



Multiscale simulations of complex systems by learning their effective dynamics

Pantelis R. Vlachas^{1,2}, Georgios Arampatzis^{1,2}, Caroline Uhler³ and Petros Koumoutsakos^{1,2}

Predictive simulations of complex systems are essential for applications ranging from weather forecasting to drug design. The veracity of these predictions hinges on their capacity to capture effective system dynamics. Massively parallel simulations predict the system dynamics by resolving all spatiotemporal scales, often at a cost that prevents experimentation, while their findings may not allow for generalization. On the other hand, reduced-order models are fast but limited by the frequently adopted linearization of the system dynamics and the utilization of heuristic closures. Here we present a novel systematic framework that bridges large-scale simulations and reduced-order models to learn the effective dynamics of diverse, complex systems. The framework forms algorithmic alloys between nonlinear machine learning algorithms and the equation-free approach for modelling complex systems. Learning the effective dynamics deploys autoencoders to formulate a mapping between fine- and coarse-grained representations and evolves the latent space dynamics using recurrent neural networks. The algorithm is validated on benchmark problems, and we find that it outperforms state-of-the-art reduced-order models in terms of predictability, and large-scale simulations in terms of cost. Learning the effective dynamics is applicable to systems ranging from chemistry to fluid mechanics and reduces the computational effort by up to two orders of magnitude while maintaining the prediction accuracy of the full system dynamics. We argue that learning the effective dynamics provides a potent novel modality for accurately predicting complex systems.

Some of the most important scientific advances and engineering designs are founded on the study of complex systems that exhibit dynamics spanning multiple spatiotemporal scales. Examples include protein dynamics¹, morphogenesis², brain dynamics³, climate⁴, ocean dynamics⁵ and social systems⁶. Over the last 50 years, simulations have become a key component of these studies thanks to a confluence of advances in computing architectures, numerical methods and software. Large-scale simulations have led to unprecedented insight, acting as *in silico* microscopes⁷ or telescopes to reveal the dynamics of galaxy formations⁸. At the same time, these simulations have led to the understanding that resolving the full range of spatiotemporal scales in such complex systems will remain out of reach for the foreseeable future.

In recent years there have been intense efforts to develop efficient simulations that exploit the multiscale character of the systems under investigation^{9–12}. Multiscale methods rely on judicious approximations of the interactions between processes occurring over different scales, and a number of potent frameworks have been proposed, including the equation-free framework (EFF)^{10,12–14}, the heterogeneous multiscale method (HMM)^{11,15,16} and the Flow Averaged Integrator (FLAVOR)¹⁷. In these algorithms the system dynamics are distinguished into fine and coarse scales or expensive and affordable simulations, respectively. Their success depends on the separation of scales that are inherent to the system dynamics and their capability to capture the transfer of information between scales. Effective applications of multiscale methodologies minimize the computational effort while maximizing the accuracy of the propagated dynamics. The EFF relies on a few fine-scale simulations that are used to acquire, through ‘restricting’, information about the evolution of the coarse-grained quantities of interest. In turn, various time-stepping procedures are used to propagate the coarse-grained

dynamics. The fine-scale dynamics are obtained by judiciously ‘lifting’ the coarse scales to return to the fine-scale description of the system and repeat. When the EFF reproduces trajectories of the original system, the identified low-order dynamics represent the intrinsic system dynamics, also called effective dynamics, inertial manifold^{18,19} or reaction coordinates in molecular kinetics.

While it is undisputed that the EFF, HMM, FLAVOR and related frameworks have revolutionized the field of multiscale modelling and simulation, we identify two critical issues that currently limit their potential. First, the accuracy of propagating the coarse-grained/latent dynamics hinges on the employed time integrators. Second, the choice of information transfer, particularly from coarse- to fine-scale dynamics in lifting, affects the forecasting capacity of the methods.

In the present work, these two critical issues are resolved through machine learning (ML) algorithms that (1) deploy recurrent neural networks (RNNs) with gating mechanisms to evolve the coarse-grained dynamics and (2) employ advanced (convolutional, or probabilistic) autoencoders (AEs) to transfer in a systematic, data-driven manner the information between coarse- and fine-scale descriptions.

