

Kinetic data-driven approach to turbulence subgrid modeling

G. Ortali,^{1,2} A. Gabbana,^{1,3} N. Demo,² G. Rozza,² and F. Toschi^{1,3,4}

¹*Department of Applied Physics and Science Education,
Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands*

²*SISSA (International School for Advanced Studies), Trieste, Italy*

³*Eindhoven Artificial Intelligence Systems Institute,
Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands*

⁴*CNR-IAC, I-00185 Rome, Italy*

Numerical simulations of turbulent flows are well known to pose extreme computational challenges due to the huge number of dynamical degrees of freedom required to correctly describe the complex multi-scale statistical correlations of the velocity. On the other hand, kinetic mesoscale approaches based on the Boltzmann equation, have the potential to describe a broad range of flows, stretching well beyond the special case of gases close to equilibrium, which results in the ordinary Navier-Stokes dynamics. Here we demonstrate that, by properly tuning, a kinetic approach can statistically reproduce the quantitative dynamics of the larger scales in turbulence, thereby providing an alternative, computationally efficient and physically rooted approach towards subgrid scale (SGS) modeling in turbulence. More specifically we show that by leveraging on data from fully resolved Direct Numerical Simulation (DNS) data we can learn a collision operator for the discretized Boltzmann equation solver (the lattice Boltzmann method), which effectively implies a turbulence subgrid closure model. The mesoscopic nature of our formulation makes the learning problem fully local in both space and time, leading to reduced computational costs and enhanced generalization capabilities. We show that the model offers superior performance compared to traditional methods, such as the Smagorinsky model, being less dissipative and, therefore, being able to more closely capture the intermittency of higher-order velocity correlations.

Introduction.— Studying turbulent flows by means of fully resolved numerical simulations is known to pose outstanding computational challenges. The number of dynamical degrees of freedom, whose dynamics needs to be accurately resolved, is typically huge as it rapidly grows as the 9/4 power of the Reynolds number (Re) [1–3]. Despite the ever increasing availability of computing resources, Direct Numerical Simulations (DNS) of turbulent flows are still far out of reach for most real-world applications, strongly motivating a continuous effort towards the development of accurate and computationally cheaper reduced order models. Alongside Reynolds-Averaged Navier-Stokes (RANS) models [4], where the Navier-Stokes equations are averaged over time (separating the flow into mean and fluctuating components) large-eddy simulations (LES) [5] are among the most popular choices. In LES only a portion of the dynamical degrees of freedom, associated to the larger scales, are directly resolved, whereas the effect of the smaller scales on the large scales is parametrized by a subgrid-scale (SGS) model. In general, considering the filtered version of the macroscopic fields of interest, $\bar{\mathbf{u}}$ and \bar{p} (respectively, velocity and pressure), one can write down the Navier-Stokes equations filtered to describe only scales larger than ℓ :

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} + \nabla \cdot (\bar{\mathbf{u}} \bar{\mathbf{u}} + \bar{\mathbf{T}}) = -\nabla \bar{p} + \nu \nabla^2 \bar{\mathbf{u}}. \quad (1)$$

In this filtered equation, we have assumed the density $\rho = 1$, ν is the kinematic viscosity of the fluid and $\bar{\mathbf{T}} = \overline{\mathbf{u}\mathbf{u}} - \bar{\mathbf{u}} \bar{\mathbf{u}}$ is the Reynolds stress tensor, accounting for

the SGS fluctuations, from which one can define a local subgrid energy flux [6]. The problem of defining an SGS closure consists in modelling the unknown term $\bar{\mathbf{T}}$ as a function of only the resolved velocity field $\bar{\mathbf{u}}$. Over the years, several approaches to SGS turbulence modeling have been developed, relying on specific assumptions to approximate the effects of the unresolved scales. The first SGS model was proposed by Smagorinsky [7] and, in view of its simplicity and stability, it is one of the most widespread adopted approaches to estimate the action of smaller scales on the larger scales by means of an effective eddy viscosity, defined on the basis of the local derivatives of the resolved velocity field:

$$\bar{\mathbf{T}}^{\text{smag}} = 2\nu_{\text{eff}}|\bar{\mathbf{S}}| - \bar{p}\mathbf{I}, \quad \nu_{\text{eff}} = \nu + (C_s\Delta)^2|\bar{\mathbf{S}}|, \quad (2)$$

where $|\bar{\mathbf{S}}| = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}}$, $\bar{S}_{ij} = 1/2(\partial_i\bar{u}_j + \partial_j\bar{u}_i)$ represents the magnitude of the filtered strain-rate, Δ is the filter width, and C_s is a non-dimensional coefficient called the Smagorinsky constant. The Smagorinsky model can be shown to be a direct consequence of the Refined Kolmogorov Similarity Hypothesis [8] and has been generalized to non homogeneous flows [9]. Other approaches, such as the Spalart-Allmaras (S-A) model [10] and the $k - \epsilon$ and $k - \omega$ models [11], resolve the eddy viscosity in terms of additional variables, supplementing the equations for mass and momentum with additional equations. Despite their widespread use, these models often face limitations in accurately representing the (statistical) properties of turbulent flows, particularly in scenarios with strong inhomogeneity and anisotropy. Recent advances in Machine Learning have opened up new perspectives

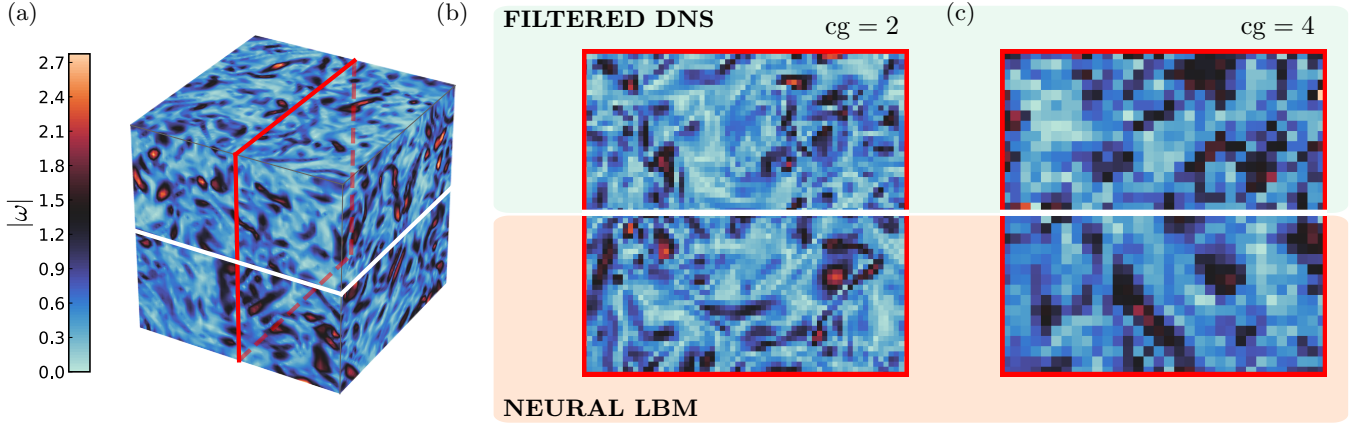


FIG. 1: Snapshots of the vorticity magnitude ($|\omega|$) from 3D simulations of HIT at $Re \approx 6000$. In panel a) the upper half of the domain is taken from filtered DNS data with $cg = 2$, while the lower half is obtained from a simulation using NLBM. We show that at a qualitative level the structures generated by NLBM closely resemble those from filtered DNS. This is further highlighted in panel b) and c) where we show 2D slices of $|\omega|$ (cf. red box in panel a)), respectively at coarse graining factor $cg = 2$ and $cg = 4$.

for employing Artificial Neural Networks (ANN) [12] to enhance computational fluid dynamic solvers [13–15], and for the development of data-driven turbulence modelling [16–18]. Specifically to the LES context, several attempts have been reported in the literature [19–24] to establish SGS closure models from extensive datasets of fully resolved turbulent flows, leveraging on the capability of ANN to handle high-dimensional and statistically complex data. In general, two main approaches have been adopted; the first, which can be regarded as the “black-box” approach, consists in making no assumption on the form of the SGS terms [25–28], while in the second one the task of the ANN consists in tuning the free-parameters of an already established SGS model [29–31].

In this Letter we take a fresh look at the problem of establishing a SGS closure model with ANN, and introduce for the first time, to the best of our knowledge, a data-driven kinetic-based approach to turbulence modeling. We employ Physics-Informed Machine Learning (PIML) to enhance the capabilities of the Lattice Boltzmann Method (LBM) [34], and exploit the extra degrees of freedom provided by the mesoscopic description to learn a new collision operator which effectively acts as SGS model. Our framework relies on the pathway connecting kinetic theory to hydrodynamics, offering the possibility to learn from data macroscopic equations that extend beyond the Navier-Stokes level. Thanks to the locality of the collision operator, at no point this scenario involves the evaluation of spatial derivatives of macroscopic fields [32], drastically simplifying the training process and, in turn, the interpretability of the model. Remarkably, and at variance with respect to other fully local SGS models, our formalism shows the potential for capturing the inverse transfer of energy from small to large scales.

Methods.— The Lattice Boltzmann Method (LBM) [33–35] has emerged in the past decades as a popular solver for computational fluid dynamics. At variance with methods that explicitly discretize the Navies-Stokes equations, LBM operates at the mesoscopic level, providing the description of a fluid system in terms of a (small) set of particle distribution functions (populations) whose dynamic is governed by the discrete Boltzmann equation:

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \Omega_i(\mathbf{x}, t), \quad (3)$$

where at each grid node \mathbf{x} , lattice populations $f_i(\mathbf{x}, t)$ are defined along the discrete components of a velocity stencil $\{\mathbf{c}_i\}, i = 1, \dots, Q$. A popular choice for the collision operator Ω is the single-time relaxation BGK model [36]:

$$\Omega_i(\mathbf{x}, t) = -\frac{\Delta t}{\tau} (f_i(\mathbf{x}, t) - f_i^{\text{eq}}(\mathbf{x}, t)), \quad (4)$$

which models collisions as a relaxation towards an equilibrium distribution, with τ the relaxation rate. The macroscopic variables of interest can be obtained as the lower-order moments of the lattice populations. Moreover, it can be shown via an asymptotic analysis that Eq. 3 yields a second order approximation of the Navier-Stokes equations [35].

