or inverse filtering (23–25). In SL the parameters of the NN are commonly derived via a gradient descent algorithm to minimize the model prediction error. As the error is required to be differentiable with respect to the model parameters, and due to the computational challenge of obtaining chain-derivatives through a flow solver, SL approaches often define one-step target values for the model (e.g. reference SGS stresses computed from filtered DNS). Therefore it is necessary to differentiate between *a priori* and *a posteriori* testing. The first measures the accuracy of the SL model in predicting the target values on a database of reference simulations, typically obtained via DNS. *A posteriori* testing is performed after training, by integrating in time the NSE along with trained SL closure and comparing the obtained statistical quantities to that of DNS or other references. Due to the single-step cost function, the resultant NN model is not trained to compensate for the systematic discrepancies between DNS and LES and the compounding errors. The issue of ill-conditioning of data-driven SGS models has been exposed by studies that perform *a posteriori* testing (26). For example, in the work by (20), while the SGS stresses are accurately recovered, the mean flow velocities are not. Moreover, (27) shows that in many cases the perfect SGS model is structurally unstable and diverges from the original trajectory under perturbation. Likewise, (28) shows that a deep NN trained by SL, while closely matching the perfect SGS model for any single step, accumulates high-spatial frequency errors which cause instability.

We introduce Reinforcement Learning (RL) as a framework for the automated discovery of closure models of non-linear conservation laws, here applied to the construction of SGS models for LES. The key distinction between RL and SL is that RL optimizes a parametric model by direct exploration-exploitation of the underlying task specification. Moreover, the performance of a RL strategy is not measured by a differentiable objective function but by a cumulative reward. These features are especially beneficial in the case of TM as they permit avoiding the distinction between *a priori* and *a posteriori* evaluation. RL training is not performed on a database of reference data, but is performed by integrating in time the model and its consequences. In the case of LES, the performance of the RL strategy may be measured by comparing the statistical properties of the simulation to those of reference data. Indeed, rather than perfectly recovering SGS computed from filtered simulations, which may produce numerically unstable LES (27), RL can develop novel models which are optimized to accurately reproduce the quantities of interest.

Multi-agent RL for sub-grid scale modeling

RL identifies optimal strategies for agents that perform *actions*, contingent on their *state*, which affect the environment, and measure their performance via scalar *reward* functions. Todate, RL has been used in fluid mechanics solely in applications of control (29–32). In these examples the control action is defined by an embodied agent capable of spontaneous motion. By interacting with the flow field, agents trained through RL were able to gather relevant information and optimize their decision process to perform collective swimming (29), soar (31), minimize their drag (30, 33), or reach a target location (32, 34). Here we cast the TM problem as an optimization (35) and introduce RL to control an under-resolved simulation (LES) with the objective of reproducing quantities of interest computed by fully resolved DNS. The methodology by which RL is incorporated as part of the flow solver has a considerable effect on the computational efficiency of the resulting algorithm. As an example, following the common practice in video games (36), the state of the agent could be defined as the full three-dimensional flow field at a given time-step and the action as some quantity used to compute the SGS terms for

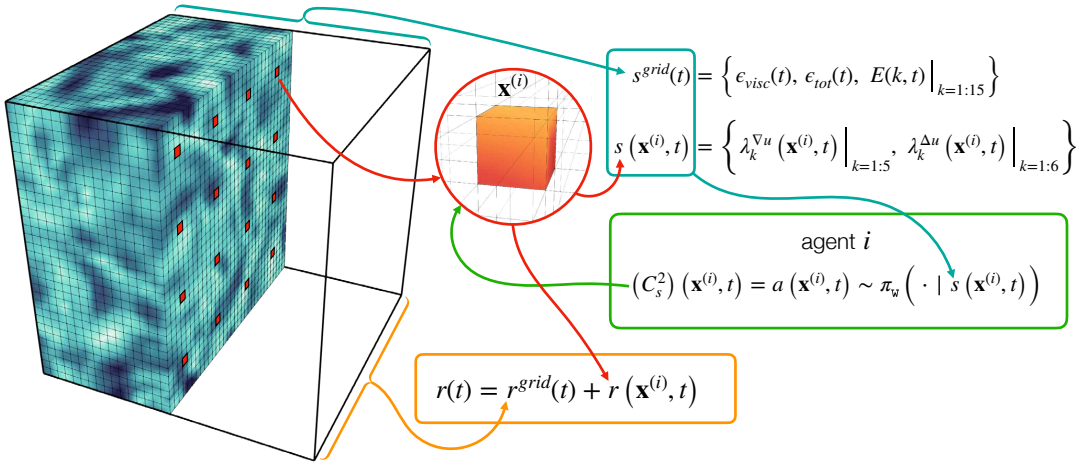


Fig. 1. Schematic representation of the integration of RL with the flow solver. The dispersed agents compute the SGS dissipation coefficient $\left(C_s^2\right)$ for each grid-point of the simulation. In order to embed tensorial invariance into the NN inputs (17), the local components of the state vector are the 5 invariants (18) of the gradient ($\lambda_k^{\nabla \mathbf{u}}$) and the 6 invariants of the Hessian of the velocity field ($\lambda_k^{\Delta \mathbf{u}}$) computed at the agents' location. The global components of the state are the energy spectrum up to the Nyquist frequency, the ratio of viscous dissipation (ϵ_{visc}) and total dissipation (ϵ_{tot}) relative to the energy injection rate ϵ. For a LES grid-size $N = 32^3$, we have state dimensionality $dim_S = 28$, far fewer variables than the full state of the system (i.e. the entire velocity field and $dim_S = 3 \cdot 32^3$).

all grid-points. However, such architecture would have the following challenges: it would be mesh-size dependent, it would involve a very large underlying NN, and the memory needed to store the experiences of the agent would be prohibitively large. We overcome these issues by deploying N_{agents} dispersed RL agents (marked as red cubes in Fig. 1) with localized actuation that use a combination of local and global information on the flow field, encoded in the state $s(\boldsymbol{x},t) \in \mathbb{R}^{dim_S}$ (here, $\boldsymbol{x}$ is the spatial coordinate and t the time step). A policy-network with parameters $\mathtt{w}$ is trained by RL to select the local SGS dissipation coefficient $C_s^2(\boldsymbol{x},t) \sim \pi_{\mathtt{w}}(\cdot|\mathbf{s})$. The multi-agent RL (MARL) advance in turns by updating $C_s^2(\boldsymbol{x},t)$ for the entire flow and integrating in time for Δt_{RL} to the next RL step until T_{end} or if any numerical instability arises. The learning objective is to find the parameters $\mathtt{w}$ of the policy $\pi_{\mathtt{w}}$ that maximize the expected sum of rewards over the LES: $J(\mathtt{w}) = \mathbb{E}_{\pi_{\mathtt{w}}}\left[\sum_{t=1}^{T_{end}} r_t\right]$.

RL requires scalar reward signals measuring the agents' performance. Here we consider two reward functionals: The first, r^G, is based on the Germano identity (11, 12) which states that the sum of resolved and modeled contributions to the SGS stress tensor should be independent of LES resolution. The second, r^{LL}, penalizes discrepancies from the target energy spectra obtained by high-fidelity simulations (DNS) at a given Re_λ. While r^G is computed locally for each agent (see SI Appendix for details), r^{LL} is a global relation equal for all agents which measures the distance of the RL-LES statistics from those of fully resolved DNS. We remark that the target statistics involve spatial and temporal averages and can be computed from a limited number of DNS, which for this study are four orders of magnitude more computationally expensive than LES. This is an additional benefit of RL over SL, as it avoids the need of acquiring a large reservoir of training examples which should encompass all feasible flow realizations.

On the other hand, RL is known to require large quantities of interaction data, which in this case is acquired by performing LES with modest but non-negligible cost (which is orders of magnitude higher than the cost of ordinary differential equations or video games). Therefore, the design of a successful RL approach must take into account the actual computational implementation. Here we rely on the open-source RL library `smarties`*, which was designed to ease high-performance interoperability with existing simulation software. More importantly, we perform policy optimization with Remember Forget Experience Replay, ReF-ER (37). Three features of ReF-ER make it particularly suitable for the present task: First, it relies on Experience Replay (ER). ER improves the sample-efficiency of compatible RL algorithms by reusing experiences over multiple policy iterations and increases the accuracy of gradient updates by computing expectations from uncorrelated experiences. Second, ReF-ER is stable, reaches state-of-the-art performance on benchmark problems, and can even surpass optimal control methods on applicable problems (34). Third, and crucial to MARL, ReF-ER explicitly controls the pace of policy changes. Here agents collaborate to compute the SGS closure from partial state information, without explicitly coordinating their actions. Increasing N_{agents} improves the adaptability of the MARL to localized flow features. However, the RL gradients are defined for single agents in the environment; other agents' actions are confounding factors that increases the update variance (38). For example, if the C_s^2 coefficient selected by one agent causes numerical instability, all agents receive negative feedback, regardless of their choices. We found ReF-ER with strict policy constrains, which limits how much the policy is allowed to change from individual experiences, necessary to compensate for the imprecision of the RL update rules and to stabilize training.