In recent years, ML algorithms have exploited the ample availability of data, and powerful computing architectures, to provide us with remarkable successes across scientific disciplines^{20,21}. The particular elements of our algorithms have been employed in the modelling of dynamical systems. AEs have been used to identify a latent linear space on the basis of the Koopman framework²², model high-dimensional fluid flows^{23,24} or sample effectively the state space in the kinetics of proteins²⁵. More recently, AEs have been coupled with dynamic importance sampling²⁶ to accelerate multiscale simulations and investigate the interactions of RAS proteins with a plasma

¹Computational Science and Engineering Laboratory, ETH Zürich, Zürich, Switzerland. ²School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. ³Institute for Data, Systems, and Society, Massachusetts Institute of Technology, Cambridge, MA, USA. e-mail: petros@seas.harvard.edu

membrane. RNNs with gating mechanisms have been successfully applied to a wide range of applications, from speech processing²⁷ to complex systems²⁸, but their effectiveness in a multiscale setting has yet to be investigated. AEs coupled with RNNs are used in refs. 29–31 to model fluid flows. In ref. 32, the authors build on the EFF framework, identify a partial differential equation (PDE) on a coarse representation by diffusion maps, Gaussian processes or neural networks and utilize forward integration in the coarse representation. These previous works, however, fail to employ one or more of the following mechanisms, in contrast to our framework: consider the coarse-scale dynamics^{23,24}, account for their non-Markovian^{26,32} or nonlinear nature²², exploit a probabilistic generative mapping^{23,29–31} from the coarse to the fine scale, learn simultaneously the latent space and its dynamics in an end-to-end fashion and not sequentially^{22,23,26,29–32}, alternate between micro and macrodynamics^{22,23,29–32} and scale to high-dimensional systems^{29,30,32}.

Augmenting multiscale frameworks (including EFF, HMM and FLAVOR) with state-of-the-art ML algorithms allows for evolution of the coarse-scale dynamics by taking into account their time history and by providing consistent lifting (decoding) and restriction (encoding) operators to transfer information between fine and coarse scales. We demonstrate that the proposed framework allows for simulations of complex multiscale systems that reduce the computational cost by orders of magnitude to capture spatiotemporal scales that would be impossible to resolve with existing computing resources.

Learning the effective dynamics (LED)

We propose a framework for LED of complex systems, that allows for accurate prediction of the system evolution at a considerably reduced computational cost.

In the following, the high-dimensional state of a dynamical system is given by $\mathbf{s}_t \in \mathbb{R}^{d_s}$, and the discrete time dynamics are given by

$$\mathbf{s}_{t+\Delta t} = \mathbf{F}(\mathbf{s}_t),$$

where Δt is the sampling period and \mathbf{F} may be nonlinear, deterministic or stochastic. We assume that the state of the system at time t can be described by a vector $\mathbf{z}_t \in \mathcal{Z}$, where $\mathcal{Z} \subset \mathbb{R}^{d_z}$ is a low-dimensional manifold with $d_z \ll d_s$. To identify this manifold, an encoder $\mathcal{E}^{\mathbf{w}_\mathcal{E}} : \mathbb{R}^{d_s} \rightarrow \mathbb{R}^{d_z}$ is utilized, where $\mathbf{w}_\mathcal{E}$ are trainable parameters, transforming the high-dimensional state \mathbf{s}_t to $\mathbf{z}_t = \mathcal{E}^{\mathbf{w}_\mathcal{E}}(\mathbf{s}_t)$. In turn, a decoder maps this latent representation back to the high-dimensional state, that is, $\tilde{\mathbf{s}}_t = \mathcal{D}^{\mathbf{w}_\mathcal{D}}(\mathbf{z}_t)$.

For deterministic systems, the optimal parameters $\{\mathbf{w}_\mathcal{E}^*, \mathbf{w}_\mathcal{D}^*\}$ are identified by minimizing the mean squared reconstruction error (MSE):

$$\mathbf{w}_\mathcal{E}^*, \mathbf{w}_\mathcal{D}^* = \arg \min_{\mathbf{w}_\mathcal{E}, \mathbf{w}_\mathcal{D}} \langle (\mathbf{s}_t - \tilde{\mathbf{s}}_t)^2 \rangle = \arg \min_{\mathbf{w}_\mathcal{E}, \mathbf{w}_\mathcal{D}} \langle \{\mathbf{s}_t - \mathcal{D}^{\mathbf{w}_\mathcal{D}}[\mathcal{E}^{\mathbf{w}_\mathcal{E}}(\mathbf{s}_t)]\}^2 \rangle,$$

where $\langle \cdot \rangle$ denotes the mean. Convolutional neural network³³ auto-encoders (CNN-AEs) that take advantage of the spatial structure of the data are embedded into LED.