It is an expedient for the description of our method to split the time evolution of Eq. 3 into two steps, following the *stream* and *collide* paradigm:

$$f_i^{\text{post}}(\mathbf{x}, t) = f_i^{\text{pre}}(\mathbf{x}, t) + \Omega_i(\mathbf{x}, t), \quad (5)$$

$$f_i^{\text{pre}}(\mathbf{x}, t + \Delta t) = f_i^{\text{post}}(\mathbf{x} - \mathbf{c}_i \Delta t, t), \quad (6)$$

where here and in what follows we will denote with f_i^{pre} (f_i^{post}) the pre(post)-collision populations.

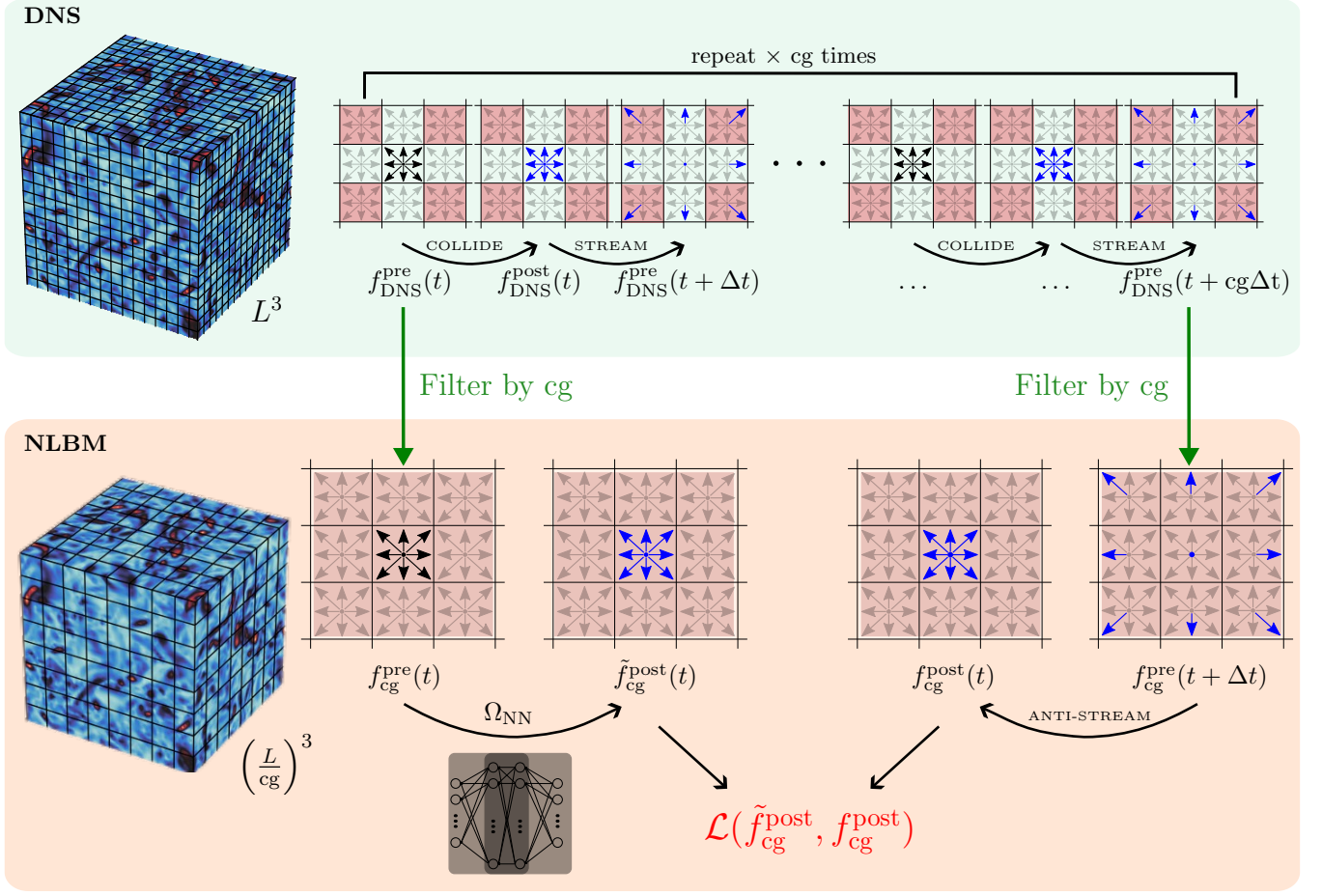


FIG. 2: Schematic representation of the training process for the turbulence SGS model. The upper panel corresponds to the DNS simulation on a L^3 grid, while in lower panel the Neural LBM (NLBM) operates on a $(L/cg)^3$ grid, with coarse-graining factor cg . The mapping between DNS into coarse-grained data is given by the application of a filter: the pre-collision state at a generic time step t at the coarse grained level ($f_{cg}^{pre}(t)$) is obtained by filtering the DNS pre-collision state ($f_{DNS}^{pre}(t)$). Similarly, the post-collision data at the coarse-grained level ($f_{cg}^{post}(t)$) is obtained by first filtering the post-collision DNS state at time step $t + cg\Delta t$, and then by applying the inverse of the streaming operator. Following this procedure it is possible to create an arbitrarily large dataset for training an ANN to which we assign the task of minimizing the mismatch between $\tilde{f}_{cg}^{post}(t) = \Omega_{NN}(f_{cg}^{pre}(t))$ and $f_{cg}^{post}(t)$ under a given error-metric \mathcal{L} .

Following the framework introduced in Ref. [37], we replace the collision operator with a ANN:

$$\tilde{f}_i^{post}(\mathbf{x}, t) = f_i^{pre}(\mathbf{x}, t) + \Omega^{NN}(f_i^{pre}(\mathbf{x}, t)). \quad (7)$$

The resulting algorithm, to which we refer as Neural Lattice Boltzmann Method (NLBM), employs a physics-constrained ANN to establish a data-driven SGS model. Our approach leverages on two key ingredients of the LBM algorithm:

1. The LBM mesoscopic representation makes use of a larger number of degrees of freedom (i.e. the number of discrete lattice populations) than the macroscopic observable of interest. This observation opens up the possibility of using ANN to encode extra information in the model.

2. The non-linear terms encountered at the NS level of description (cf. Eq. 1) are fully embedded in the LBM via the collision operator and thus purely local (in LBM “non-linearity is local, non-locality is linear” [34]). This observation drastically reduces the cost of training ANN, since it offers the possibility to restrict the input and output of the network to local quantities without explicit evaluation of gradients of macroscopic fields.

We summarize here the main steps required to train the ANN, as sketched in Fig. 2, leaving full technical details to the Supplementary Material. We consider simulations of Homogeneous Isotropic Turbulence (HIT), fully resolved on a L^3 domain using the standard LBM formulation (Eq. 3) with BGK collision operator (Eq. 4). We

regard this as the target ground truth DNS data (upper panel in Fig. 2). Next, we define a coarse-graining factor cg , and define a filter which projects, at a given time step t , data from the DNS grid to a coarse-grained grid of size $(L/cg)^3$, where the SGS model will operate (lower panel in Fig. 2). In order to define an arbitrarily large training dataset consisting of pre and post collision data at the coarse-grained level, respectively $f_{cg}^{pre}(t)$ and $f_{cg}^{post}(t)$, we take the following steps: We start from the pre-collision populations at the DNS level, $f_{DNS}^{pre}(t)$, and immediately apply the filter in order to obtain $f_{cg}^{pre}(t)$. Next, we advance the DNS simulation for $cg\Delta t$ time-steps, yielding the pre-collision (and post-streaming) populations $f_{DNS}^{pre}(t + cg\Delta t)$. Since Δt steps at the coarse-grained level correspond to $cg\Delta t$ steps at the DNS level, by filtering $f_{DNS}^{pre}(t + cg\Delta t)$ we obtain $f_{cg}^{pre}(t + \Delta t)$. Finally, we obtain $f_{cg}^{post}(t)$ by reversing the streaming operation, i.e. we anti-stream populations on the coarse grid with respect to the corresponding velocity component $-c_i$.

At this stage, we can train a ANN which, under a given error-metric \mathcal{L} , minimizes the mismatch between $f_{cg}^{post}(t)$ and the prediction of the network taking $f_{cg}^{pre}(t)$ as input:

$$\mathcal{L}(\Omega^{NN}(f_{cg}^{pre}(t)), f_{cg}^{post}(t)). \quad (8)$$

We have observed that the two main ingredients allowing for training models which deliver accurate and stable results throughout an entire simulation are i) imposing hard-constraints on the preservation of mass and momentum [37] and ii) using *unrolled training* [38] to compute the loss over consecutive timesteps. Full details on the ANN architecture and the training process are provided as Supplementary Information.

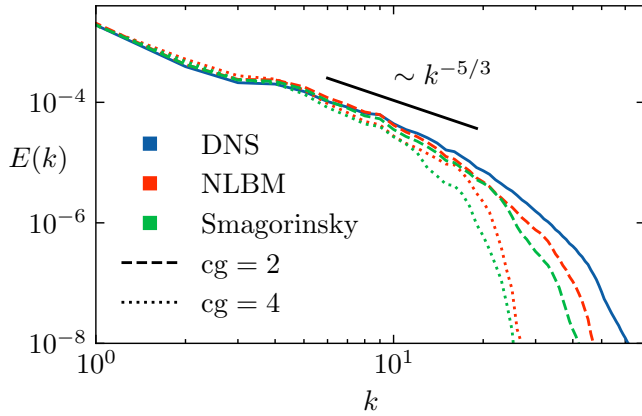


FIG. 3: Energy spectrum for simulations of HIT at $Re \approx 6000$. The results from DNS (blue curve) are compared with NLBM (red) and Smagorinsky (green).

For the two SGS models we report the average spectrum from 100 simulations starting from different initial conditions. The shaded curves corresponds to one standard deviation from the average value.