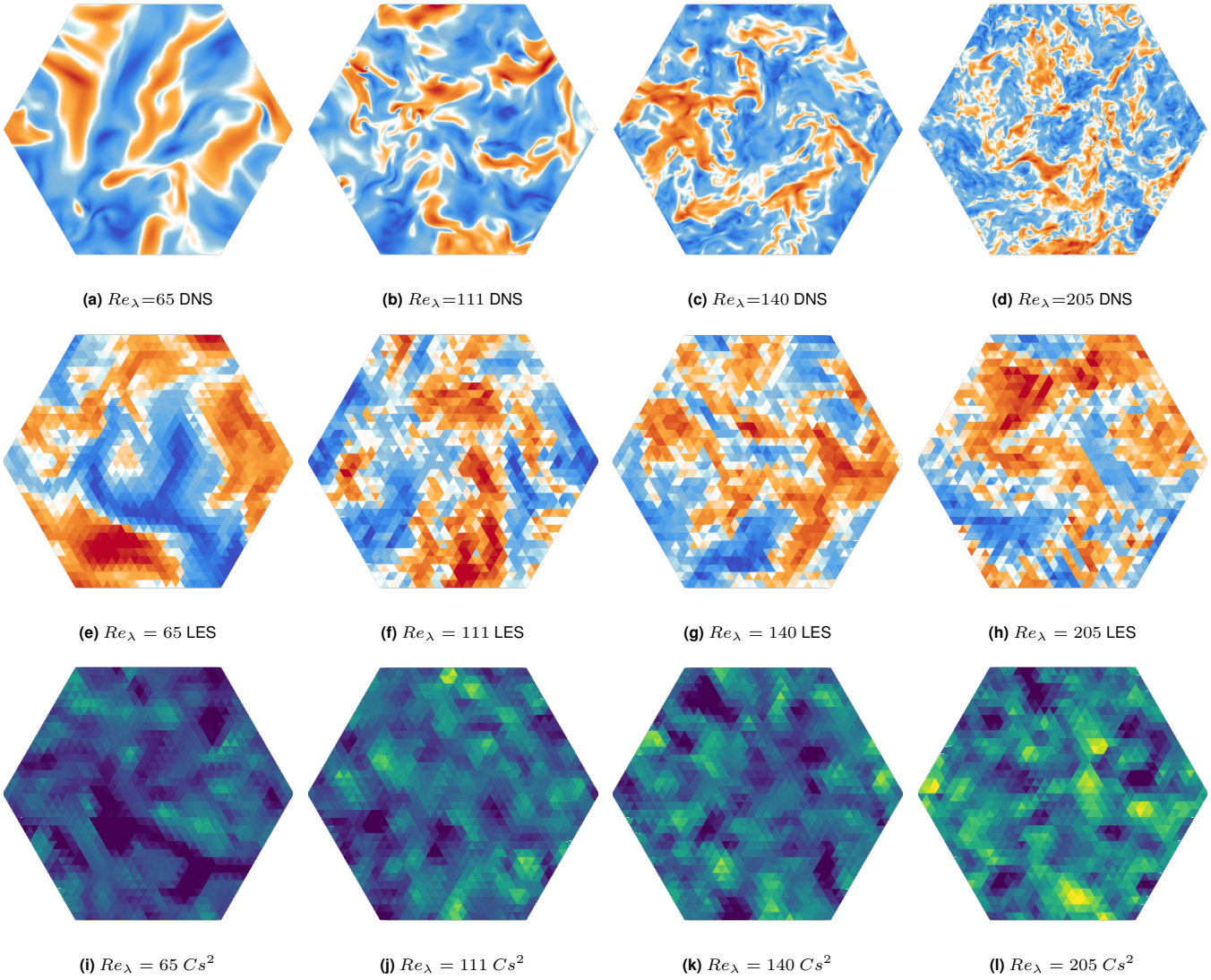
*https://github.com/cselab/smarties

Results

The Taylor-Reynolds number (Re_λ) characterizes the breadth of the spectrum of vortical structures present in a homogeneous isotropic turbulent (HIT) flow (3, 4, 39). Figure 2 illustrates the challenge in developing a reliable SGS model for a wide range of Re_λ and for a severely under-resolved grid. At the lower end of Reynolds numbers, e.g. $Re_\lambda = 65$, the SGS model is barely able to reproduce the flow features of DNS. However, for higher Re_λ all the qualitative visual features of DNS happen at length-scales that are much smaller than the LES grid-size. As the name suggests, only the large eddies are resolved and individual snapshots from Re_λ=111 to Re_λ=205 are visually indistinguishable. What changes are the time scales, which become faster, and the amount of energy contained in the SGS, which leads to instability if these are not accurately modeled.

In figure 4 we measure the accuracy of the LES by comparing the time-averaged LES energy spectra to the empirical log-normal distribution of energy spectra obtained via DNS. We consider the following SGS models: the RL policy $\pi_{\mathtt{w}}^G$ trained to maximize returns of the reward r^G (Eq. 19), the RL policy $\pi_{\mathtt{w}}^{LL}$ trained with the reward r^{LL} (Eq. 20), and two classical approaches SSM and DSM which stand for standard and dynamic Smagorinsky models respectively (see Sec. D). DSM, which is derived from the Germano identity is known to be more accurate and less numerically-diffusive. But, at the present LES resolution, DSM exhibits growing energy build-up at the smaller scales which causes numerical instability after $Re_\lambda \approx 140$. Surprisingly, the RL policy trained to satisfy the Germano identity is vastly over-diffusive. The reason is that $\pi_{\mathtt{w}}^G$ aims to minimize the Germano-error over all future steps, unlike DSM which minimizes the instantaneous Germano-error. Therefore, $\pi_{\mathtt{w}}^G$ picks actions that dissipate energy, smoothing the velocity field, and making it easier for future actions to minimize the Germano error. This is further confirmed by figure 4, which shows the empirical distribution of Smagorinsky coefficients chosen by the dynamic SGS models. While outwardly DSM and $\pi_{\mathtt{w}}^G$ minimize the same relation, $\pi_{\mathtt{w}}^G$ introduces much more artificial viscosity.

The most direct approach, rewarding the similarity of the energy spectra obtained by RL-LES to that of DNS, represented by $\pi_{\mathtt{w}}^{LL}$, produces the SGS model of the highest quality. The accuracy of the average spectrum is similar to DSM, but $\pi_{\mathtt{w}}^{LL}$ avoids the energy build-up and remains stable at higher Re_λ. Moreover, while DSM and $\pi_{\mathtt{w}}^{LL}$ have almost equal SGS dissipation (Fig. 7), we observe that $\pi_{\mathtt{w}}^{LL}$ achieves its accuracy by producing a narrower distribution of C_s^2. In this respect, $\pi_{\mathtt{w}}^{LL}$ stands in contrast to a model trained by SL to reproduce the SGS stresses computed from filtered DNS. By filtering the DNS results to the same resolution as the LES, thus isolating the unresolved scales, we emulate the distribution of C_s^2 that would be produced by a SGS model trained by SL. We find



(a) Re_λ=65 DNS **(b)** Re_λ=111 DNS **(c)** Re_λ=140 DNS **(d)** Re_λ=205 DNS

(e) $Re_\lambda = 65$ LES **(f)** $Re_\lambda = 111$ LES **(g)** $Re_\lambda = 140$ LES **(h)** $Re_\lambda = 205$ LES

(i) $Re_\lambda = 65\ Cs^2$ **(j)** $Re_\lambda = 111\ Cs^2$ **(k)** $Re_\lambda = 140\ Cs^2$ **(l)** $Re_\lambda = 205\ Cs^2$

Fig. 2. Representative contours of momentum flux across a diagonal slice ($x + y + z = 0$) of the cubical domain ($\boldsymbol{u} \cdot \boldsymbol{n}$, blue indicates negative flux) for DNS of homogeneous isotropic turbulence (HIT) with resolution 1024^3 (a-d), for LES with resolution 32^3 and SGS modeling with a RL policy trained for r^{LL} (e-h), and contours of the Smagorinsky coefficient C_s^2 across the same diagonal slice of the LES (i-l).

that such model would have lower SGS dissipation than both DSM and $\pi_{\mathtt{w}}^{LL}$, suggesting that, with the present numerical discretization schemes, it would produce numerically unstable LES (as pointed out by Refs. (27, 28)). This further highlights that the ability of RL to systematically optimize high-level objectives, such as matching the statistics of DNS, makes it a potent method to derive data-driven closure equations.

In figure 5, we evaluate the soundness of the RL-LES with values of Re_λ that were *not included* in the training. The numerical scales of flow quantities, and therefore the RL state components (Fig. 6), vary with Re_λ. The results for $Re_\lambda = 70$, 111, and 151 measure the RL-SGS model accuracy for dynamical scales that are interposed with the training ones. The results for $Re_\lambda = 60$, 176, 190 and 205 measures the ability of the RL-SGS model to generalize beyond the training set. For lower values of Re_λ, the DSM closure, with its decreased diffusivity, is marginally more accurate than the SGS model defined by $\pi_{\mathtt{w}}^{LL}$. However, $\pi_{\mathtt{w}}^{LL}$ remains valid and stable up to $Re_\lambda = 205$. Higher Reynolds numbers were not tested as they would have required increased spatial and temporal resolution to carry out accurate DNS, with increasingly prohibitive computational cost. We evaluate the difficulty of generalizing beyond the training data by comparing $\pi_{\mathtt{w}}^{LL}$ to a policy fitted exclusively for $Re_\lambda = 111$ ($\pi_{\mathtt{w}}^{LL,\ 111}$). Figure 6 shows the specialized policy to have higher accuracy at $Re_\lambda = 111$, but becomes rapidly invalid when varying the dynamical scales. This result supports that data-driven SGS models should be trained on varied flow conditions rather than with a training set produced by a single simulation.

The energy spectrum is just one of many statistical quantities that a physically sound LES should accurately reproduce. In figure 7 we compare the total kinetic energy (TKE), the characteristic length scale of the largest eddies (l_{int}), and dissipation rates among LES models and DNS. LES do not include the energy contained in the SGS and are more diffusive. Therefore, for a given energy injection rate ϵ, LES underestimate

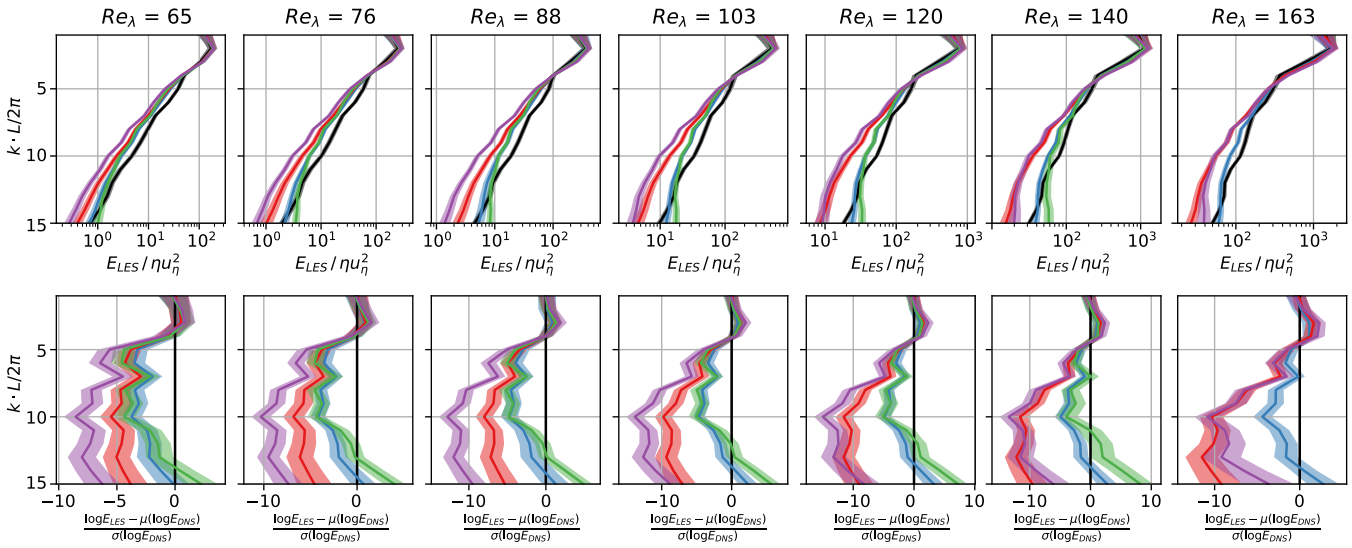


Fig. 3. Time-averaged energy spectra for LES of HIT for values of Re_λ that were *included* during the training phase of the RL agents for: (—) DNS, (—) SSM, (—) DSM, (—) RL policy trained for r^{LL}, and (—) RL policy trained for r^G. In the second row, for each Re_λ, we normalize the log-energy of each mode with the target mean and the corresponding standard deviation for k up to $N_{nyquist}$. This measure essentially quantifies the contributions of individual modes to the objective log-likelihood (Eq. 6). A perfect SGS model (—) would produce a spectrum with time-averaged $\log E_{DNS}^{Re_\lambda}(k)$ with the same statistics as DNS.