For stochastic systems, $\mathcal{D}^{\mathbf{w}_\mathcal{D}}$ is modelled with a mixture density (MD) decoder³⁴. Further details are provided in Supplementary Section 1E on the implementation of the MD decoder, along with other components embedded in LED: AEs in Supplementary Section 1A, variational AEs in Supplementary Section 1B and CNNs in Supplementary Section 1C.

We demonstrate the modularity of LED, as it can be coupled with a permutation-invariant layer (see details in Supplementary Section 1D) and utilized later in the modelling of the dynamics of a large set of particles governed by the advection–diffusion equation (see details in Supplementary Section 3A; hyperparameter tuning of LED reported in Supplementary Tables 1 and 2).

As a nonlinear propagator in the low-order manifold (coarse scale), an RNN is employed, capturing non-Markovian, memory effects by keeping an internal memory state. The RNN is learning a forecasting rule

$$\mathbf{h}_t = \mathcal{H}^{\mathbf{w}_\mathcal{H}}(\mathbf{z}_t, \mathbf{h}_{t-\Delta t}), \quad \tilde{\mathbf{z}}_{t+\Delta t} = \mathcal{R}^{\mathbf{w}_\mathcal{R}}(\mathbf{h}_t),$$

where $\mathbf{h}_t \in \mathbb{R}^{d_h}$ is an internal hidden memory state, $\tilde{\mathbf{z}}_{t+\Delta t}$ is a latent-state prediction, $\mathcal{H}^{\mathbf{w}_\mathcal{H}}$ and $\mathcal{R}^{\mathbf{w}_\mathcal{R}}$ are the hidden-to-hidden and the hidden-to-output mappings, and $\mathbf{w}_\mathcal{H}$ and $\mathbf{w}_\mathcal{R}$ are the trainable parameters of the RNN. One possible implementation of $\mathcal{H}^{\mathbf{w}_\mathcal{H}}$ and $\mathcal{R}^{\mathbf{w}_\mathcal{R}}$ is the long short-term memory (LSTM)³⁵, presented in Supplementary Section 1F.

The role of the RNN is twofold. First, it is updating its hidden memory state \mathbf{h}_t , given the current state provided at the input \mathbf{z}_t and the hidden memory state at the previous time-step $\mathbf{h}_{t-\Delta t}$, tracking the history of the low-order state to model non-Markovian dynamics. Second, given the updated \mathbf{h}_t , the RNN forecasts the latent state at the next time-step(s) $\tilde{\mathbf{z}}_{t+\Delta t}$. The RNN is trained to minimize the forecasting loss $\|\tilde{\mathbf{z}}_{t+\Delta t} - \mathbf{z}_{t+\Delta t}\|_2^2$ by backpropagation through time³⁶.

The LSTM and the AE, jointly referred to as LED, are trained on data from simulations of the fully resolved (or microscale) dynamical system. The two networks can be trained either sequentially or together. In the first case, the AE is pretrained to minimize the reconstruction loss, and then the LSTM is trained to minimize the prediction loss on the latent space (AE-LSTM). In the second case, they are seen as one network trying to minimize the sum of reconstruction and prediction losses (AE-LSTM-end2end). For large, high-dimensional systems, the latter approach of end-to-end training is computationally expensive. After training, LED is employed to forecast the dynamics on unseen data by propagating the low-order latent state with the RNN and avoiding the computationally expensive simulation of high-dimensional dynamics. We refer to this mode of propagation, iteratively propagating only the latent/macro-dynamics, as Latent-LED. We note that, as non-Markovian models are not self-starting, an initial small warm-up period is required, feeding the LED with data from the microdynamics.