Numerical results.— We consider numerical simula-

tions of HIT, with for reference a DNS on a $L = 128^3$ grid at Reynolds number $Re \approx 6000$. The parameters are selected in such a way that the plain LBM algorithm would encounter numerical instabilities when working on coarse-grained grids with $cg = \{2, 4\}$. Fig. 1 qualitatively summarizes our findings. Starting from the same initial configuration, we present 3D and 2D representation of the absolute value of the vorticity $|\omega| = |\nabla \times \mathbf{u}|$ at a late stage of the simulation, comparing results from the filtered DNS with NLBM. Our model provides stable simulations, with flow patterns virtually indistinguishable from those of the filtered DNS.

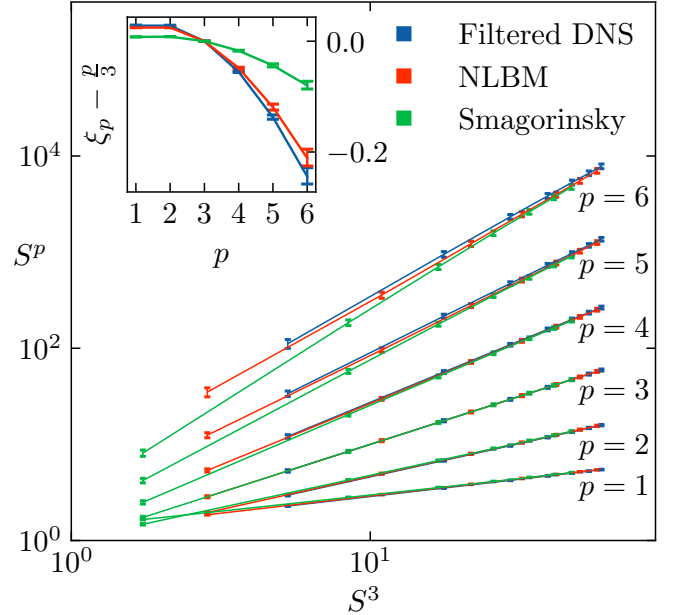


FIG. 4: Structure functions (cf. Eq. 9) of order p , ranging between $p = 1$ to $p = 6$, versus S^3 , with in blue data from filtered DNS with $cg = 4$, in red data from simulations using NLBM, and in green data from simulations using the Smagorinsky model. The inset shows the deviation for the scaling exponents ξ_p from the K41 scaling $\frac{p}{3}$.

On a more quantitative ground, we now turn to the evaluation of the statistical properties of the turbulent flows produced by NLBM simulations. For comparison, we hereafter also take into consideration LBM simulations equipped with the Smagorinsky SGS (Eq. 2), which allows for stable simulations of the coarse domains. In Fig. 3 we present the kinetic energy spectrum $E(k)$. We observe that NLBM well compares with the results produced by Smagorinsky, with slightly superior results at $cg = 4$, where we can observe a spectrum less dissipative than the Smagorinsky one, and closer to the scaling of DNS data. Next, we study the scaling behaviour of high-order Eulerian Structure function, which for a generic

order p can be computed from numerical data as

$$S^p(l) = \left\langle \left[(\mathbf{u}(\mathbf{x} + \mathbf{l}) - \mathbf{u}(\mathbf{x})) \cdot \hat{\mathbf{l}} \right]^p \right\rangle. \quad (9)$$

We use the extended self similarity (ESS) [39] to determine the scaling exponents ξ_p . In Fig. 4 we plot the structure functions of order p , ranging between $p = 1$ to $p = 6$, versus $S^3(l)$, comparing data from filtered DNS with $\text{cg} = 4$, and data from simulations using NLBM and the Smagorinsky model. In the inset of Fig. 4 we report the deviation of ξ_p from the K41 scaling $p/3$. The results highlight that, due to its lower dissipation at small scales, NLBM provides anomalous scaling within error bars from filtered DNS. On the other hand, the discrepancies observed in the Smagorinsky model are due to the fact that under these parameters the inertial range is shrunk (see the supplementary material for a more detailed analysis).

A major advantage of our kinetic-based approach lies in the possibility of having a physical interpretation of the action of the ANN, by projecting the lattice populations to the velocity moment space (see Supplementary Material). We have observed that NLBM extends the single-relaxation BGK collision operator, from where DNS data was generated, to a multi-relaxation collision operator [40, 41]. The model introduces a non-linear relation for pre and post collision values of the moments related to the bulk-viscosity, something which is often used to increase the stability of numerical simulations. The model preserves a linear-dependency for the pre and post collision values of the moments related to the kinetic viscosity. This allows to fit the effective viscosity from the numerical data, which can be used to establish a direct comparison with the Smagorinsky model (Eq. 2). In Fig. 5 we compare the probability distribution function (PDF) of the value of the Smagorinsky constant C^2 fitted from NLBM data. The average value of the Smagorinsky constant for NLBM ($\langle C_{\text{NLBM}}^2 \rangle \approx 0.11$) is about a factor two smaller than the one used in simulations with the Smagorinsky SGS model ($C^2 = 0.2$). Remarkably, in Fig. 5 we observe a tail taking negative values, suggesting that our model occasionally displays an inverse transfer of energy from small to large scales, a feature completely lacking to the Smagorinsky model which is, by its own nature, fully dissipative.

Conclusion.— Summarizing, we have introduced a novel kinetic-based approach to SGS modeling, combining LBM with physics informed ANN. Our model allows for stable simulations on coarse domains, offering the possibility of reducing the computational costs of DNS, in turn preserving the statistical properties of turbulent flows under HIT settings. The model compares well with Smagorinsky, and our results highlight a better agreement with DNS in terms of energy spectra and estimation of the anomalous scaling exponents. Moreover, we have shown that the model in principle supports the possibility to describe the inverse transfer of energy from small

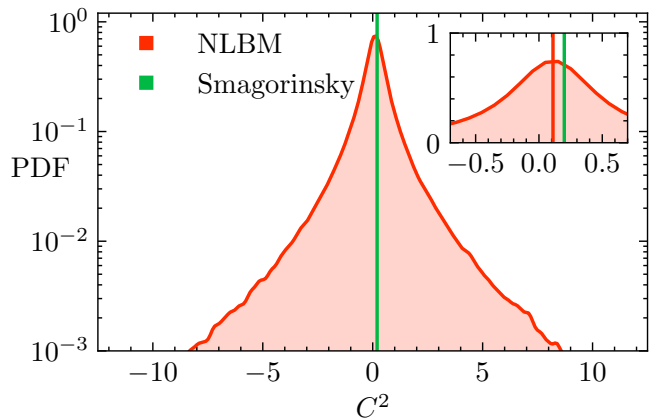


FIG. 5: Probability distribution function (PDF) of the fitted value of the Smagorinsky constant C^2 from NLBM data. The inset highlights that the average value $\langle C_{\text{NLBM}}^2 \rangle \approx 0.11$ is about a factor two smaller than the one used in simulations with the Smagorinsky SGS model ($C^2 = 0.2$). The presence of a tail with negative values highlights the fact that in NLBM it is possible to capture the inverse transfer of energy from small to large scales.

to large scales, which we regard as promising particularly in vision of future application to more involved numerical setups. To conclude, this work opens up the possibility of exploiting the extra degrees of freedom of the mesoscopic representation in the LBM to develop novel SGS models. In future works, we plan to extend our analysis to flows subject to anisotropy as well as wall-bounded flows. A further intriguing question concerns the application of this framework beyond SGS. Work to explore employing our kinetic data-driven approach in other contexts is currently underway.

Acknowledgments.— We wish to thank Roberto Benzi for useful discussions. This work was partially funded by the Dutch Research Council (NWO) through the UNRAVEL project (with Project No. OCENW.GROOT.2019.044).

-
- [1] U. Frisch, *Turbulence: The Legacy of A.N. Kolmogorov*, (Cambridge University Press, 1995).
 - [2] S. B. Pope, *Turbulent Flows*, (Cambridge University Press, 2000).
 - [3] R. Benzi and F. Toschi, *Physics Reports* **1021**, 1–106 (2023).
 - [4] G. Alfonsi, *Applied Mechanics Reviews* **62**, 040802 (2009).
 - [5] Y. Zhiyin, *Chinese Journal of Aeronautics* **28**, 11–24 (2015).
 - [6] V. Borue and S. A. Orszag, *Journal of Fluid Mechanics* **366**, 1–31 (1998).
 - [7] J. Smagorinsky, *Monthly weather review* **91**, 99–164