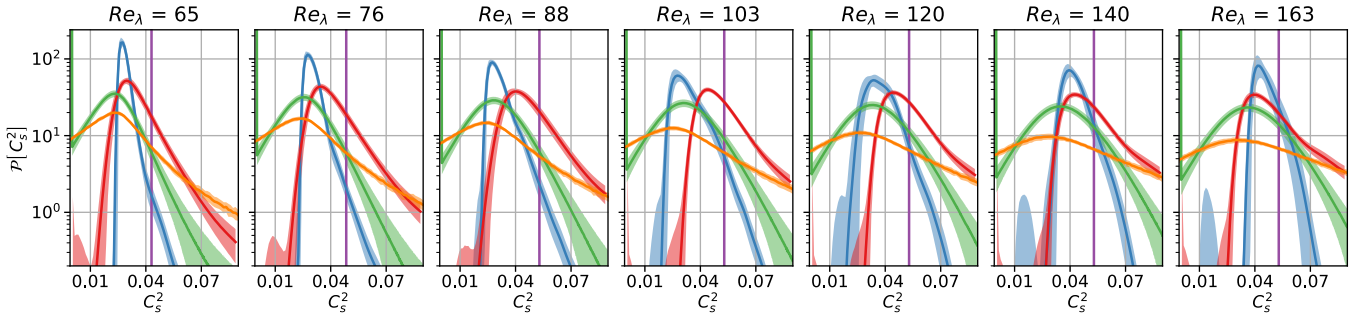


Fig. 4. Empirical probability distributions of the Smagorinsky model coefficient $\left(C_s^2\right)$ for values $Re_\lambda \in [65,\ 163]$ and: (—) SSM, (—) DSM, (—) RL agent with r^{LL}, (—) RL agent with r^G, and (—) optimal Smagorinsky computed computed from DNS filtered with uniform box test-filter.

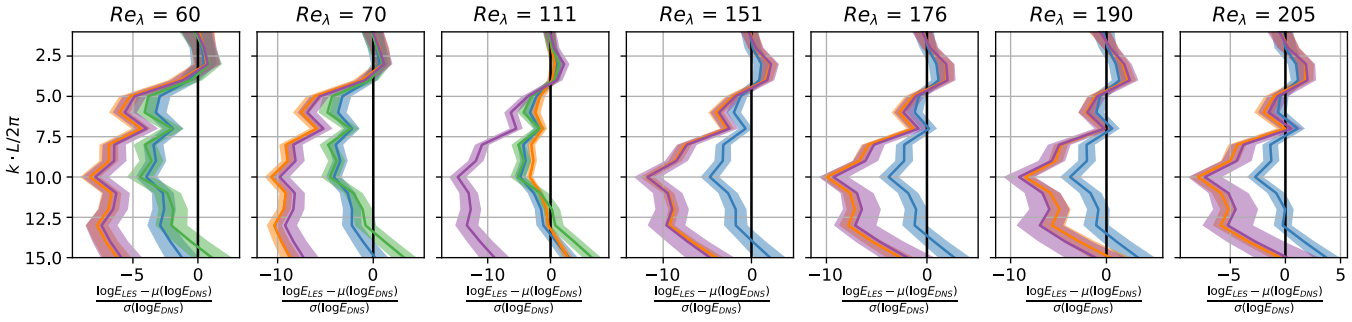


Fig. 5. Time-averaged energy spectra for LES of HIT normalized with mean and standard deviation obtained by DNS for values of Re_λ that were *not included* during the training phase of the RL agents for: (—) SSM, (—) DSM, (—) RL trained with r^{LL} for values of Re_λ shown in Fig. 3, and (—) RL trained with r^{LL} for $Re_\lambda = 111$.

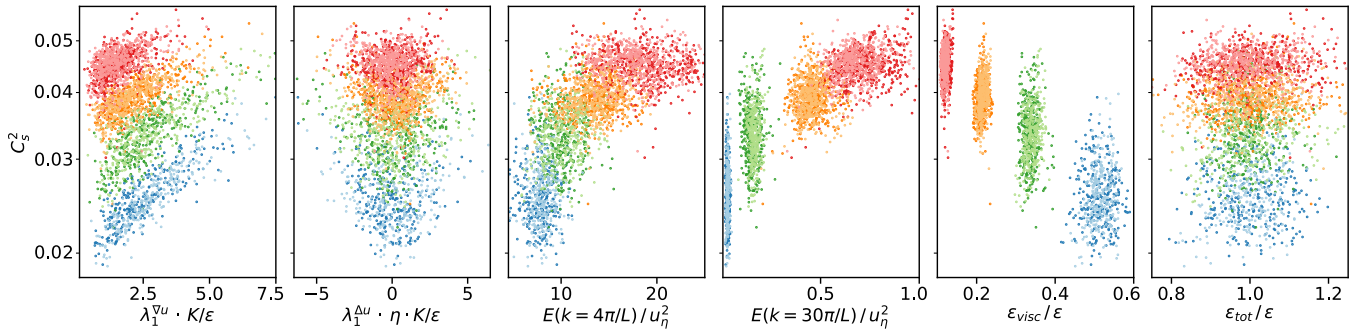


Fig. 6. Partial visualization of two independent realizations of the policy π_w^{LL}. Each figure shows 1000 values of C_s^2 (uniformly sub-sampled from a single simulation) computed by the mean of the trained policy (i.e. not stochastic samples) plotted against a single component of the RL-state vector. The colors correspond to (●) $Re_\lambda = 65$, (●) $Re_\lambda = 88$, (●) $Re_\lambda = 120$, (●) $Re_\lambda = 163$. Lighter and darker hues distinguish the two independent training runs.

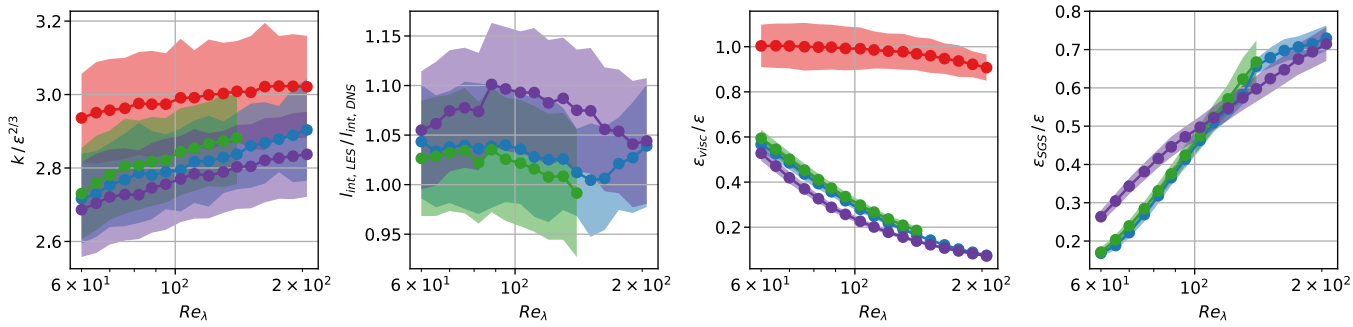


Fig. 7. Time averaged statistical properties of LES: turbulent kinetic energy, integral length scale, ratio of viscous dissipation to energy injection, and ratio of SGS dissipation to energy injection. The remaining component of energy dissipation is due to numerical discretization. (●) SSM, (●) DSM, (●) MARL with π_w^{LL}, and (●) DNS simulation, when applicable.

the TKE. Up to the point of instability at $Re_\lambda \approx 140$, DSM yields a better estimate for TKE and l_{int}. Despite these quantities not being directly included in the RL reward functional, π_w^{LL} remains more accurate than SSM. While in DNS energy is dissipated entirely by viscosity (and if under-resolved by numerical diffusion), in LES the bulk of viscous effects occur at length-scales below the grid size, especially at high Re_λ. We find that for $Re_\lambda = 205$ the SGS models dissipate approximately 10 times more energy than viscous dissipation, which underlines the crucial role of turbulence modeling.

Finally, we evaluate MARL across grid resolutions. Here, we keep the model defined by π_w^{LL}, trained for $N_{grid} = 32^3$, and evaluate it, along with DSM, for $N_{grid} = 64^3$ and 128^3. Accordingly, we increase the number of agents per simulation by a factor of 8 and 64. Figure 8 compares LES spectra up to each grid's Nyquist frequency (respectively, 15, 31 and 63). By construction, only the first 15 components of the spectra are available to π_w^{LL} ($N_{nyquist}$ for the grid size used for training). Finer resolutions are able to capture sharper velocity gradients, which are not experienced during training. As a consequence, π_w^{LL} was found to be markedly more diffusive than DSM, especially at the highest frequencies. However, as before, the SGS model derived by MARL remains stable throughout the evaluation and therefore allows, at $Re_\lambda \geq 180$, LES with spatial resolution reduced by a factor of 64 compared to DSM.

Discussion

This paper introduces multi-agent RL (MARL) as a strategy to automate the derivation of closure equations for turbulence modeling (TM). We demonstrate the feasibility and potential of this approach on large-eddy simulations (LES) of forced homogeneous and isotropic turbulence. RL agents are incorporated into the flow solver and observe local (e.g. invariants of the velocity gradient) as well as global (e.g. the energy spectrum) flow quantities. MARL develops the sub-grid scale (SGS) model as a policy that relates agent observations and actions. The agents cooperate to compute SGS residual-stresses of the flow field through the Smagorinsky (10) formulation in order to minimize the discrepancies between the time-averaged energy spectrum of the LES and that computed from fully resolved simulations (DNS), which are orders of magnitude more computationally expensive. The Remember and Forget Experience Replay (ReF-ER) method, combining the sample-efficiency of ER and the stability of constrained policy updates of on-policy RL (37), is instrumental for the present results.

The results of the present study open new horizons for TM. RL maximizes high-level objectives computed from direct application of the learned model and produces SGS models that are stable under perturbation and resistant to compounding errors. Moreover, MARL offers new paths to solve many of the classic challenges of LES, such as wall-layer modeling and inflow boundary conditions, which are difficult to formulate analytically or in terms of supervised learning (40). We empirically quantified and explored the ability of MARL to converge to an accurate model and to generalize to un-

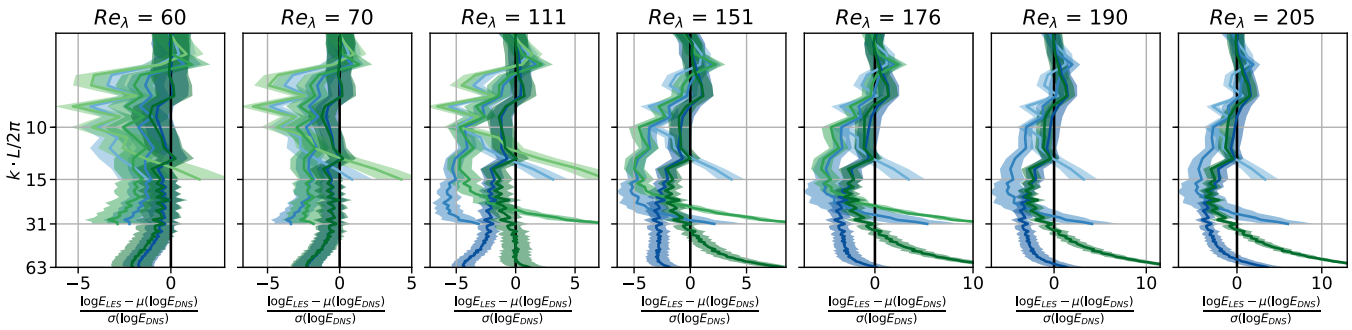


Fig. 8. Time-averaged energy spectra for LES of HIT normalized with mean ($\mu\,(\log E_{DNS})$) and standard deviation ($\sigma\,(\log E_{DNS})$) obtained by DNS for values of Re_λ that were *not included* during the training phase of the RL agents. The green curves are obtained by DSM with grid sizes $N_{grid} \in \{32^3,\ 64^3,\ 128^3\}$ (respectively, light to dark green). The blue curves are obtained by one RL policy trained for $N_{grid} = 32^3$ and evaluated on $N_{grid} \in \{32^3,\ 64^3,\ 128^3\}$ (respectively, light to dark blue).

seen flow conditions. At the same time new questions emerge from integrating RL and TM. The control policies trained by the present MARL method (e.g. $\pi_{\mathtt{w}}^{LL}$) are functions with 28-dimensional input and 6'211 parameters. Figure 6 provides some snapshots of the complex correlations between input and Smagorinsky coefficient selected by $\pi_{\mathtt{w}}^{LL}$. We observe that two independent training runs, over a range of Re_λ, produce overlapping distributions of actuation strategies, analogous to dynamical systems with the same attractor. While machine learning approaches can be faulted for the lack of generality guarantees and for the difficulty of interpreting the trained model, we envision that sparse RL methods could enable the analysis of causal processes in turbulent energy dissipation and the distillation of mechanistic models.