The LED framework allows for data-driven information transfer between coarse and fine scales through the AE. Moreover, it propagates the latent space dynamics without the need to upscale back to the high-dimensional state space at every time-step. As is the case for any approximate iterative integrator (here the RNN), the initial model errors will propagate. To mitigate potential instabilities, inspired by the equation-free approach¹⁰, we propose the multiscale forecasting scheme in Fig. 1, alternating between microdynamics for T_μ and macrodynamics for T_m . In this way, the approximation error can be reduced at the cost of the computational complexity associated with evolving the high-dimensional dynamics. We refer to this mode of propagation as Multiscale-LED, and the ratio $\rho = T_m/T_\mu$ as the multiscale ratio. In Multiscale-LED, the interface with the high-dimensional state space is enabled only at the time-steps and scales of interest. This is in contrast to refs. 37,38, and is easily adaptable to the needs of particular applications, thus augmenting the arsenal of models developed for multiscale problems.

Training of LED models is performed with the Adam stochastic optimization method³⁹, and validation-based early stopping is employed to avoid overfitting. All LED models are implemented in PyTorch, mapped to a single Nvidia Tesla P100 graphics processing unit, and executed on the XC50 compute nodes of the Piz Daint supercomputer at the Swiss National Supercomputing Centre (CSCS).

Results

We demonstrate the application of LED in a number of benchmark problems and compare its performance with that of existing

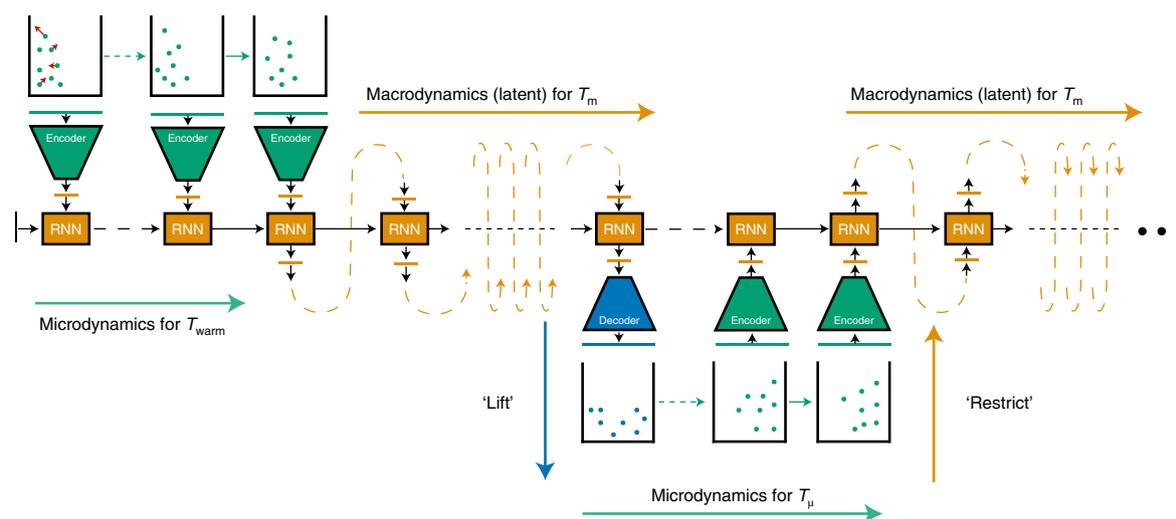


Fig. 1 | Multiscale-LED. Starting from an initial condition, use the equations/first principles to evolve the high-dimensional dynamics for a short period T_{warm} . During this warm-up period, \mathbf{s}_t is passed through the encoder network. The outputs of the AE are iteratively provided as inputs to the RNN, to warm up its hidden state. Next, iteratively, (1) starting from the last latent state \mathbf{z}_t propagate the latent dynamics for $T_m \gg T_{\text{warm}}$ with the RNN, (2) lift the latent dynamics at $t = T_{\text{warm}} + T_m$ back to the high-dimensional state and (3) starting from this high-dimensional state as an initial condition use the equations/first principles to evolve the dynamics for $T_\mu \ll T_m$.

state-of-the-art algorithms. In Supplementary Section 3D, we provide additional results on LED applied to alanine dipeptide in water. The stochastic dynamics of the molecular system are handled with an MD decoder, and an MD-LSTM in the latent space⁴⁰.