- (1963).
- [8] F. Toschi, E. L  v  que, and G. Ruiz-Chavarria, *Phys. Rev. Lett.* **85**, 1436–1439 (2000).
 - [9] E. L  v  que, F. Toschi, L. Shao, and J. P. Bertoglio, *Journal of Fluid Mechanics* **570**, 491–502 (2007).
 - [10] P. Spalart and S. Allmaras *In 30th aerospace sciences meeting and exhibit, AIAA Paper 1992-0439* (1992).
 - [11] D. C. Wilcox, *Turbulence Modeling for CFD* (DCW Industries, 2006).
 - [12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).
 - [13] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, *Proceedings of the National Academy of Sciences* **118**, e2101784118 (2021).
 - [14] R. Vinuesa and S. L. Brunton, *Nature Computational Science* **2**, 358–366 (2022).
 - [15] M. Lino, S. Fotiadis, A. A. Bharath, and C. D. Cantwell, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **479**, 20230058 (2023).
 - [16] J. Ling, A. Kurzwski, and J. Templeton, *Journal of Fluid Mechanics* **807**, 155–166 (2016).
 - [17] K. Duraisamy, G. Iaccarino, and H. Xiao, *Annual review of fluid mechanics* **51**, 357–377 (2019).
 - [18] R. Wang, K. Kashinath, M. Mustafa, A. Albert, and R. Yu, *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1457–1466 (2020).
 - [19] Z. Wang, K. Luo, D. Li, J. Tan, and J. Fan, *Physics of Fluids* **30**, 125101 (2018).
 - [20] R. Maulik, O. San, A. Rasheed, and P. Vedula, *Journal of Fluid Mechanics* **858**, 122–144 (2019).
 - [21] C. Xie, J. Wang, and W. E, *Phys. Rev. Fluids* **5**, 054606 (2020).
 - [22] J. Park and H. Choi, *Journal of Fluid Mechanics* **914**, A16 (2021).
 - [23] H. Frezat, G. Balarac, J. Le Sommer, R. Fablet, and R. Lguensat, *Phys. Rev. Fluids* **6**, 024607 (2021).
 - [24] Y. Tian, M. Woodward, M. Stepanov, C. Fryer, C. Hyett, D. Livescu, and M. Chertkov, *Proceedings of the National Academy of Sciences* **120**, e2213638120 (2023).
 - [25] A. Beck, D. Flad, and C.-D. Munz, *Journal of Computational Physics* **398**, 108910 (2019).
 - [26] Z. Zhou, G. He, S. Wang, and G. Jin, *Computers & Fluids* **195**, 104319 (2019).
 - [27] C. Xie, J. Wang, K. Li, and C. Ma, *Phys. Rev. E* **99**, 053113 (2019).
 - [28] M. Kurz, P. Offenh  user, and A. Beck, *International Journal of Heat and Fluid Flow* **99**, 109094 (2023).
 - [29] F. Sarghini, G. de Felice, and S. Santini, *Computers & Fluids* **32**, 97–108 (2003).
 - [30] C. Xie, J. Wang, H. Li, M. Wan, and S. Chen, *Physics of Fluids* **31**, 085112 (2019).
 - [31] Y. Wang, Z. Yuan, C. Xie, and J. Wang, *AIP Advances* **11**, (2021).
 - [32] H. Chen, S. Kandasamy, S. Orszag, R. Shock, S. Succi, and V. Yakhot, *Science* **301**, 633–636 (2003).
 - [33] R. Benzi, S. Succi, and M. Vergassola, *Physics Reports* **222**, 145–197 (1992).
 - [34] S. Succi, *The Lattice Boltzmann Equation: For Complex States of Flowing Matter* (Oxford Scholarship Online, Oxford, 2018).
 - [35] T. Krueger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, E. Viggen, *The Lattice Boltzmann Method: Principles and Practice* (Springer, 2016).
 - [36] P.L. Bhatnagar, E.P. Gross, M.Krook, *Phys. Rev.* **94**, 511 (1954).
 - [37] A. Corbetta, A. Gabbana, V. Gyrya, D. Livescu, J. Prins, and F. Toschi, *The European Physical Journal E* **46**, 10 (2023).
 - [38] J. Brandstetter, D. E. Worrall, and M. Welling, *arXiv:2202.03376*, (2022).
 - [39] R. Benzi, S. Ciliberto, R. Tripiccone, C. Baudet, F. Massaioli, and S. Succi, *Phys. Rev. E* **48**, R29–R32 (1993).
 - [40] P. Lallemand and L.-S. Luo, *Phys. Rev. E* **61**, 6546–6562 (2000).
 - [41] D. D’Humi  res, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo, *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* **360**, 437–51 (2002).

Supplementary Information for “Kinetic data-driven approach to turbulence subgrid modeling”

NUMERICAL METHOD

In this section, we provide a basic introduction to the Lattice Boltzmann Method (LBM). The reader is referred to Ref. [S1, S2] for a thorough introduction to the topic. The LBM is a class of numerical fluid-dynamics solvers. At variance with conventional methods that explicitly discretize the Navies-Stokes equations, LBM takes root from the mesoscopic Boltzmann equation:

$$\frac{\partial f}{\partial t} + \xi_\alpha \frac{\partial f}{\partial x_\alpha} + F_\alpha \frac{\partial f}{\partial \xi_\alpha} = \Omega(f) \quad (\text{S1})$$

where $f(\mathbf{x}, \boldsymbol{\xi}, t)$ is the particle distribution function, representing the average number of particles in a small element of phase-space centered at position \mathbf{x} with velocity $\boldsymbol{\xi}$ at time t , and F_α is the sum of external forces acting on the system. The right hand side of the equation, $\Omega(f)$, is the collision operator, describing the changes in f due to particle collisions.

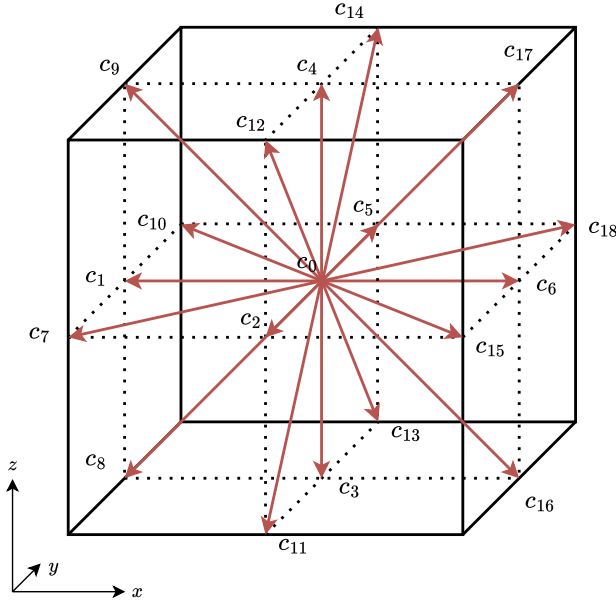


FIG. S1: D3Q19 Velocity stencil

Index	\mathbf{c}_i	w_i
0	(0, 0, 0)	1/3
1	(-1, 0, 0)	1/18
2	(0, -1, 0)	1/18
3	(0, 0, -1)	1/18
4	(0, 0, 1)	1/18
5	(0, 1, 0)	1/18
6	(1, 0, 0)	1/18
7	(-1, -1, 0)	1/36
8	(-1, 0, -1)	1/36
9	(-1, 0, 1)	1/36
10	(-1, 1, 0)	1/36
11	(0, -1, -1)	1/36
12	(0, -1, 1)	1/36
13	(0, 1, -1)	1/36
14	(0, 1, 1)	1/36
15	(1, -1, 0)	1/36
16	(1, 0, -1)	1/36
17	(1, 0, 1)	1/36
18	(1, 1, 0)	1/36

TABLE S1: D3Q19 Velocity stencil - discrete velocity vectors \mathbf{c}_i and corresponding weights w_i .

The lattice time-discrete counterpart of Eq. S1 reads as

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \Omega_i(\mathbf{x}, t) + f_i^{\text{ext}}, \quad (\text{S2})$$

In the above equation, known as lattice Boltzmann equation, the discrete particle distribution functions (populations) $f_i(\mathbf{x}, t)$ move at the next time step $t + \Delta t$ with velocity \mathbf{c}_i to a neighboring grid cell $\mathbf{x} + \mathbf{c}_i \Delta t$. Moreover, f_i^{ext} is the discrete version of the external force term. The discretization of the velocity space is a key element in the definition of LBM; it consists in replacing the continuum velocity space with a small set of discrete velocities $\mathcal{V} = \{\mathbf{c}_i \in \mathbb{R}^3\}$ corresponding to the abscissa of a Gauss-Hermite quadrature (with corresponding weights w_i) that ensure that all moments of the distribution function up to a prescribed order are correctly recovered [S3, S4]. In this work, we use the D3Q19 velocity stencil (Fig. S1), defined in Tab. S1, which allows to express mass and momentum as weighted sums of $f_i(\mathbf{x}, t)$:

$$\rho = \sum_i f_i, \quad \rho \mathbf{u} = \sum_i f_i \mathbf{c}_i. \quad (\text{S3})$$

There are several possible choices to approximate the collision operator Ω_i . Arguably the simplest one that can be used for Navier-Stokes simulations is the Bhatnagar-Gross-Krook (BGK) [S5] operator:

$$\Omega_i(f) = \frac{\Delta t}{\tau}(f_i^{\text{eq}} - f_i), \quad (\text{S4})$$

which relaxes the populations at rate τ towards a local equilibrium defined by the Maxwell-Boltzmann distribution, for which we use a second order polynomial expansion:

$$f_i^{\text{eq}}(\rho, \mathbf{u}) = w_i \rho \left(1 + \frac{\mathbf{u} \cdot \mathbf{c}_i}{c_s^2} + \frac{(\mathbf{u} \cdot \mathbf{c}_i)^2 - (c_s |\mathbf{u}|)^2}{2c_s^4} \right),$$

with $c_s = 1/\sqrt{3}$ the sound speed in the lattice, and w_i a lattice-dependent set of weighting factors (see Tab. S1).

Several options are available for implementing the body forces in LBM [S1]. In this work we adopt the Exact Difference Method (EDM) scheme [S6], where the external forcing term is computed as

$$f_i^{\text{ext}} = f_i^{\text{eq,shift}} - f_i^{\text{eq}}, \quad (\text{S5})$$

where the shifted equilibrium $f_i^{\text{eq,shift}}$ is computed from Eq. S5 based on the shifted velocity $\mathbf{u}^{\text{shift}} = \mathbf{u} + \frac{\mathbf{F}}{\rho}$.

Following an asymptotic analysis, such as the Chapman-Enskog expansion [S7], it can be shown that Eq. S2 combined with Eq. S4 delivers in the low-Mach number limit a second order approximation of the Navier-Stokes equations, with the following expression putting in relationship the relaxation time parameter τ with the kinematic viscosity ν of the fluid:

$$\nu = \left(\tau - \frac{\Delta t}{2} \right) c_s^2. \quad (\text{S6})$$

Implementing the Smagorinsky Turbulence Model in LBM

In the main text we have reported numerical results of LBM simulations equipped with the Smagorinsky SGS model. The model involves adding an eddy viscosity term to the governing equations to account for unresolved turbulent fluctuations at the grid scale. This eddy viscosity is computed as:

$$\nu_{\text{eff}} = \nu + (C_s \Delta)^2 |\bar{\mathbf{S}}|, \quad (\text{S7})$$

where $|\bar{\mathbf{S}}| = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}}$, $\bar{S}_{ij} = 1/2(\partial_i \bar{u}_j + \partial_j \bar{u}_i)$ is the magnitude of the filtered strain-rate, Δ is the filter width, and C_s is a non-dimensional coefficient called the Smagorinsky constant.