Materials and Methods

Methods We perform three dimensional simulations of the incompressible Navier-Stokes equations (NSE) for forced homogeneous and isotropic turbulence. The large-eddy simulations (LES) are based on the Smagorinsky model (10) of the sub-grid scale residual-stress tensor. Forcing turbulence allows to reach a statistically stationary flow by maintaining the large-scale motions over time. Both DNS and LES were performed on the open-source flow solver `CubismUP` [†]. The data-driven turbulence model is developed via a multi-agent reinforcement learning (MARL) formulation trained by Remember and Forget Experience Replay (37). Details regarding the simulation methods and the MARL algorithm are provided in the SI Appendix.

ACKNOWLEDGMENTS. We thank Dr. Jacopo Canton and Martin Boden (ETH Zurich) for several discussions throughout the course of this work. We acknowledge support by the European Research Council Advanced Investigator Award 341117. Computational resources were provided by Swiss National Supercomputing Centre (CSCS) Project s929.

1. AN Kolmogorov, The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. *Dokl. Akad. Nauk S.S.S.R.* **30**, 299–301 (1941).
2. M Li, P Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*. (Springer), (1997).
3. GI Taylor, Statistical theory of turbulence: Parts i-ii. *Proc. Royal Soc. London. Ser. A-Mathematical Phys. Sci.* **151**, 444–454 (1935).
4. SB Pope, Turbulent flows (2001).
5. JI Cardesa, A Vela-Martín, J Jiménez, The turbulent cascade in five dimensions. *Science* **357**, 782–784 (2017).
6. P Moin, K Mahesh, Direct numerical simulation: A tool in turbulence research. *Annu. Rev. Fluid Mech.* **30**, 539–578 (1998).
7. RD Moser, J Kim, NN Mansour, Direct numerical simulation of turbulent channel flow up to re τ= 590. *Phys. fluids* **11**, 943–945 (1999).
8. PA Durbin, Some recent developments in turbulence closure modeling. *Annu. Rev. Fluid Mech.* **50**, 77–103 (2018).
9. A Leonard, , et al., Energy cascade in large-eddy simulations of turbulent fluid flows. *Adv. Geophys. A* **18**, 237–248 (1974).
10. J Smagorinsky, General circulation experiments with the primitive equations: I. the basic experiment. *Mon. weather review* **91**, 99–164 (1963).
11. M Germano, U Piomelli, P Moin, WH Cabot, A dynamic subgrid-scale eddy viscosity model. *Phys. Fluids A: Fluid Dyn.* **3**, 1760–1765 (1991).
12. M Germano, Turbulence: the filtering approach. *J. Fluid Mech.* **238**, 325–336 (1992).
13. DK Lilly, A proposed modification of the germano subgrid-scale closure method. *Phys. Fluids A: Fluid Dyn.* **4**, 633–635 (1992).
14. C Lee, J Kim, D Babcock, R Goodman, Application of neural networks to turbulence control for drag reduction. *Phys. Fluids* **9**, 1740–1747 (1997).
15. M Milano, P Koumoutsakos, Neural network modeling for near wall turbulent flow. *J. Comput. Phys.* **182**, 1–26 (2002).
16. K Duraisamy, G Iaccarino, H Xiao, Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51**, 357–377 (2019).
17. J Ling, A Kurzawski, J Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166 (2016).
18. S Pope, A more general effective-viscosity hypothesis. *J. Fluid Mech.* **72**, 331–340 (1975).
19. F Sarghini, G De Felice, S Santini, Neural networks based subgrid scale modeling in large eddy simulations. *Comput. & fluids* **32**, 97–108 (2003).
20. M Gamahara, Y Hattori, Searching for turbulence models by artificial neural network. *Phys. Rev. Fluids* **2**, 054604 (2017).
21. C Xie, J Wang, H Li, M Wan, S Chen, Artificial neural network mixed model for large eddy simulation of compressible isotropic turbulence. *Phys. Fluids* **31**, 085112 (2019).
22. A Vollant, G Balarac, C Corre, Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures. *J. Turbul.* **18**, 854–878 (2017).
23. S Hickel, S Franz, N Adams, P Koumoutsakos, Optimization of an implicit subgrid-scale model for les in *Proceedings of the 21st International Congress of Theoretical and Applied Mechanics, Warsaw, Poland.* (2004).
24. R Maulik, O San, A neural network approach for the blind deconvolution of turbulent flows. *J. Fluid Mech.* **831**, 151–181 (2017).
25. K Fukami, K Fukagata, K Taira, Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid Mech.* **870**, 106–120 (2019).
26. JL Wu, H Xiao, E Paterson, Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Phys. Rev. Fluids* **3**, 074602 (2018).
27. B Nadiga, D Livescu, Instability of the perfect subgrid model in implicit-filtering large eddy simulation of geostrophic turbulence. *Phys. Rev. E* **75**, 046303 (2007).
28. A Beck, D Flad, CD Munz, Deep neural networks for data-driven les closure models. *J. Comput. Phys.* **398**, 108910 (2019).
29. M Gazzola, B Hejazialhosseini, P Koumoutsakos, Reinforcement learning and wavelet adapted vortex methods for simulations of self-propelled swimmers. *SIAM J. on Sci. Comput.* **36**, B622–B639 (2014).
30. S Verma, G Novati, P Koumoutsakos, Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proc. Natl. Acad. Sci.*, 201800923 (2018).
31. G Reddy, A Celani, T Sejnowski, M Vergassola, Learning to soar in turbulent environments. *Proc. Natl. Acad. Sci.*, 201606075 (2016).
32. L Biferale, F Bonaccorso, M Buzzicotti, P Clark Di Leoni, K Gustavsson, Zermelo's problem: Optimal point-to-point navigation in 2d turbulent flows using reinforcement learning. *Chaos: An Interdiscip. J. Nonlinear Sci.* **29**, 103138 (2019).
33. G Novati, et al., Synchronisation through learning for two self-propelled swimmers. *Bioinspiration & Biomimetics* **12**, 036001 (2017).
34. G Novati, L Mahadevan, P Koumoutsakos, Controlled gliding and perching through deep-reinforcement-learning. *Phys. Rev. Fluids* **4**, 093902 (2019).
35. JA Langford, RD Moser, Optimal les formulations for isotropic turbulence. *J. fluid mechanics* **398**, 321–346 (1999).
36. V Mnih, et al., Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
37. G Novati, P Koumoutsakos, Remember and forget for experience replay in *Proceedings of*

[†] https://github.com/novatig/CubismUP_3D

the 36th International Conference on Machine Learning. (2019).
38. L Buşoniu, R Babuška, B De Schutter, Multi-agent reinforcement learning: An overview in *Innovations in multi-agent systems and applications-1*. (Springer), pp. 183–221 (2010).
39. SA Orszag, GS Patterson, Numerical simulation of three-dimensional homogeneous isotropic turbulence. *Phys. Rev. Lett.* **28**, 76–79 (1972).
40. Y Zhiyin, Large-eddy simulation: Past, present and the future. *Chin. journal Aeronaut.* **28**, 11–24 (2015).
41. S Ghosal, TS Lund, P Moin, K Akselvoll, A dynamic localization model for large-eddy simulation of turbulent flows. *J. fluid mechanics* **286**, 229–255 (1995).
42. AJ Chorin, A numerical method for solving incompressible viscous flow problems. *J. computational physics* **2**, 12–26 (1967).
43. RS Rogallo, P Moin, Numerical simulation of turbulent flows. *Annu. review fluid mechanics* **16**, 99–137 (1984).
44. J Volker, *Large eddy simulation of turbulent incompressible flows: analytical and numerical results for a class of LES models*. (Springer Science & Business Media) Vol. 34, (2003).
45. S Gu, T Lillicrap, I Sutskever, S Levine, Continuous deep q-learning with model-based acceleration in *International Conference on Machine Learning*. pp. 2829–2838 (2016).
46. TP Lillicrap, et al., Continuous control with deep reinforcement learning in *International Conference on Learning Representations (ICLR)*. (2016).
47. Z Wang, et al., Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224* (2016).
48. T Degris, M White, RS Sutton, Off-policy actor-critic. *arXiv preprint arXiv:1205.4839* (2012).
49. R Munos, T Stepleton, A Harutyunyan, M Bellemare, Safe and efficient off-policy reinforcement learning in *Advances in Neural Information Processing Systems*. pp. 1054–1062 (2016).
50. X Glorot, Y Bengio, Understanding the difficulty of training deep feedforward neural networks in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. pp. 249–256 (2010).
51. DP Kingma, J Ba, Adam: A method for stochastic optimization in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. (2014).
52. J Chung, C Gulcehre, K Cho, Y Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).
53. I Sutskever, *Training recurrent neural networks*. (University of Toronto Toronto, Ontario, Canada), (2013).

Supporting Information Appendix for "Automating Turbulence Modeling by Multi-Agent Reinforcement Learning"

1. Forced Homogeneous and Isotropic Turbulence

We use as a benchmark problem the simulations of Forced Homogeneous Isotropic Turbulence (F-HIT) with a linear, low-wavenumber forcing term. These methods have been implemented in the open-source three-dimensional incompressible Navier-Stokes solver `CubismUP`[‡].

A. Turbulent Kinetic Energy. A turbulent flow is *homogeneous* and *isotropic* when the averaged quantities of the flow are invariant under arbitrary translations and rotations. The flow statistics are independent of space and the mean velocity of the flow is zero. Forced, homogeneous, isotropic turbulence is governed by the incompressible Navier-Stokes equations,

$$\begin{cases} \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} & = -\nabla p + \nabla\cdot(2\nu S) + \boldsymbol{f} \\ \nabla\cdot\boldsymbol{u} & = 0 \end{cases} \quad [1]$$

where $S = \frac{1}{2}(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)$ is the rate-of-strain tensor. The the turbulent kinetic energy (TKE, the second order statistics of the velocity field) is expressed as:

$$e(\boldsymbol{x},t) \equiv \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}, \quad K(t) \equiv \frac{1}{2}\left\langle \boldsymbol{u}\cdot\boldsymbol{u}\right\rangle, \quad [2]$$

where the angle brackets $\langle\cdot\rangle \equiv \frac{1}{\mathcal{V}}\int_{\mathcal{D}}\cdot$ denote an ensemble average over the domain $\mathcal{D}$ with volume $\mathcal{V}$. For a flow with *periodic boundary conditions* the evolution of the kinetic energy is described as:

$$\frac{dK}{dt} = -\nu\int_{\mathcal{D}}\|\nabla\boldsymbol{u}\|^2 + \int_{\mathcal{D}}\boldsymbol{u}\boldsymbol{f} \;= -2\nu\left\langle Z\right\rangle + \left\langle \boldsymbol{u}\cdot\boldsymbol{f}\right\rangle \quad [3]$$

where the energy dissipation due to viscosity, is expressed in term of the norm of the vorticity $\boldsymbol{\omega} \equiv \nabla\times\boldsymbol{u}$ and the *enstrophy* $Z = \frac{1}{2}\boldsymbol{\omega}^2$. This equation clarifies that the vorticity of the flow field is responsible for energy dissipation that can only be conserved if there is a source of energy.