FitzHugh–Nagumo model. LED is employed to capture the dynamics of the FitzHugh–Nagumo equations (FHN)^{41,42}. The FHN model describes the evolution of an activator $u(x, t) = \rho^{\text{ac}}(x, t)$ and an inhibitor density $v(x, t) = \rho^{\text{in}}(x, t)$ on the domain $x \in [0, L]$:

$$\begin{aligned} \frac{\partial u}{\partial t} &= D^u \frac{\partial^2 u}{\partial x^2} + u - u^3 - v, \\ \frac{\partial v}{\partial t} &= D^v \frac{\partial^2 v}{\partial x^2} + \epsilon(u - \alpha_1 v - \alpha_0). \end{aligned} \quad (1)$$

The system evolves periodically under two timescales, with the activator/inhibitor density acting as the ‘fast’/‘slow’ variable respectively. The bifurcation parameter $\epsilon = 0.006$ controls the difference in the timescales. We choose $D^u = 1$, $D^v = 4$, $L = 20$, $\alpha_0 = -0.03$ and $\alpha_1 = 2$.

Equation (1) is discretized with $N = 101$ grid points and solved using the lattice Boltzmann (LB) method⁴³, with time-step $\delta_t = 0.005$. To facilitate comparison with ref. ³², we employ the LB method to gather data starting from six different initial conditions to obtain the mesoscopic solution considered here as the fine-scale solution. The data are subsampled, retaining every 200th data point, that is, the coarse time-step is $\Delta t = 1$. Three time series with 451 points are considered for training, two time series with 451 points for validation and 10^4 data points from a different initial condition for testing. For the identification of the latent space, we compare principal component analysis (PCA), diffusion maps, feedforward AE and CNN-AE, in terms of the MSE of the reconstruction in the test data, plotted in Fig. 2a. The MSE is plateauing after $d_z = 2$, and the AE and CNN-AE exhibit at least an order of magnitude lower MSR compared with PCA and diffusion maps. For this reason, we employ an AE with $d_z = 2$ for the LED. The hyperparameters of the networks (reported in Supplementary Table 3 along with training times) are tuned on the basis of the MSE on the validation data. The architecture of the CNN is reported in Supplementary Table 5 and depicted in Supplementary Fig. 10.

In Fig. 2b, we compare various propagators in the forecasting of the macro- (latent) dynamics, starting from 32 different initial conditions in the test data, up to a horizon of $T_f = 8,000$. We benchmark an AE-LSTM trained end-to-end (AE-LSTM-end2end), an AE-LSTM where the AE is pretrained (AE-LSTM), a multilayered perceptron (AE-MLP), Reservoir Computers (AE-RC)^{28,44} and the SINDy algorithm (AE-SINDy)⁴⁵. As a comparison metric, we consider the mean normalized absolute difference (MNAD), averaged over the 32 initial conditions. The definition of the MNAD is provided in Supplementary Section 2. The MNAD is computed on the inhibitor density, as the difference between the result of the LB simulation $v(x, t)$, considered as ground truth, and the model forecasts \hat{v} . The warm-up period for all propagators is set to $T_{\text{warm}} = 60$. The hyperparameters of the networks (reported in Supplementary Tables 4, 6 and 7, along with the training times) are tuned on the basis of the MNAD on the validation data. The LSTM-end2end and the RC show the lowest test error, while the variance of the RC is larger. In the following, we consider an LSTM-end2end propagator for the LED.

LED is benchmarked against EFF variants³² in the FHN equation in Fig. 2c. As a metric for the accuracy, the MNAD is considered consistent with ref. ³² to facilitate comparison. The EFF variants³² are based on the identification of PDEs on the coarse level (CSPDE). LED is compared with CSPDEs in forecasting the dynamics of the FHN equation starting from an initial condition from the test data up to final time $T_f = 451$. CSPDE variants are utilizing Gaussian processes (GP) or neural networks (NN), features of the fine-scale dynamics obtained through diffusion maps (F1 to F3) and forward integration to propagate the coarse representation in time. LED outperforms CSPDE variants by an order of magnitude. In Fig. 2f, the latent space of LED is plotted against the attractor of the data embedded in the latent space. Even for long time horizons (here $T_f = 8,000$), the LED forecasts remain on the periodic attractor.