In order to incorporate the Smagorinsky model in a LBM scheme one needs to i) compute the strain rate tensor in each cell to then ii) compute the eddy viscosity ν_{eff} to finally iii) locally adjust the relaxation time to match ν_{eff} . In LBM the strain rate tensor can be computed locally from the non-equilibrium part of the distribution function:

$$\bar{S}_{ij} = -\frac{3\tau}{2} \Pi_{ij}^{\text{neq}}, \quad \Pi_{ij}^{\text{neq}} = \sum_q c_{qi} c_{qj} (f_q - f_q^{\text{eq}}). \quad (\text{S8})$$

By combining Eq. S6 with Eq. S7 it is then possible to compute the local relaxation time due to the Smagorinsky model:

$$\tau_{\text{eff}} = \frac{\nu_{\text{eff}}}{c_s^2} + \frac{\Delta t}{2}. \quad (\text{S9})$$

ARTIFICIAL NEURAL NETWORK ARCHITECTURE AND TRAINING PROCESS

Definition of the Training Dataset

The ground truth data used to train the ANN consists of N pairs of 19-tuples

$$\left\{ (f_{i,k}^{\text{pre}}, f_{i,k}^{\text{post}}), k = 1, 2, \dots, N \right\}. \quad (\text{S10})$$

The data is obtained from Direct Numerical Simulation (DNS) of Homogeneous Isotropic Turbulence. We conduct LBM simulations in a three-dimensional cubic domain of side $L = 128$, with periodic boundary conditions and $\Delta t = 1$. Simulations are driven by a stationary external force, ensuring a divergence-free flow, defined for the x , y and z components as:

$$\begin{cases} F_x(y) = F \sin\left(\frac{2\pi}{L}y\right) \\ F_y(z) = F \sin\left(\frac{2\pi}{L}z\right) \\ F_z(x) = F \sin\left(\frac{2\pi}{L}x\right) \end{cases} . \quad (\text{S11})$$

We work at a value of the Reynolds number $\text{Re} \approx 6000$, achieved tuning the relaxation time τ and the magnitude of the external force as detailed in Tab. S2. We collect data after an initial transient phase of T_0 timesteps, ensuring that the dynamics reaches a statistically stationary state. The full state of the system (in terms of lattice populations) is sampled every δ_T timesteps, for a total of N_T timesteps. Next, we define a coarse graining factor cg and a filter ϕ , which allows mapping at a generic time step t lattice populations from the L^3 DNS grid to the $(L/\text{cg})^3$ coarse-grained grid:

$$f_i^{\text{cg}}(t) \leftarrow \phi(f_i^{\text{DNS}}(t)). \quad (\text{S12})$$

Several possible choices can be made for the filter ϕ , which may substantially impact the quality of the results. In this work we consider a sub-sampling strategy, and define the coarse-grained grid by sampling one every cg lattice points along the different coordinates. We leave as a future work a careful evaluation and comparison of different choices for ϕ .

At this stage, we can establish the training set (Eq. S10) at the coarse grained level, following the procedure sketched in Fig. 2 of the main text, which we here summarize:

1. We start from the pre-collision populations at the DNS level, $f_{\text{DNS}}^{\text{pre}}(t)$, and apply a filter in order to obtain $f_{\text{cg}}^{\text{pre}}(t)$.
2. We advance the DNS simulation for $\text{cg}\Delta t$ time-steps obtaining the post-streaming state, i.e. $f_{\text{DNS}}^{\text{pre}}(t + \text{cg}\Delta t)$.
3. By filtering the DNS level we obtain $f_{\text{cg}}^{\text{pre}}(t + \Delta t)$.
4. In order to obtain the pre-streaming state, i.e. the post-collision values at time t , we apply the inverse of the streaming operation at the coarse grained level by reversing the streaming direction, i.e. with respect to $-\mathbf{c}_i$.

For the results reported in the main text we have used $\text{cg} = 2$ and $\text{cg} = 4$, resulting in effective lattice sizes of, respectively, $L' = 64$ and $L' = 32$. Note that to achieve the same Re on the coarse grids one could use the same numerical parameters provided in Tab. S2, alongside a rescaling of the relaxation time parameter

$$\tau' = \frac{1}{\text{cg}} \left(\tau - \frac{1}{2} \right) + \frac{1}{2} . \quad (\text{S13})$$

However, we shall remark that the choice of the parameters is such that plain LBM simulations at the coarse grid level would lead to numerical instabilities.

Neural Network Architecture and Optimization

In this section we detail our ANN model, which consists in learning a correction term adding to the BGK collision operator. We define the post-collision state as

$$\tilde{f}_i^{\text{post}} = f_i^{\text{pre}} + \Omega(f_i^{\text{pre}}) + \frac{1}{\tau} \Omega^{\text{NN}}(f_i^{\text{pre}}) \quad (\text{S14})$$

where Ω is the BGK collision operator in Eq. S4, while Ω^{NN} is the correction term due to the ANN.

We consider an ANN consisting of a Multi-layer Perceptron (MLP) with input and output layers of size 19 (equal to the number of lattice populations sitting on one single grid point), and three fully connected hidden layers of size [300, 300, 100], in combination with LeakyReLU activation function. Following Ref. [S8], before the output layer we

Parameter	Description	Value
Re	Reynolds Number	6×10^3
τ	Relaxation time Parameter	0.5032
F	Force term Amplitude	5×10^{-6}
L	Lattice Size for DNS	128
Δ_x	Grid size	1
Δ_t	Timestep	1
T_0	Transient Phase Duration	5×10^4
δ_T	Sampling Interval	10^3
N_T	Total Number of Timesteps	10^3
M	Lattice Points Sampled per Timestep	512
λ	Learning rate	1×10^{-4}
ω	Weight decay rate	1×10^{-6}

TABLE S2: The table reports the numerical parameters used in DNS numerical simulations (top), for the definition of the training dataset (middle) and for the hyper-parameters used by the ADAM optimizer to train the ANN.

also include a non-trainable layer posing hard constraints on the conservation of mass and momentum. In our case we require that the correction term given by the ANN does not introduce additional mass and momentum in the output:

$$\sum_i \Omega^{\text{NN}}(f_i^{\text{pre}}) = 0, \quad (\text{S15})$$

$$\sum_i \Omega^{\text{NN}}(f_i^{\text{pre}}) \mathbf{c}_i = \mathbf{0}. \quad (\text{S16})$$

The above equations are satisfied using

$$\hat{f}_i = \Omega^{\text{NN}}(f_i^{\text{pre}}) + \kappa_1 + \kappa_2 c_{i,x} + \kappa_3 c_{i,y} + \kappa_4 c_{i,z}, \quad (\text{S17})$$

with the following coefficients specific to the D3Q19 model:

$$\begin{aligned} \kappa_1 &= -\frac{1}{19} \sum_i \Omega^{\text{NN}}(f_i^{\text{pre}}), \\ \kappa_2 &= -\frac{1}{10} \sum_i \Omega^{\text{NN}}(f_i^{\text{pre}}) c_{i,x}, \\ \kappa_3 &= -\frac{1}{10} \sum_i \Omega^{\text{NN}}(f_i^{\text{pre}}) c_{i,y}, \\ \kappa_4 &= -\frac{1}{10} \sum_i \Omega^{\text{NN}}(f_i^{\text{pre}}) c_{i,z}. \end{aligned}$$

We have observed that the model defined in Eq. S14, albeit performing well in one-step predictions, once plugged in a simulation typically leads to numerical instabilities after just few iterations. This behavior is commonly observed in autoregressive models, i.e. models that feed in the output as an input for the next step [S9]. This is due to accumulations of errors that eventually deviate the dynamics of the system towards configurations not covered in the training dataset, a phenomenon known in literature as *distribution shift* [S10]. In the next section we discuss a procedure which allows to mitigate this problem.

Unrolled training

In this section we discuss the *Unrolled training* technique, which we employ to enhance the stability of the model. In an auto-regressive model, i.e. a model that recursively uses its output as an input for the successive timesteps, one must ensure the stability of the model with respect to the accumulation of errors. This is a requirement, for example, for any time integration scheme, where the stability hinges on both the system's dynamics and the discretization scheme employed. Moreover, in the domain of Machine Learning, another phenomenon may arise, since the accumulation of

errors may lead the dynamics of the system into unexplored regions of the optimization space, deviating from the training conditions. This problem is typically framed in the context of *distribution shift*, and deals with scenarios in which the data distribution during training differs from that encountered during testing. Fields such as language modeling and robotics commonly encounter such challenges. For instance, maintaining coherence in generated text over multiple iterations is a complex task for language models [S11].

Several solutions have been proposed to deal with this problem. One possibility is to modify the architecture of the Neural Network, for example forcing the spectra of the linear matrices, stabilizing the dynamics. Examples of this approach are Antisymmetric Neural Networks [S12]. Another possibility is to use regularization strategies such as noise injection in the input and/or the output of the Neural Network [S13].

In this work, we employ a technique known in literature as *unrolled training* or *trajectory learning* [S9]. In short, it consists in applying the model to make predictions over L_T consecutive timesteps at training time, backpropagating the gradients through some or all such timesteps. This is similar to the *Backpropagation Through Time* method, used for training Recurrent Neural Networks, with the main difference being that in *unrolled training* no memory term is present in the Neural Network.

We train the model using the following error metric:

$$\mathcal{L} = \sum_{x_i \in S} \left(\tilde{f}^{\text{post}}(x_i, t_0 + L_t) - f^{\text{post}}(x_i, t_0 + L_t) \right)^2, \quad (\text{S18})$$

where for each training epoch S represents a randomly selected subset of the lattice points. One single training epoch consists of the following steps:

1. We select a snapshot from the ground truth dataset, i.e. the pre-collision populations on the entire filtered grid of size $(L/\text{cg})^3$ at a fixed time step. This serves as the initial condition.
2. We randomly select a subset of lattice points, denoted as $S = \{x_i\}_{i=0}^{\text{batch-size}}$, which will be used compute the loss.
3. We evolve the system for L_T timesteps, alternating the collision step via Eq. S14, and the streaming step.
4. We compute the loss via Eq. S18 comparing against ground-truth values at time $t + L_T$.
5. Finally, we backpropagate the loss through all the timesteps to compute the gradients and update the weights of the Neural Network.