We investigate the behaviour of homogeneous isotropic turbulence in a statistically stationary state by injecting energy through forcing. In generic flow configurations the role of this forcing is taken up by the large-scale structures and it is assumed that it does not influence smaller scale statistics, which are driven by viscous dissipation. The injected energy is transferred from large-scale motion to smaller scales due to the non-linearity of Navier-Stokes equations. We implement a classic low-wavenumber (low-k) forcing term (41) for homogeneous isotropic turbulence that is proportional to the local fluid velocity as filtered from its large wave number components:

$$\tilde{\boldsymbol{f}}(\boldsymbol{k},t) \equiv \alpha\; G(\boldsymbol{k},k_f)\;\tilde{\boldsymbol{u}}(\boldsymbol{k},t) = \alpha\;\tilde{\boldsymbol{u}}_<(\boldsymbol{k},t), \quad [4]$$

where the tilde symbol denotes a three-dimensional Fourier transform, $G(\boldsymbol{k},k_f)$ is a low-pass filter with cutoff wavelength k_f, α a constant, and $\tilde{\boldsymbol{u}}_<$ is the filtered velocity field. By applying Parseval's theorem, the rate-of-change of energy in the system due to the force is:

$$\left\langle \boldsymbol{f}\cdot\boldsymbol{u}\right\rangle = \frac{1}{2}\sum_{\boldsymbol{k}}\left(\tilde{\boldsymbol{f}}^*\cdot\tilde{\boldsymbol{u}} + \tilde{\boldsymbol{f}}\cdot\tilde{\boldsymbol{u}}^*\right) = \alpha\sum_{\boldsymbol{k}}\tilde{\boldsymbol{u}}_<^2 = 2\alpha K_<. \quad [5]$$

Here, $K_<$ is the kinetic energy of the filtered field. We set $\alpha = \epsilon/2K_<$ and $k_f = 4\pi/L$, meaning that we simulate a time-constant rate of energy injection ϵ which forces only the seven lowest modes of the energy spectrum. The constant injection rate is counter-balanced by the viscous dissipation $\epsilon_{visc} = 2\nu\left\langle Z\right\rangle$, the dissipation due to the numerical errors ϵ_{num}, and, by a subgrid-scale (SGS) model of turbulence (ϵ_{sgs}, when it is employed - see Sec. D). When the statistics of the flow reach steady state, the time-averaged total rate of energy dissipation $\epsilon_{\text{tot}} = \epsilon_{visc} + \epsilon_{\text{num}} + \epsilon_{\text{sgs}}$ is equal to the rate of energy injection ϵ.

B. The Characteristic Scales of Turbulence. Turbulent flows are characterized by a large separation in temporal and spatial scales and long-term dynamics. These scales can be estimated by means of dimensional analysis, and can be used to characterize turbulent flows. At the *Kolmogorov scales* energy is dissipated into heat: $\eta = \left(\nu^3/\epsilon\right)^{1/4}$, $\tau_\eta = (\nu/\epsilon)^{1/2}$, $u_\eta = (\epsilon\nu)^{1/4}$. These quantities are independent of large-scale effects including boundary conditions or external forcing. The *integral scales* are the scales of the largest eddies of the flow: $l_{\text{I}} = \frac{3\pi}{4K}\int_0^\infty \frac{\tilde{E}(k)}{k}\mathrm{d}k$, $\tau_{\text{I}} = \frac{l_{\text{I}}}{\sqrt{2K/3}}$. The Taylor-Reynolds number is used to characterize flows with zero mean bulk velocity:

$$Re_\lambda = K\sqrt{\frac{20}{3\nu\epsilon}}$$

Under the assumptions of homogeneous and isotropic flow we study the statistical properties of turbulence in Fourier space. In the text, unless explicitly stated, we analyze quantities computed from simulations at statistically steady state and we omit the temporal dependencies. The energy spectrum $\tilde{E}(k)$, which can be derived from the two point velocity correlation tensor, is $\tilde{E}(k) \equiv \frac{1}{2}\tilde{\boldsymbol{u}}^2(k)$. Kolmogorov's theory of turbulence predicts the well-known $-\frac{5}{3}$ spectrum (i.e. $\tilde{E}(k) \propto \epsilon^{2/3}k^{-5/3}$) for the turbulent energy in the inertial range $k_{\text{I}} \ll k \ll k_\eta$.

C. Direct Numerical Simulations (DNS). Data from DNS serve as reference for the SGS models and as targets for creating training rewards for the RL agents. The DNS are carried out on a uniform grid of size 512^3 for a periodic cubic domain $(2\pi)^3$. The solver is based on finite differences, third-order upwind for advection and second-order centered differences for diffusion, and pressure projection (42). Time stepping is performed with second-order explicit Runge-Kutta with variable integration step-size determined with a Courant–Friedrichs–Lewy (CFL) coefficient $CFL = 0.1$. We performed DNS for Taylor-Reynolds numbers in log increments between $Re_\lambda \in [60, 205]$ (Fig. 1a). The initial velocity field is synthesized using the procedure of (43) by generating a distribution of random Fourier coefficients matching a radial target spectrum $\tilde{E}(k)$: $\tilde{E}(k) = c_k\;\epsilon^{2/3}k^{-5/3}\;f_L(kL)\;f_\eta(k\eta)$, where $f_l(kl_{\text{I}}) = \left[\frac{kl_{\text{I}}}{\sqrt{(kl_{\text{I}})^2+c_l}}\right]^{5/3+p_0}$ and $f_\eta(k\eta) = \exp\left\{-\beta\left[\sqrt[4]{(k\eta)^4 + c_\eta^4} - c_\eta\right]\right\}$

[‡] https://github.com/novatig/CubismUP_3D

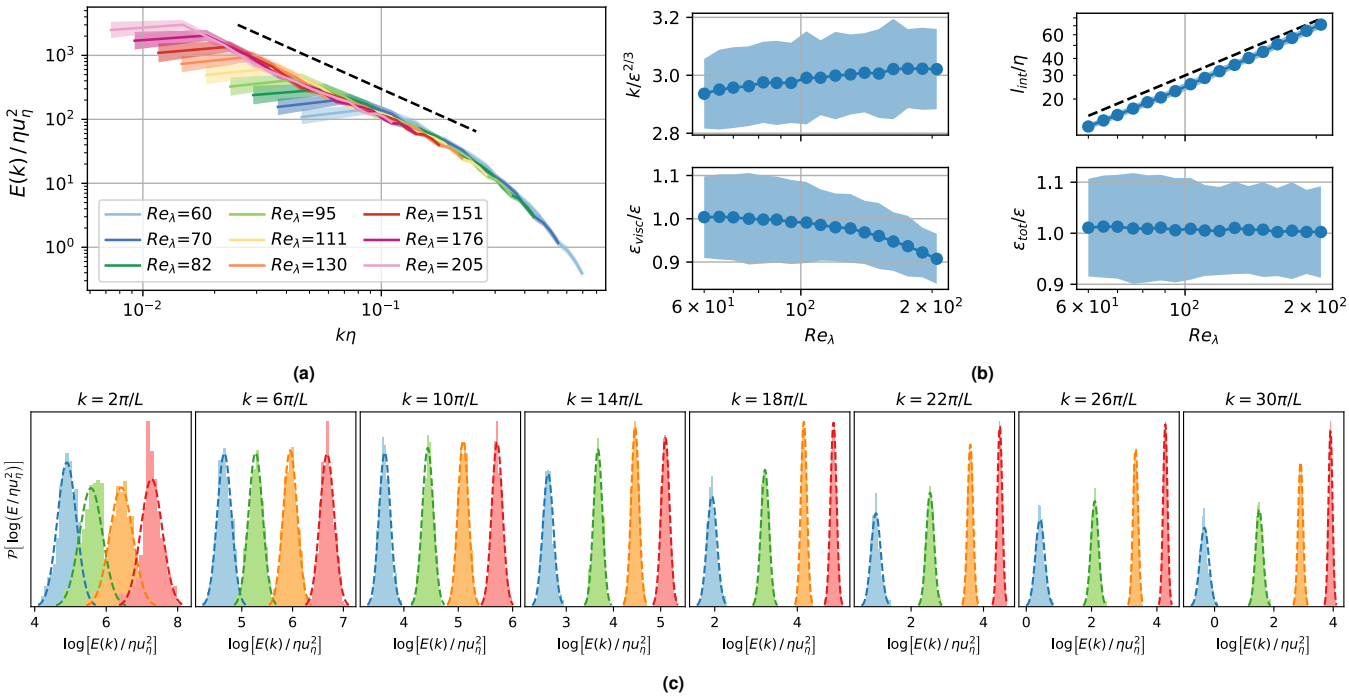


Figure S 1. (a) Time averaged energy spectra for DNS simulations of Forced Homogeneous Isotropic Turbulence (F-HIT) for log-increments of $Re_\lambda \in [60, 205]$ compared to Kolmogorov's spectrum $\propto k^{-5/3}$ (dashed line). (b) Time averaged statistical quantities of the flow as function of Re_λ. From left to right and top to bottom: the average TKE is approximately proportional to $\epsilon^{2/3}$, the ratio of integral length scale to η compared to the relation predicted by Kolmogorov scaling $\propto Re_\lambda^{4/3}$ (4) (dashed line), the ratio of viscous dissipation to energy injection, and the total dissipation (viscous and numerics) is on average equal to energy injection. (c) Distributions of values of the energy spectrum for single modes at $Re_\lambda = 65$ (blue), 88 (green), 110 (orange), and 163 (red).

determine the spectrum in the integral- and the dissipation-ranges respectively and the constants $p_0 = 4$, $\beta = 5.2$ are fixed (4). Further, we set $c_l = 0.001$, $c_\eta = 0.22$, and $c_k = 2.8$. The choice of initial spectrum determines how quickly the F-HIT simulation reaches statistical steady state, at which point Re_λ fluctuates around a constant value. The time-averaged quantities (Fig. 1) are computed from 20 independent DNS with measurements taken every τ_η. Each DNS lasts 20 τ_I and the initial 10 τ_I are not included in the measurements, which found to be ample time to avoid the initial transient. Figure 1c shows that the distribution of energy content for each mode $\tilde{E}(k)$ is well approximated by a log-normal distribution such that $\log E_{DNS}^{Re_\lambda} \sim \mathcal{N}\left(\mu_{DNS}^{Re_\lambda},\ \Sigma_{DNS}^{Re_\lambda}\right)$, where $\mu_{DNS}^{Re_\lambda}$ is the empirical average of the log-energy spectrum for a given Re_λ and $\Sigma_{DNS}^{Re_\lambda}$ is its covariance matrix. When comparing SGS models and formulating objective functions, we will extensively rely on a regularized log-likelihood given the collected DNS data:

$$\widetilde{LL}(E_{LES}^{Re_\lambda}|E_{DNS}^{Re_\lambda}) = \log \mathcal{P}(E_{LES}^{Re_\lambda}|E_{DNS}^{Re_\lambda})/N_{nyquist}. \quad [6]$$

Here $N_{nyquist}$ is the Nyquist frequency of the LES grid and the probability metric is

$$\mathcal{P}(E_{LES}^{Re_\lambda}|E_{DNS}^{Re_\lambda}) \propto \exp\left[-\frac{1}{2}\left(\log E_{LES}^{Re_\lambda} - \bar{\mu}_{DNS}^{Re_\lambda}\right)^T \left(\bar{\Sigma}_{DNS}^{Re_\lambda}\right)^{-1} \left(\log E_{LES}^{Re_\lambda} - \bar{\mu}_{DNS}^{Re_\lambda}\right)\right] \quad [7]$$

with E_{LES} the time-averaged energy spectrum of the LES up to $N_{nyquist}$, $\bar{\mu}_{DNS}^{Re_\lambda}$ and $\bar{\Sigma}_{DNS}^{Re_\lambda}$ the target statistics up to $N_{nyquist}$.