Latent-LED propagates the low-order dynamics and upscales back to the inhibitor density, forecasting its evolution accurately while being 60 times faster than the LB solver. This speed-up can be decisive in accelerating simulations and achieving much longer time horizons.

In Multiscale-LED, the approximation error of LED decreases at the cost of reduced speed-up. This interplay can be seen in Fig. 2d,e.

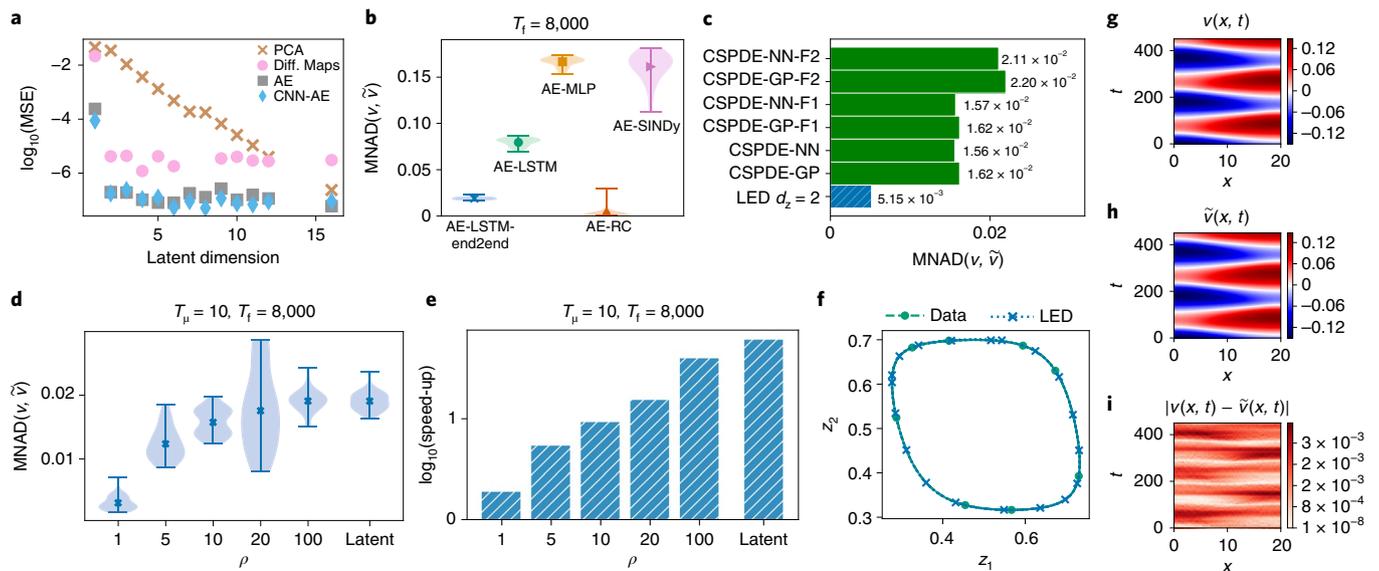


Fig. 2 | FitzHugh-Nagumo model. **a**, Comparison of the reconstruction MSE in the test data in the FHN between PCA, diffusion maps, AE and convolutional AE. **b**, Comparison of macrodynamics propagators in iterative latent forecasting. **c**, Comparison of Latent-LED with $d_z = 2$ with equation-free variants from ref. ³². **d**, The inhibitor MNAD of Multiscale-LED (AE-LSTM-end2end, $d_z = 2$) plotted as a function of ρ . **e**, The evolution of the LED latent state compared against the test data embedded in the latent space. **f**, The evolution of the LED latent state compared against the test data embedded in the latent space. **g–i**, A trajectory of the inhibitor density starting from a testing initial condition (**g**), along with the Latent-LED prediction (**h**) and absolute difference (**i**).