We remark that, while the full domain is evolved using the ANN, the loss term is computed only on randomly selected subset of points. The reason for this is twofold. First, it allows reducing computational costs, both in terms of memory requirements as well as overall training time. Second, it allows to increase stochasticity in the computation of gradients for updating the weights of the network.

This strategy introduces a few extra hyperparameters in the training of the network. The results reported in the main text have been obtained using a batch size $|S|$ of 512 elements, with a trajectory length of $L_T = 12$. The choice of these parameters impacts the time and memory requirements of the training process (longer trajectories and bigger batch sizes require more time and memory), and on the resulting stability of the model (too short trajectories give less stable models). The optimization is done using Adam Optimizer, training for a total of 2×10^4 epochs with early stopping based on test loss. The optimization parameters are reported in Table S2.

STRUCTURE FUNCTIONS AND EXTENDED SELF SIMILARITY

In Fig. 4 in the main text we have reported results with of high order statistical properties of turbulence, analyzing data from simulations using our data-driven SGS model. In this section we provide further details; for the convenience of the reader we state once again the definition of the Eulerian Structure function of order p :

$$S^p(l) = \left\langle \left[(\mathbf{u}(\mathbf{x} + \mathbf{l}) - \mathbf{u}(\mathbf{x})) \cdot \hat{\mathbf{l}} \right]^p \right\rangle. \quad (\text{S19})$$

In Fig. S2, the first row refers to results for a grid with coarse graining factor $\text{cg} = 2$, whereas the second row shows results for $\text{cg} = 4$. For all cases we show in blue color filtered DNS data, in red our NLBM model, and for comparison in green results using Smagorinsky. In the first column we plot the structure functions S^p vs l , with p

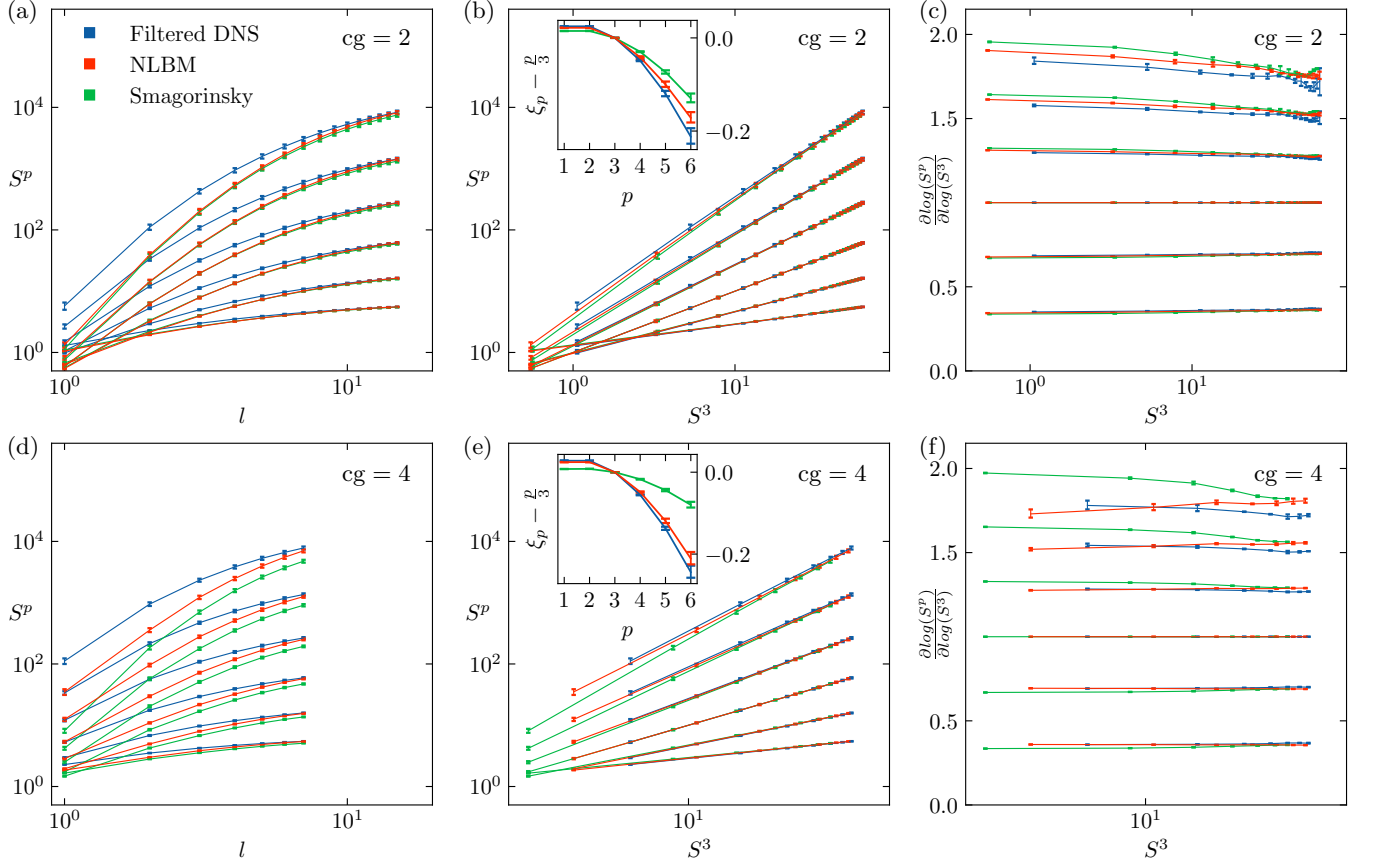


FIG. S2: Results for structure functions (cf. Eq. 9) of order p , ranging between $p = 1$ to $p = 6$, with in blue data from filtered DNS, in red data from simulations using NLBM, and in green data from simulations using the Smagorinsky model. On the top row (panels (a)-(c)) results for $cg = 2$, on the bottom row (panels (d)-(f)) results for $cg = 4$. In the first column we report the Structure function $S^p(l)$ versus l , in the second column S^p versus S^3 , and in the third column the logarithmic derivative $\frac{\partial \log(S^p)}{\partial \log(S^3)}$.

ranging between 1 to 6. The central column shows the results of the extended self similarity analysis, where this time we plot the structure functions S^p versus S^3 . The insets show the deviation of the scaling exponents ξ_p from K41 scaling $p/3$. The scaling exponents have been computed fitting the slopes, which are better highlighted in the third column, where we show the logarithmic derivative $\frac{\partial \log(S^p)}{\partial \log(S^3)}$. The fitting ranges are defined considering intervals where the logarithmic derivative stays approximately constant, avoiding the dissipative interval by excluding the first point, resulting in S^3 values of $[4, 20]$ for $cg = 2$ and $[8, 40]$ for $cg = 4$.

The results here presented strengthen the message and the considerations made in the main text, showcasing that NLBM better captures the kinetic energy at the large scales with respect than Smagorinsky, which exhibits instead larger dissipation.

GENERALIZATION

A primary concern in data-driven modelling is over-fitting and poor generalization [S12], connected to the challenge of generating a comprehensive, high-quality training dataset that can cover all potential uses. Additionally, ANN models often lack interpretability and physical consistency, rendering them unreliable in application contexts. In this section, we discuss the generalization capabilities of our model.

We consider the ANN trained with the parameters from Tab. S2, with $cg = 4$, and consider a few scenarios slightly departing from the training conditions. In Fig. S3 we report the energy spectrum for four different cases, comparing the results of DNS simulations, NLBM and Smagorinsky, changing the following parameters. Panels from (a) to (d) cover the following set of parameters:

(a) Different magnitude of the forcing term (for reference, value used during training: $F = 5 \times 10^{-6}$, $Re \approx 6 \times 10^3$)

- $F = 1 \times 10^{-5}$ ($Re \approx 1.2 \times 10^4$) ,
- $F = 1 \times 10^{-6}$ ($Re \approx 1.2 \times 10^3$) .

(b) A non-homogeneous forcing term:

$$\begin{cases} F_x(y) = 2F \sin\left(\frac{2\pi}{L}y\right) \\ F_y(z) = 0 \\ F_z(x) = 0 \end{cases} .$$

(c) A higher wave number in the *sine* function:

$$\begin{cases} F_x(y) = F \sin\left(2\frac{2\pi}{L}y\right) \\ F_y(z) = F \sin\left(2\frac{2\pi}{L}z\right) \\ F_z(x) = F \sin\left(2\frac{2\pi}{L}x\right) \end{cases} .$$

(d) Higher and lower values of the kinematic viscosity, changing the relaxation time (for reference, value used in training $\tau = 0.5032$, $Re \approx 6 \times 10^3$)

- $\tau = 0.5048$ ($Re \approx 4 \times 10^3$) ,
- $\tau = 0.5024$ ($Re \approx 8 \times 10^3$) .

The results show that the NLBM yields stable simulations for all the cases covered, considering both higher and lower value for Re with respect to the one considered for training the ANN. NLBM provides results systematically in better agreement with DNS with respect to Smagorinsky, with the exception of one case in Fig. S3(a) where the absolute value of the external force was taken to be 5 times smaller than the one used during training. We attribute this to a possible overfitting of the small scales, which, as can be seen in the plot, overlap for NLBM regardless of the magnitude of the external force.

We should also remark that the generalization capabilities of the model do not extend to different grid sizes and different coarse graining factor; such cases currently require the training of a new model from scratch.