D. Large-Eddy Simulations (LES). LES (9) resolve the large scale dynamics of turbulence and model their interaction with the sub grid-scales (SGS). The flow field $\bar{\boldsymbol{u}}$ on the grid is viewed as the result of *filtering* out the residual small-scales of a latent velocity field $\boldsymbol{u}$. The filtered velocity field is expressed as:

$$\bar{\boldsymbol{u}}(\boldsymbol{x}) = (\mathcal{G} * \boldsymbol{u})(\boldsymbol{x}), \quad [8]$$

where $*$ denotes a convolution product, and $\mathcal{G}$ is some filter function. The filtered Navier-Stokes equation for the field $\bar{\boldsymbol{u}}$ reads:

$$\frac{\partial \bar{\boldsymbol{u}}}{\partial t} + (\bar{\boldsymbol{u}} \cdot \nabla)\,\bar{\boldsymbol{u}} = -\nabla \bar{p} + \nabla \cdot \left(2\nu \bar{S} - \tau^R\right) + \bar{\boldsymbol{f}} \quad [9]$$

here, the *residual-stress tensor* τ^R encloses the interaction with the unresolved scales:

$$\tau^R = \overline{\boldsymbol{u} \otimes \boldsymbol{u}} - \bar{\boldsymbol{u}} \otimes \bar{\boldsymbol{u}}. \quad [10]$$

Closure equations are used to model the sub grid-scale motions represented by $\overline{\boldsymbol{u} \otimes \boldsymbol{u}}$.

The Classic Smagorinsky Model (SSM) (10) is a linear eddy-viscosity model that relates the residual stress-tensor to the filtered rate of strain

$$\tau^R - \frac{1}{3}\mathrm{tr}\left(\tau^R\right) = -2\,\nu_t\,\bar{S}, \quad [11]$$

$$\nu_t = (C_s \Delta)^2\ \|\bar{S}\|, \quad [12]$$

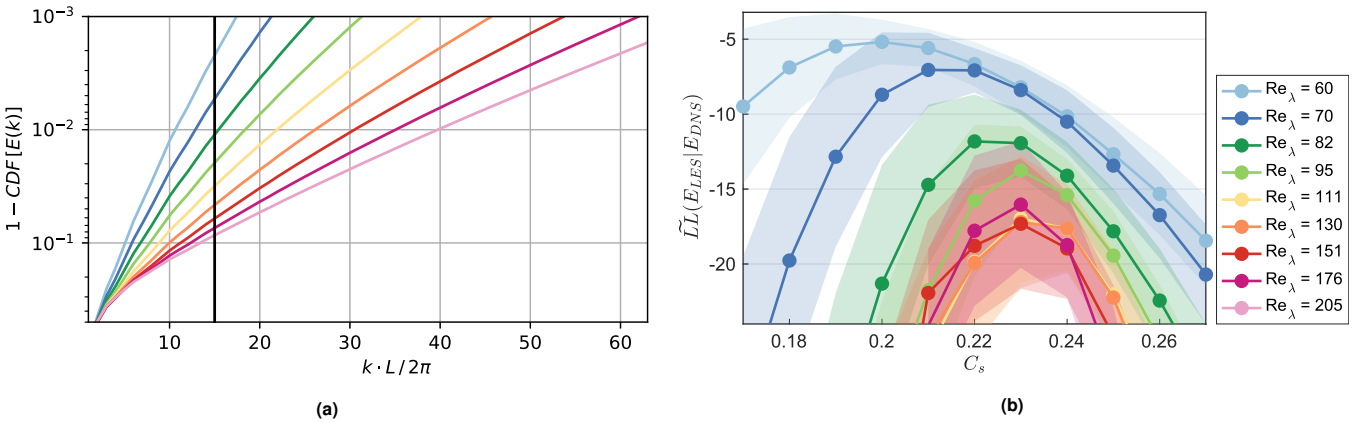


Figure S 2. (a) Time-averaged cumulative fraction of the TKE contained up to mode k for DNS simulations of F-HIT for log-increments of $Re_\lambda \in [60, 205]$ (the legend is on the right). The black vertical line corresponds to the Nyquist frequency for the grid size ($N = 32^3$) used for all LES considered throughout this study. (b) Time-averaged regularized log-likelihood (equation 6) obtained for SGS simulations as function of the C_s constant.

where Δ is the grid size and C_s is the Smagorinsky constant. This model has been shown to perform reasonably well for isotropic homogeneous turbulence and wall-bounded turbulence. The rate of transfer of energy to the residual motions, derived from the filtered energy equation, is $2\nu_t\|\bar{S}\|^2$ (4), which is always positive since $\nu_t > 0$. The energy transfer is then always from the filtered motions to the residual motions, it is proportional to the turbulent eddy-viscosity ν_t, and there is no backscatter. The Smagorinsky model closes the filtered Navier-Stokes equation together with an *a priori* prescription for the constant C_s. The main drawbacks of this model, as exposed in (44), are that (a) the turbulent eddy-viscosity does not necessarily vanish for laminar flows, (b) the Smagorinsky constant is an *a priori* input which has to be tuned to represent correctly various turbulent flows, (c) the model introduces generally too much dissipation.

The Dynamic Smagorinsky Model (DSM) (11) computes the parameter $C_s(\boldsymbol{x}, t)$ as a function of space and time. DSM's dynamic model is obtained by filtering equation 9 a second time with a so-called test filter of size $\widehat{\Delta} > \Delta$. The resolved-stress tensor $\mathcal{L}$ is defined by the Germano identity:

$$\mathcal{L}_{\overline{\boldsymbol{u}}} = \widehat{\overline{\boldsymbol{u}} \otimes \overline{\boldsymbol{u}}} - \widehat{\overline{\boldsymbol{u}}} \otimes \widehat{\overline{\boldsymbol{u}}} = T^R - \widehat{\tau^R}, \tag{13}$$

where $T^R = \widehat{\overline{\boldsymbol{u}} \otimes \boldsymbol{u}} - \widehat{\overline{\boldsymbol{u}}} \otimes \widehat{\overline{\boldsymbol{u}}}$ is the residual-stress tensor for the test filter width $\widehat{\Delta}$, and $\widehat{\tau^R}$ is the test-filtered residual stress tensor for the grid size Δ (Eq. 10). If both residual stresses are approximated by a Smagorinsky model, the Germano identity becomes:

$$\mathcal{L}_{\overline{\boldsymbol{u}}} \approx 2\, C_s^2(\boldsymbol{x}, t)\, \Delta^2 \left[\widehat{\|\bar{S}\|\bar{S}} - \frac{\widehat{\Delta}^2}{\Delta^2} \|\widehat{\bar{S}}\|\widehat{\bar{S}} \right]. \tag{14}$$

The dynamic Smagorinsky parameter (Eq. 14) forms an over-determined system for $C_s^2(\boldsymbol{x}, t)$, whose least-squares solution is (13):

$$C_s^2(\boldsymbol{x}, t) = \frac{\langle \mathcal{L}_{\overline{\boldsymbol{u}}}, \mathcal{M} \rangle_F}{2\Delta^2 \|\mathcal{M}\|^2}, \tag{15}$$

where $\mathcal{M} = \widehat{\|\bar{S}\|\bar{S}} - (\widehat{\Delta}/\Delta)^2 \|\widehat{\bar{S}}\|\widehat{\bar{S}}$, and $\langle\cdot\rangle_F$ is the Frobenius product. Because the dynamic coefficient may take negative values, which represents energy transfer from the unresolved to the resolved scales, C_s^2 is clipped to positive values for numerical stability.

The fraction of TKE contained in the unresolved scales increases with Re_λ and decreases with the grid size (Fig. 2a). For all LES considered in this study we employ a grid of size $N = 32^3$, as compared to $N = 512^3$ for the DNS. For the higer Re_λ, the SGS model accounts for up to 10% of the total TKE. We employ second-order centered discretization for the advection and the initial conditions for the velocity field are synthesized from the time-averaged DNS spectrum at the same Re_λ (43). When reporting results from SSM simulation, we imply the Smagorinsky constant C_s resulting from line-search optimization with step size 0.005 (Fig. 2b). LES statistics are computed from simulations up to $t = 100\tau_\mathrm{I}$, disregarding the initial $10\tau_\mathrm{I}$ time units. For the DSM procedure we employ an uniform box test-filter of width $\widehat{\Delta} = 2\Delta$. Finally, DSM spectra are obtained with time-stepping coefficient $CFL = 0.01$, while $CFL = 0.1$ was used for all other LES.

2. Multi-agent Reinforcement Learning

We introduce a RL formulation for the SGS model as illustrated in figure 1. In RL agents observe the state of the environment, perform actions and receive rewards. Their goal is to develop a policy for their actions so as to maximize their long term reward. In this work we consider N_{agents} RL agents in the simulation domain with $N_{agents} \leq N$ (i.e. there is at most one agent per grid-point, marked as red cubes in figure 1). Each agent receives both local and global information about the state of the simulation encoded as $\boldsymbol{s}(\boldsymbol{x}, t) \in \mathbb{R}^{dim_S}$. In order to embed tensorial invariance into the NN inputs (17), the local components of the state vector are the 5 invariants of the gradient (18) and the 6 invariants of the Hessian of the velocity field computed at the agents' location and non-dimensionalized with K/ϵ. The global components of the state are the energy spectrum up to the grid's Nyquist frequency $N_{nyquist}$ non-dimensionalized with u_η, the ratio of viscous dissipation ϵ_{visc}/ϵ and total dissipation ϵ_{tot}/ϵ relative to the energy injection rate ϵ. For $N = 32^3$, we have $N_{nyquist} = 15$ and $dim_S = 28$, far fewer variables than the full state of the system (i.e. the entire velocity field and $dim_S = 3 \cdot 32^3$).