PHYSICAL INTERPRETATION OF MODEL ACTION

In this section we provide a physical interpretation of the action of the ANN. While our model works at the kinetic level, it is simple to map the lattice populations f to the moments space, and observe the action of the model in terms of physical quantities. In order to outline the procedure, we start from Eq. S2 with the single-relaxation time BGK operator in Eq. S4, introduce an invertible matrix \mathbf{M} , and recast the equation in the following form:

$$\begin{aligned} f^{\text{post}} - f^{\text{pre}} &= \mathbf{M}^{-1} \frac{1}{\tau} \mathbf{M} (f^{\text{eq}} - f^{\text{pre}}) + f^{\text{ext}} \\ &= \mathbf{M}^{-1} \mathbf{S} \mathbf{M} (f^{\text{eq}} - f^{\text{pre}}) + f^{\text{ext}} \end{aligned} \quad (\text{S20})$$

where \mathbf{M} defines a transformation from lattice populations $f(\mathbf{x}, t)$ to macroscopic moments $\mathbf{m} = \mathbf{M}f(\mathbf{x}, t)$, and the relaxation matrix $\mathbf{S} = \text{diag}\left(\frac{1}{\tau}, \dots, \frac{1}{\tau}\right)$ acts on macroscopic moments space.

A generalization of the BGK collision operator is given by the *Multi-Relaxation Time* (MRT) collision operator [S14], which allows for individual relaxation rates for the different macroscopic moments. In this framework, several possible choices can be operated for the matrix \mathbf{M} , corresponding to a map of the lattice populations into different sets of macroscopic moments \mathbf{m} , in which the individual relaxation times τ_1, \dots, τ_{19} can have different physical interpretations.

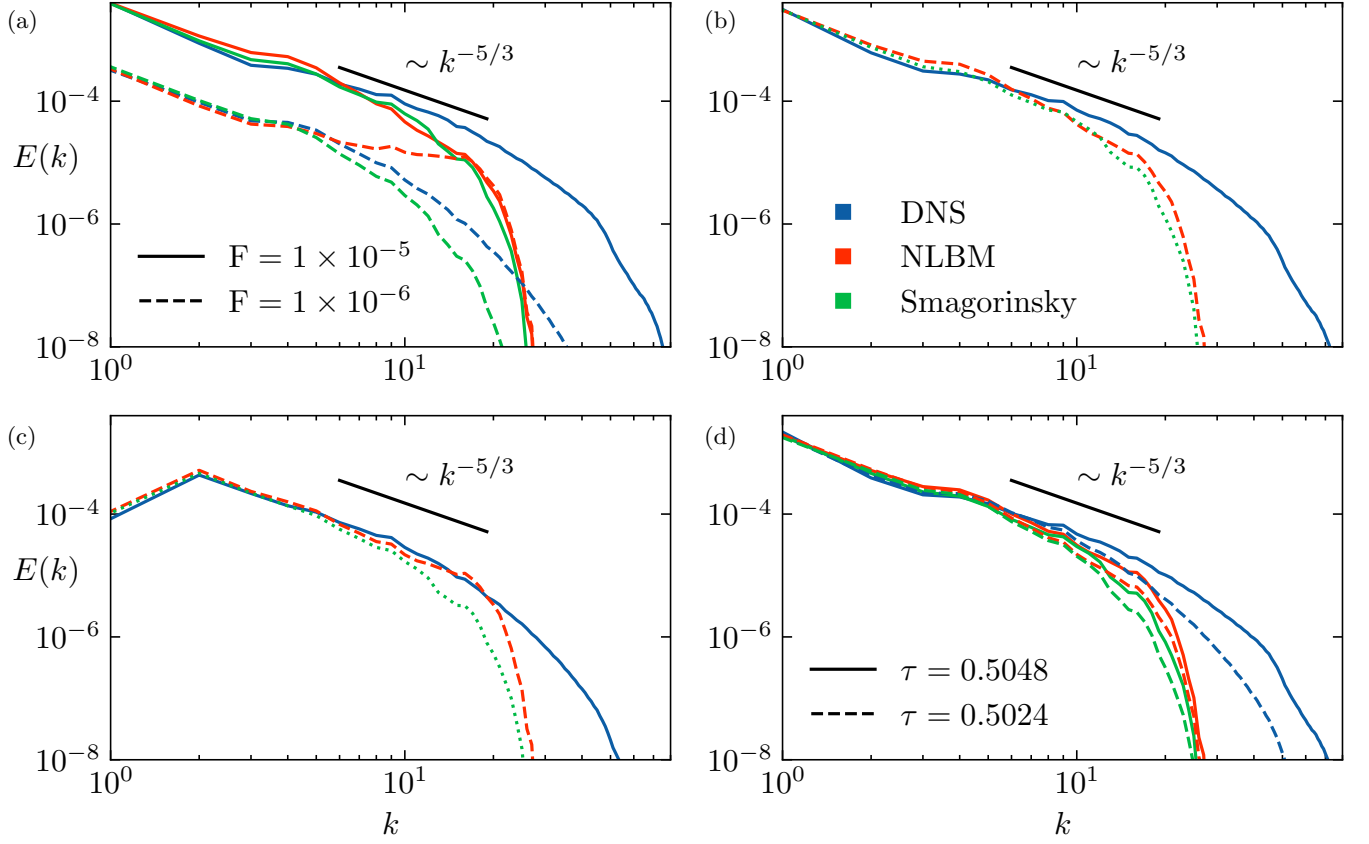


FIG. S3: Energy spectrum for simulations of HIT with different parameters to evaluate the capability of NLBM to generalize outside of the training dataset. Blue curves represent DNS results, red for NLBM and green for the Smagorinsky SGS. In panel a) we present results increasing or decreasing the magnitude of the forcing term, giving respectively $Re \approx 1.2 \times 10^4$ and $Re \approx 1.2 \times 10^3$ (vs $Re \approx 6 \times 10^3$ used at training time). In panel b) we show results using a non-homogeneous forcing term. In panel c) we show the effect of forcing with a higher wave number. In panel d) we show the effect of increasing and decreasing the relaxation time parameter (hence the kinematic viscosity).

In what follows we consider

$$\mathbf{M} = \begin{pmatrix}
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 -30 & -11 & -11 & -11 & -11 & -11 & -11 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
 12 & -4 & -4 & -4 & -4 & -4 & -4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
 0 & 4 & 0 & 0 & 0 & 0 & -4 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
 0 & 0 & -1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & 0 & 0 \\
 0 & 0 & 4 & 0 & 0 & -4 & 0 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & 0 & 0 \\
 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & -1 & 1 & -1 & 1 & 0 & -1 & 0 \\
 0 & 0 & 0 & 4 & -4 & 0 & 0 & 0 & -1 & 1 & 0 & -1 & 1 & -1 & 1 & 0 & -1 & 0 \\
 0 & 2 & -1 & -1 & -1 & -1 & 2 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 & 1 & 1 & 1 \\
 0 & -4 & 2 & 2 & 2 & 2 & -4 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 & 1 & 1 & 1 \\
 0 & 0 & 1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \\
 0 & 0 & -2 & 2 & 2 & -2 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & -1 & 1 & 1 & 1 & 0 & -1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 1 & -1 & 1 & -1 & 0 & -1 & 0
 \end{pmatrix}, \quad (\text{S21})$$

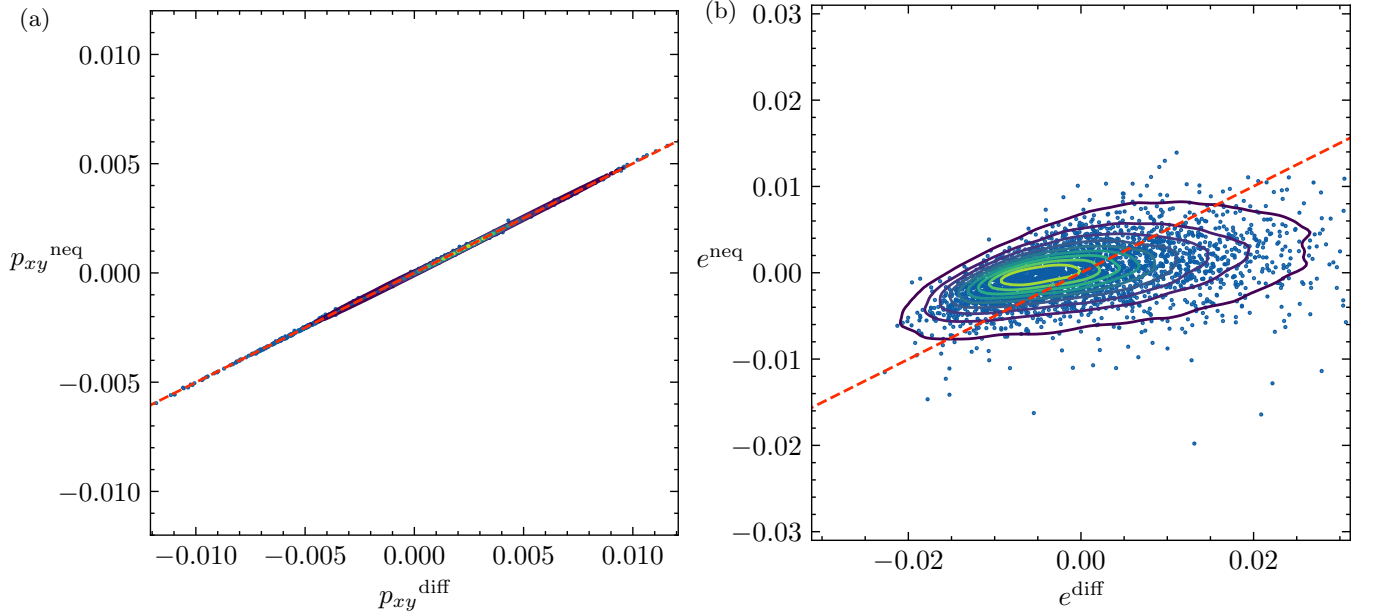


FIG. S4: Scatterplots of \mathbf{m}^{neq} versus \mathbf{m}^{diff} (see text for their definition) from NLBM data for the components corresponding to the component p_{xy} of the shear stress tensor in (a) and to the kinetic energy e in (b). The data shown here comes from a model trained using a coarse graining factor $cg = 4$. In both panels, the red dotted line correspond to the value used for the BGK model at training time $\tau' = 0.5008$ (rescaled with respect to cg as per Eq. S13). A clear linear dependence can be observed for case p_{xy} , which allows to define an effective relaxation time $\tau_{\text{eff}} \approx 0.5023$, slightly larger than τ' , hence leading to a larger effective value of the kinetic viscosity. In panel b) we see instead strong non-linear effects which does not allow for fitting a value of the bulk viscosity.

which maps to the following macroscopic moments:

$$\mathbf{m} = (\rho, e, \epsilon, j_x, q_x, j_y, q_y, j_z, q_z, p_{xx}, \pi_{xx}, p_{ww}, \pi_{ww}, p_{xy}, p_{yz}, p_{xz}, m_x, m_y, m_z), \quad (\text{S22})$$

where:

- ρ, j_x, j_y and j_z are the mass and the components of momentum along x, y, z .
- e is the kinetic energy;
- ϵ is the kinetic energy squared;
- q_x, q_y and q_z are the heat fluxes along x, y, z .
- $p_{xx}, p_{ww}, p_{xy}, p_{yz}$ and p_{xz} are the components of the symmetric traceless viscous stress tensor;
- $\pi_{xx}, \pi_{ww}, m_x, m_y$ and m_z correspond to higher order moments, with no obvious physical interpretation (in the LBM jargon they go under the name of ghost modes).