Given the state, the agents perform one action by sampling a Gaussian policy $a(\boldsymbol{x}, t) \sim \pi_{\boldsymbol{w}}(\cdot \,|\, \boldsymbol{s}(\boldsymbol{x}, t)) \equiv \mathcal{N}[\mu_{\boldsymbol{w}}(\boldsymbol{s}(\boldsymbol{x}, t)), \sigma_{\boldsymbol{w}}(\boldsymbol{s}(\boldsymbol{x}, t))]$ with $a(\boldsymbol{x}, t) \in \mathbb{R}$. The agents are uniformly dispersed in the domain with distance $\Delta_{agents} = 2\pi \sqrt[3]{N/N_{agents}}$. The action corresponds to the

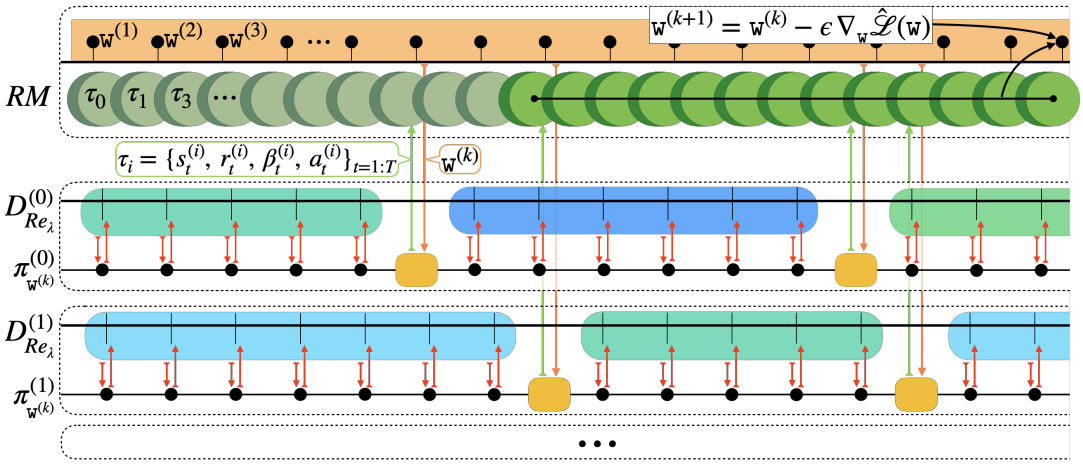


Figure S 3. Schematic description of the training procedure implemented with the `smarties` library. Each dashed line represents a computational node. Worker processes receive updated policy parameters $\mathtt{w}^{(k)}$ and run LES for randomly sampled Re_λ. At the top, a master process receives RL data from completed simulations. Policy updates are computed by sampling mini-batches from the N most recently collected RL steps.

local Smagorinsky coefficient and is interpolated to the grid according to the three-dimensional triangular kernel:

$$C_s^2(\boldsymbol{x},t) = \sum_{i=1}^{N_{agents}} a(\boldsymbol{x}_i,t) \prod_{j=1}^{3} \max\left\{1 - \frac{|x^{(j)} - x_i^{(j)}|}{\Delta_{agents}},\ 0\right\}, \quad [16]$$

where $\boldsymbol{x}_i$ is the location of agent i, and $x^{(j)}$ is the j-th Cartesian component of the position vector. If $N_{agents} = N$, no interpolation is required. The learning objective is to find the parameters $\mathtt{w}$ of the policy $\pi_{\mathtt{w}}$ that maximize the expected sum of rewards over the LES. The reward can be cast in both local and global relations. We define a reward functional such that the optimal $\pi_{\mathtt{w}}$ yields a stable SGS model for a wide range of Re_λ with statistical properties closely matching those of DNS. The base reward is a distance measure from the target DNS, derived from the regularized log-likelihood (Eq. 6):

$$r^{grid}(t) = \exp\left[-\sqrt{-\widetilde{LL}(\langle\tilde{E}\rangle(t)|E_{DNS}^{Re_\lambda})}\right]. \quad [17]$$

This regularized distance is preferred because a reward directly proportional to the probability $\mathcal{P}(\langle\tilde{E}\rangle(t)|E_{DNS}^{Re_\lambda})$ vanishes to zero too quickly for imperfect SGS models and therefore yields too flat an optimization landscape. The average LES spectrum is computed with an exponential moving average with effective window Δ_{RL}:

$$\langle\tilde{E}\rangle(t) = \langle\tilde{E}\rangle(t-\delta t) + \frac{\delta t}{\Delta_{RL}}\left(\tilde{E}(t) - \langle\tilde{E}\rangle(t-\delta t)\right) \quad [18]$$

We consider two variants for the reward. The first adds a local term, non-dimensionalized with u_τ^4, to reward actions that satisfy the Germano identity (Eq. 13):

$$r^G(\boldsymbol{x},t) = r^{grid}(t) - \frac{1}{u_\tau^4}\|\mathcal{L}_{\overline{\boldsymbol{u}}}(\boldsymbol{x},t) - T^R(\boldsymbol{x},t) + \tau^{\widehat{R}}(\boldsymbol{x},t)\|^2 \quad [19]$$

The second reward is a purely global quantity to further reward matching the DNS:

$$r^{LL}(t) = r^{grid}(t) + \frac{\tau_\eta}{\Delta_{RL}}\left[\widetilde{LL}(\langle\tilde{E}\rangle(t)|E_{DNS}^{Re_\lambda}) - \widetilde{LL}(\langle\tilde{E}\rangle(t-\Delta_{RL})|E_{DNS}^{Re_\lambda})\right] \quad [20]$$

Which can be interpreted as a non-dimensional derivative of the log-likelihood over the RL step, or a measure of the contribution of each round of SGS model update to the instantaneous accuracy of the LES. We note that r^{LL} is equal for all agents.

A. The Reinforcement Learning framework. RL algorithms advance by trial-and-error exploration and are known to require large quantities of interaction data, in this case thousands of LES. As mentioned in the main text, we interface the flow solver with the RL library `smarties`. `smarties` efficiently leverages the computing resources by separating the task of updating the policy parameters from the task of collecting interaction data (Fig. 3). The flow simulations are distributed across $N_{workers}$ computational nodes ("workers"). The workers collect, for each agent, experiences organized into episodes:

$$\tau_i = \left\{s_t^{(i)}, r_t^{(i)}, \mu_t^{(i)}, \sigma_t^{(i)}, a_t^{(i)}\right\}_{t=0:T_{end}^{(i)}}$$

where t tracks in-episode RL steps; $\mu_t^{(i)}$ and $\sigma_t^{(i)}$ are the statistics of the Gaussian policy used to sample $a_t^{(i)}$ with the policy parameters available to the worker at time step t of the i-th episode, often termed "behavior policy" $\beta_t^{(i)} \equiv \mathcal{N}(\mu_t^{(i)}, \sigma_t^{(i)})$ in the off-policy RL literature. When a simulation concludes, the worker sends one episode per agent to the central learning process ("master") and receives updated policy parameters. The master stores the episodes into a Replay Memory (RM), which is sampled to update the policy parameters according to *Remember and Forget Experience Replay* (ReF-ER, (37)).

ReF-ER can be combined with many ER-based RL algorithms as it consists in a modification of the optimization objective. For example, it has been applied to Q-learning (e.g. NAF (45)), deterministic policy gradients (46), off-policy policy gradients (47). Here we

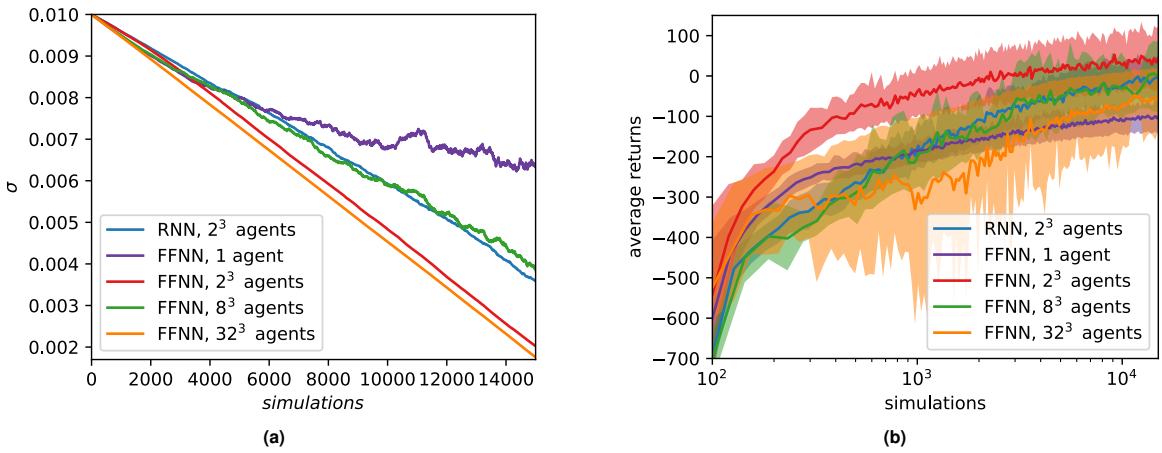


Figure S 4. Progress of average returns (b, cumulative rewards over a simulation) and policy standard deviation $\sigma_{\mathtt{w}}$ (a) during training for varying numbers of agents in the simulation domain. If fewer than 32^3 (grid size) agents are placed in the domain, the SGS coefficient $\left(C_s^2\right)$ is computed throughout the grid by linear interpolation.

employ V-RACER, a variant of off-policy policy optimization proposed in conjunction with ReF-ER (37) which supports continuous state and action spaces. V-RACER trains a Neural Network (NN) which, given input s_t, outputs the mean $\mu_{\mathtt{w}}(s_t)$ and standard deviation $\sigma_{\mathtt{w}}(s_t)$ of the policy $\pi_{\mathtt{w}}$, and a state-value estimate $v_{\mathtt{w}}(s_t)$. One gradient is defined per NN output. The statistics $\mu_{\mathtt{w}}$ and $\sigma_{\mathtt{w}}$ are updated with the *off-policy policy gradient* (*off-PG*) (48):

$$g^{\text{pol}}(\mathtt{w}) = \mathbb{E}\left[\left(\hat{q}_t - v_{\mathtt{w}}(s_t)\right)\frac{\pi_{\mathtt{w}}(a_t|s_t)}{\mathcal{P}(a_t|\mu_t,\sigma_t)}\nabla_{\mathtt{w}}\log\pi_{\mathtt{w}}(a_t|s_t)\,\middle|\,\{s_t,r_t,\mu_t,\sigma_t,a_t,\hat{q}_t\}\sim RM\right]. \tag{21}$$

Here $\mathcal{P}(a_t|\mu_t,\sigma_t)$ is the probability of sampling a_t from a Gaussian distribution with statistics μ_t and σ_t, and $\hat{q}_t$ estimates the cumulative rewards by following the current policy from (s_t,a_t) and is computed with the Retrace algorithm (49):

$$\hat{q}_t = r_{t+1} + \gamma v_{\mathtt{w}}(s_{t+1}) + \gamma\min\left\{1,\frac{\pi_{\mathtt{w}}(a_t|s_t)}{\mathcal{P}(a_t|\mu_t,\sigma_t)}\right\}\left[\hat{q}_{t+1} - v_{\mathtt{w}}(s_{t+1})\right], \tag{22}$$

with $\gamma = 0.995$ the discount factor for rewards into the future. Equation 22 is computed via backward recursion when episodes are entered into the RM (note that $\hat{q}_{T_{end}} \equiv 0$), and iteratively updated as individual steps are sampled. Retrace is also used to derive the gradient for the state-value estimate:

$$g^{val}(\mathtt{w}) = \mathbb{E}\left[\min\left\{1,\frac{\pi_{\mathtt{w}}(a_t|s_t)}{\mathcal{P}(a_t|\mu_t,\sigma_t)}\right\}\left(\hat{q}_t - v_{\mathtt{w}}(s_t)\right)\,\middle|\,\{s_t,r_t,\mu_t,\sigma_t,a_t,\hat{q}_t\}\sim RM\right] \tag{23}$$

The *off-PG* formalizes trial-and-error learning; it moves the policy to make actions with better-than-expected returns ($\hat{q}_t > v_{\mathtt{w}}(s_t)$) more likely, and those with worse outcomes ($\hat{q}_t < v_{\mathtt{w}}(s_t)$) less likely. Both Eq. 21 and Eq. 23 involve expectations over the empirical distribution of experiences contained in the RM, which are approximated by Monte Carlo sampling from the N_{RM} most recent experiences $\hat{g}(\mathtt{w}) = \sum_{i=1}^{B}\hat{g}_i(\mathtt{w})$, where B the mini-batch size. Owing to its use of ER and importance sampling, V-RACER and similar algorithms become unstable if the policy $\pi_{\mathtt{w}}$, and the distribution of states that would be visited by $\pi_{\mathtt{w}}$, diverges from the distribution of experiences in the RM. A practical reason for the instability may be the numerically vanishing or exploding importance weights $\pi_{\mathtt{w}}(a_t|s_t)/\mathcal{P}(a_t|\mu_t,\sigma_t)$. More generally, NN updates computed from interaction data that is no longer relevant to the current policy undermine its optimization. ReF-ER is an extended ER procedure which constrains policy changes and increases the accuracy of the gradient estimates by modifying the update rules of the RL algorithm:

$$\hat{g}_t(\mathtt{w}) \leftarrow \begin{cases}\beta\hat{g}_t(\mathtt{w}) - (1-\beta)\nabla_{\mathtt{w}}D_{KL}\left(\pi_{\mathtt{w}}(\cdot|s_t)\,\|\,\mathcal{P}(\cdot|\mu_t,\sigma_t)\right) & \text{if } \frac{1}{C} < \frac{\pi_{\mathtt{w}}(a_t|s_t)}{\mathcal{P}(a_t|\mu_t,\sigma_t)} < C\\ -(1-\beta)\nabla_{\mathtt{w}}D_{KL}\left(\pi_{\mathtt{w}}(\cdot|s_t)\,\|\,\mathcal{P}(\cdot|\mu_t,\sigma_t)\right) & \text{otherwise}\end{cases} \tag{24}$$

here, $D_{KL}(P\,\|\,Q)$ is the Kullback-Leibler divergence measuring the distance between distributions P and Q. Equation 24 modifies the NN gradient by: 1) Rejecting samples whose importance weight is outside of a trust region determined by $C > 1$. 2) Adding a penalization term to attract $\pi_{\mathtt{w}}(a_t|s_t)$ towards prior policies. The coefficient β is iteratively updated to keep a constant fraction $D \in [0,1]$ of samples in the RM within the trust region:

$$\beta \leftarrow \begin{cases}(1-\eta)\beta & \text{if } n_{far}/N_{RM} > D\\ \beta + (1-\eta)\beta & \text{otherwise}\end{cases} \tag{25}$$

Here n_{far}/N_{RM} is the fraction of the RM with importance weights outside the trust region.

B. Overview of the training set-up. We summarize here the training set-up and hyper-parameters of V-RACER. Each LES is initialized for uniformly sampled $Re_\lambda \in \{65, 76, 88, 103, 120, 140, 163\}$ and a random velocity field synthesized from the target DNS specturum. The residual-stress tensor τ^R is updated with equation 12 and agents' actions every $\Delta_{RL} = \tau_\eta/8$. The LES are interrupted at $T_{end} = 20\tau_1$ (between 750, if $Re_\lambda = 65$, and 1600, if $Re_\lambda = 163$, actions per agent) or if $\|\mathbf{u}\|_\infty > 10^3 u_\eta$, which signals numerical instability. The policy $\pi_{\mathtt{w}}$ is parameterized by a NN with 2 hidden layers of 64 units each, with tanh activations and skip connections. The NN is initialized as (50) with small outer weights and bias shifted such that the initial policy is approximately $\pi_{\mathtt{w}(0)}(\cdot|s) \approx \mathcal{N}(0.04, 10^{-4})$ and produces Smagorinsky coefficients with small perturbations around $C_t \approx 0.2$. Gradients are computed with Monte Carlo estimates with sample size $B = 512$ from a RM of size $N_{RM} = 10^6$. The parameters are updated with the Adam algorithm (51) with learning rate $\eta = 10^{-5}$. As discussed in the main text, because we use conventional RL update rules in a multi-agent setting, single parameter updates are imprecise. We found that ReF-ER with hyper-parameters $C = 2$ (Eq. 24) and $D = 0.05$ (Eq. 25) to stabilize training. Figure 4b shows the two

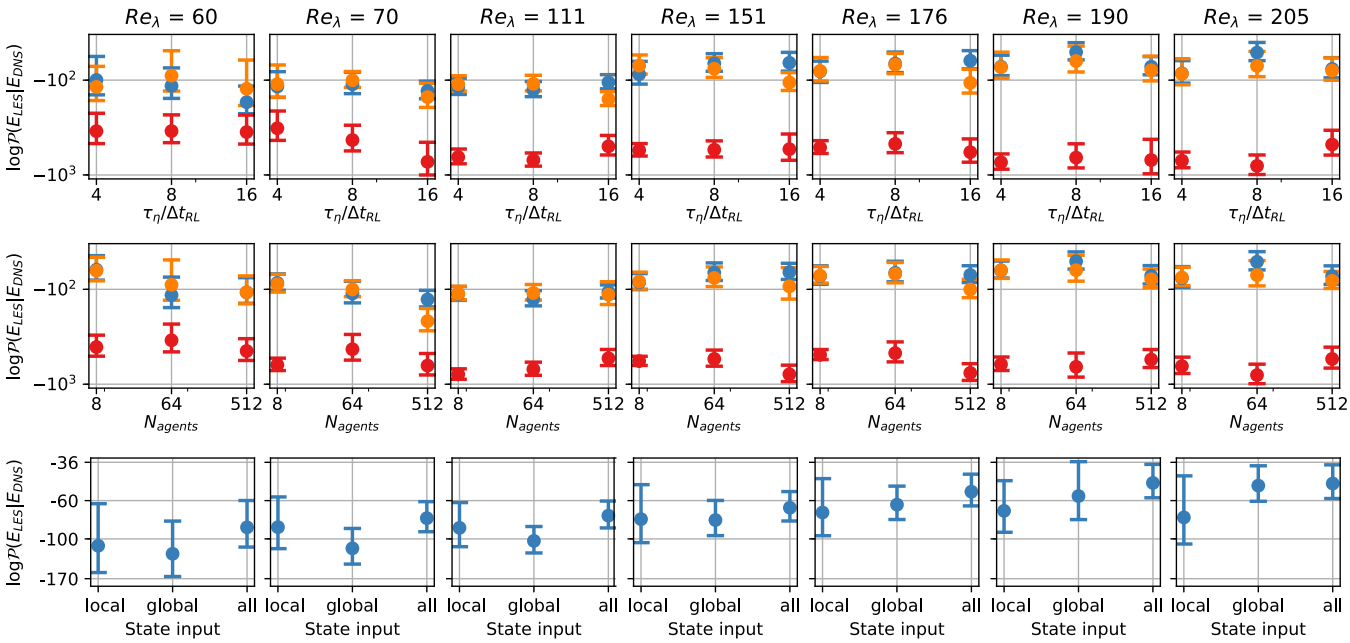


Figure S 5. Time averaged log-likelihood obtained by trained RL policies with varying hyper-parameter settings. In the first row we vary the actuation frequency Δt_{RL}, in the second row we vary the number of agents distributed in the simulation domain, and in the third row we isolate the contribution of local (i.e. invariants of the Hessian and velocity gradient) and global (i.e. energy spectrum and average dissipation rates) information to the overall accuracy of the model. (-●-) RL agent with r^{LL}, (-●-) RL agent with r^{G}, and (-●-) RL agent with r^{LL} employing a RNN policy.

asymptotically extreme settings $N_{agents} = 1$ (i.e. C_s^2 constant in space) and $N_{agents} = 32^3$ (i.e. C_s^2 independently chosen by each grid-point) to perform worse than intermediate ones. Unless otherwise stated, we set $N_{agents} = 4^3$ and analyze this parameter further in the next section. Finally, the reduced description of the system's state mitigates the computational cost and simplifies $\pi_{\mathbf{w}}$. We considered *Recurrent* NN policies, which allow RL to deal with partial observability, but we find them of no use to the present problem (Fig. 4b).

We ran multiple training runs per reward function and whenever we vary the hyper-parameters, but we observe consistent training progress regardless of the initial random seed. The trained policies are evaluated by deterministically setting actions equal to the mean of the Gaussian $a(\mathbf{x}, t) = \mu_{\mathbf{w}}(s(\mathbf{x}, t))$, rather than via sampling.

C. Hyper-parameter analysis. The two most notable hyper-parameters used in our description of the MARL setup are the actuation frequency (determined by Δt_{RL}) and the spatial resolution for the interpolation of the RL actions onto the grid (determined by N_{agents}). Both hyper-parameters serve the purpose of cutting down the amount of experiences collected during each simulation. The alternative would be to use the policy to compute C_s^2 for each grid-point of the domain and update its value on every simulation time-step. This would produce $\mathcal{O}(10^9)$ experiences per simulation and would make the temporal credit-assignment task (i.e. the RL objective of finding causal correlation between single actions and the observed reward) all the more difficult. The default values $\Delta t_{RL} = \tau_\eta/8$ and $N_{agents} = 4^3$ reduce the number of experiences generated per simulation to $\mathcal{O}(10^5)$. In figure 5 we train multiple $\pi_{\mathbf{w}}^{LL}$ policies by varying Δt_{RL} and N_{agents} and we report the time-averaged log-spectrum probability (equation 7) for a set of test Re_λ. We observe the repeated trend of worsening $\log \mathcal{P}$ with either too-frequent actuation or too many dispersed agents ($\Delta t_{RL} = \tau_\eta/16$ and $N_{agents} = 8^3$). On the other hand, SGS models with too coarsely dispersed agents ($N_{agents} = 2^3$) or infrequent actuation update ($\Delta t_{RL} = \tau_\eta/4$) have reduced adaptability and therefore exhibit slightly lower precision. We repeat the same procedure for a RNN-policy, whose only difference relative to the original $\pi_{\mathbf{w}}^{LL}$ model is that the conventional fully-connected layers are replaced by GRU (52). RNN are notoriously harder to train (53) and their performance, while in general it is similar to that of $\pi_{\mathbf{w}}^{LL}$, falls off more noticeably for higher values of Δt_{RL} and N_{agents}.

Finally, we performed an ablation study of the state variables by training SGS models that rely only on local (e.g. velocity gradients and the Hessian) or on global (e.g. the energy spectrum) quantities. We find that models based exclusively on global flow information perform nearly as well as the full model for higher values of Re_λ. In these circumstances the velocity field is severely under-resolved and aliased, therefore the local state is unreliable.