In turn, the macroscopic moments are associated to the following relaxation parameters:

$$\mathbf{S} = \text{diag} (0, \tau_e, \tau_e, 0, \tau_q, 0, \tau_q, 0, \tau_q, \tau_\nu, \tau_\pi, \tau_\nu, \tau_\pi, \tau_\nu, \tau_\nu, \tau_m, \tau_m, \tau_m), \quad (\text{S23})$$

where the relaxation times associated to conserved quantities have been set to zero.

Likewise for the BGK, also in MRT [S14] it is possible to establish a connection between the relaxation time parameters and the macroscopic transport coefficients. For example, the kinematic viscosity ν and bulk viscosity ζ of the model are given by:

$$\nu = c_s^2 \left(\tau_\nu - \frac{1}{2} \right); \quad \zeta = \frac{5 - 9c_s^2}{9} \left(\tau_e - \frac{1}{2} \right), \quad (\text{S24})$$

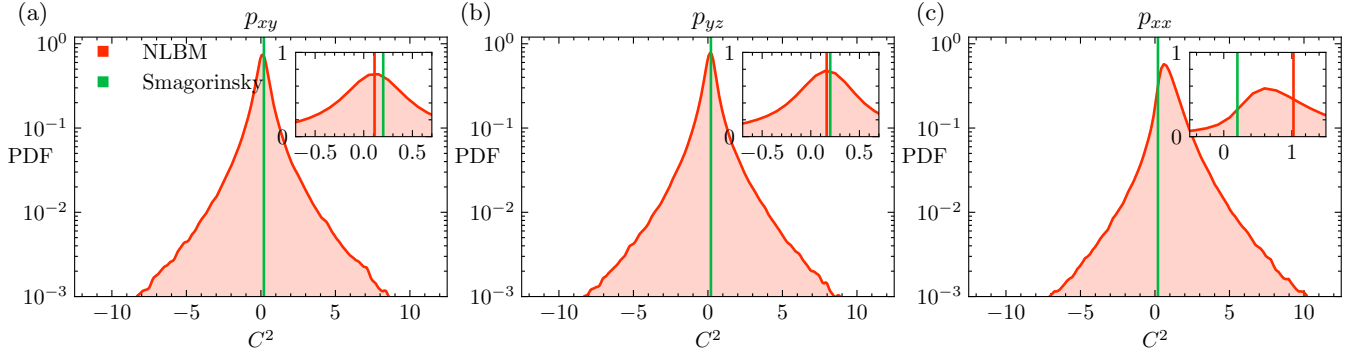


FIG. S5: Probability distribution function (PDF) of the fitted value of the Smagorinsky constant C^2 from NLBM data, for coarse graining factor $cg = 4$. In panel a) data fitted considering the p_{xy} component of the shear stress tensor, in b) using p_{yz} and in c) using p_{xx} . The inset highlights that the average value, marked with vertical bars, comparing the results of NLBM (red) against the value used in the Smagorinsky SGS model in green ($C^2 = 0.2$).

While for NLBM an asymptotic analysis for defining the transport coefficients is not viable from an analytic point of view, it is still possible to numerically evaluate how the ANN relaxes the different moments \mathbf{m} . To this aim we multiply the LHS and the RHS of Eq. S20 by \mathbf{M} :

$$\underbrace{\mathbf{M} (f^{\text{post}} - f^{\text{pre}} - f^{\text{ext}})}_{\mathbf{m}^{\text{diff}}} = \mathbf{S} \underbrace{\mathbf{M} (f^{\text{eq}} - f^{\text{pre}})}_{\mathbf{m}^{\text{neq}}}, \quad (\text{S25})$$

and define \mathbf{m}^{diff} and \mathbf{m}^{neq} , two column vectors defined in the moment space similarly to \mathbf{m} .

From numerical data, we can plot the different components of \mathbf{m}^{diff} vs \mathbf{m}^{neq} . In Fig. S4 we report an example of this analysis, for the case $cg = 4$, where the individual components are centered with respect to their average value. We remark that in a MRT collision operator we would observe a linear dependence, with a slope of the regression line $1/\tau_i$ depending on the choice of the relaxation parameters in \mathbf{S} . For NLBM we observe, to good approximation, a linear relationship for the moments related to the shear stress tensor p_{ij} . In Fig. S4(a) we provide an example for the component p_{xy} . From a linear fit we can obtain an effective value for the relaxation time τ_{eff} . At variance with Eq. S23, we observe two different relaxation times, one for the components p_{xx} and p_{yy} ($\tau_{\text{eff}} \approx 0.5125$), and a second one for p_{xy} , p_{yz} and p_{zx} , ($\tau_{\text{eff}} \approx 0.5023$). Both values are larger than the corresponding value of the relaxation time rescaled as for Eq. S13, $\tau = 0.5008$, thus implying a larger effective shear viscosity.

For all the other moments we generally observe a non linear dependence. In Fig. S4(b) we show an example for the kinetic energy coefficient which relates to the bulk viscosity. It is common practice in literature to adjust the bulk viscosity to enhance the stability of numerical methods, and recently an ANN-based approach has been reported also for LBM [S15].

To further test the importance of non-linearities introduced by the ANN we have tested a MRT collision operator in which the matrix \mathbf{S} has been obtained performing a linear fit of τ_{eff} for all the different moments independently (hence discarding the non-linear contributions):

$$\mathbf{S} = \text{diag} (0, \tau_1, \tau_2, 0, \tau_3, 0, \tau_4, 0, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9, \tau_{10}, \tau_{11}, \tau_{12}, \tau_{13}, \tau_{14}, \tau_{15},) \quad (\text{S26})$$

with

$$\begin{aligned} \tau_1 &= 0.1554, & \tau_2 &= 0.0947, & \tau_3 &= 0.5220, & \tau_4 &= 0.6921, & \tau_5 &= 0.5614, \\ \tau_6 &= 0.5122, & \tau_7 &= 0.5564, & \tau_8 &= 0.5121, & \tau_9 &= 0.5483, & \tau_{10} &= 0.5020, \\ \tau_{11} &= 0.5028, & \tau_{12} &= 0.5023, & \tau_{13} &= 0.4597, & \tau_{14} &= 0.4910, & \tau_{15} &= 0.4872 \end{aligned}$$

With this choice of \mathbf{S} numerical simulations became unstable after just few iterations, similarly to the BGK case. We leave as a future work further investigation on the role of the heat flux and of the ghost modes. We focus here instead on the effective viscosity, which can be computed from the shear stress tensor akin to the Smagorinsky turbulence model. From the linear fit of the components of the shear stress tensor it is possible to compute ν_{eff} , which in combination with Eq. S7 allows to define what would be the equivalent of the Smagorinsky constant for NLBM. In Fig. S5 we report the PDF of the fitted value of the Smagorinsky constant from NLBM data, with in panel (a) and

(b) the results for p_{xy} and p_{yz} respectively, and in (c) for p_{xx} . The average values are comparable with Smagorinsky for cases (a) and (b), about a factor two smaller than the one used in simulations with the Smagorinsky SGS model ($C^2 = 0.2$), with the largest discrepancies observed for case (c). For all cases, the presence of a tail with negative values highlights the fact that in NLBM it is possible to capture the inverse transfer of energy from small to large scales.

-
- [S1] T. Krueger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, E. Viggien, *The Lattice Boltzmann Method: Principles and Practice* (Springer, 2016).
 - [S2] S. Succi, *The Lattice Boltzmann Equation: For Complex States of Flowing Matter* (Oxford Scholarship Online, Oxford, 2018).
 - [S3] P.C. Philippi, L.A. Hegele, L.O.E. dos Santos, R. Surmas, *Phys. Rev. E* **73**, 056702 (2006).
 - [S4] X. Shan, *Journal of Computational Science* **17**, 475–481 (2016).
 - [S5] P.L. Bhatnagar, E.P. Gross, M.Krook, *Phys. Rev.* **94**, 511 (1954).
 - [S6] A. L. Kupershtokh, D.A. Medvedev, and D. Karpov, *Computers & Mathematics with Applications* **58**(5), 965–974 (2009).
 - [S7] S. Chapman and T.G. Cowling, *The Mathematical Theory of Non-Uniform Gases* (American Journal of Physics, 1952).
 - [S8] A. Corbetta, A. Gabbana, V. Gyrya, D. Livescu, J. Prins, and F. Toschi, *The European Physical Journal E* **46**, 10 (2023).
 - [S9] J. Brandstetter, D. E. Worrall, and M. Welling, [arXiv:2202.03376](#), (2022).
 - [S10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).
 - [S11] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, [arXiv:1904.09751](#), (2019).
 - [S12] E. Goles, *Discrete Applied Mathematics* **13**, 97–100 (1986).
 - [S13] A. Labach, H. Salehinejad, and S. Valaee, [arXiv:1904.13310](#), (2019).
 - [S14] D. D’Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo, *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* **360**, 437–51 (2002).
 - [S15] J. T. Horstmann, M. C. Bedrunka, and H. Foyi, *Computers & Fluids* **272**, 106191 (2024).