Latent-LED ($T_m = 0$), and Multiscale-LED, alternating between macrodynamics for $T_m = 10$ and high-dimensional dynamics for T_m , are employed to approximate the evolution and compare it against the LB solver in forecasting up to $T_i = 8,000$ starting from 32 initial conditions as before. For $T_m = T_m = 10$ ($\rho = 1$), the MNAD is reduced from approximately 0.019 to approximately 0.003 compared with Latent-LED. The speed-up, however, is reduced from 60 to 2. By varying $T_m \in \{50, 100, 200, 1,000\}$, Multiscale-LED achieves a trade-off between speed-up and MNAD.

A prediction of the Latent-LED in the inhibitor density is compared against the ground truth in Fig. 2g–i. Additional results on the activator density are given in Supplementary Section 3B.

Kuramoto–Sivashinsky equation. The Kuramoto–Sivashinsky equation (KS)^{46,47} is a prototypical PDE of fourth order that exhibits a very rich range of nonlinear phenomena. In the case of high dissipation and small spatial extent L (domain size), the long-term dynamics of KS can be represented on a low-dimensional inertial manifold^{18,19} that attracts all neighbouring states at an exponential rate after a transient period. LED is employed to learn the low-order manifold of the effective dynamics in KS.

The one-dimensional KS equation is given by the PDE

$$\frac{\partial u}{\partial t} = -\nu \frac{\partial^4 u}{\partial x^4} - \frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x}, \quad (2)$$

on the domain $\Omega = [0, L]$ with periodic boundary conditions $u(0, t) = u(L, t)$ and $\nu = 1$. The special case $L = 22$ considered in this work is studied extensively in ref. ⁴⁸, and exhibits a structurally stable chaotic attractor, that is, an inertial manifold where the long-term dynamics lie. Equation (2) is discretized with a grid of size 64 points and solved using the fourth-order method for stiff PDEs introduced in ref. ⁴⁹ with a time-step of $\delta t = 2.5 \times 10^{-3}$ starting from a random initial condition. The data are subsampled to $\Delta t = 0.25$ (coarse time-step of LED). 15×10^3 samples are used for training and another 15×10^3 for validation. For testing purposes,

the process is repeated with a different random seed, generating another 15×10^3 samples.

For the identification of a reasonable latent space dimension, we compare PCA, AEs and CNNs in terms of the reconstruction MSE in the test data as a function of d_z , plotted in Fig. 3a. MSE is plateauing after $d_z = 8$, indicating arguably the dimensionality of the attractor in agreement with previous studies^{18,48}, and that the CNN is superior to the AE, and orders of magnitude better than PCA. For this reason, we employ a CNN with $d_z = 8$ for the autoencoding part of LED. The hyperparameters of the networks are tuned on the basis of the MSE on the validation data, reported in Supplementary Tables 8 and 9 with the network training times. The CNN architecture is provided in Supplementary Table 10, and depicted in Supplementary Fig. 12.

In Fig. 3b, we compare various propagators in predicting the macrodynamics of LED, starting from 100 test initial conditions, up to $T_i = 800$ (3,200 time-steps). We employ a CNN-LSTM trained end-to-end (CNN-LSTM-end2end), a CNN-LSTM where the CNN is pretrained (CNN-LSTM), a multilayered perceptron (CNN-MLP), Reservoir Computers (CNN-RC)^{28,44} and the SINDy algorithm (CNN-SINDy)⁴⁵. As a comparison metric, we consider the MNAD, averaged over the 100 initial conditions. The warm-up period for all propagators is set to $T_{\text{warm}} = 60$. The hyperparameters (reported in Supplementary Tables 11–13, along with the training times) are tuned on the basis of the MNAD on the validation data. While the MLP and RC propagators exhibit large errors, the LSTM, LSTM-end2end and SINDy show comparable accuracy. In the following, we consider an LSTM propagator for the LED.

Due to the chaoticity of the KS equation, iterative forecasting with LED is challenging, as initial errors propagate exponentially. To assess whether the iterative forecasting with LED leads to reasonable, physical predictions, we plot the density of values in the $u_x - u_{xx}$ space in Fig. 3c. The data originate from a single long trajectory of size $T_i = 8,000$ (32,000 time-steps). We observe that LED, Fig. 3d, is able to qualitatively reproduce the density of the simulation.

In Fig. 3e, f, we plot the MNAD, and correlation between forecasts of LED and the reference with respect to ρ . In Fig. 3g, the

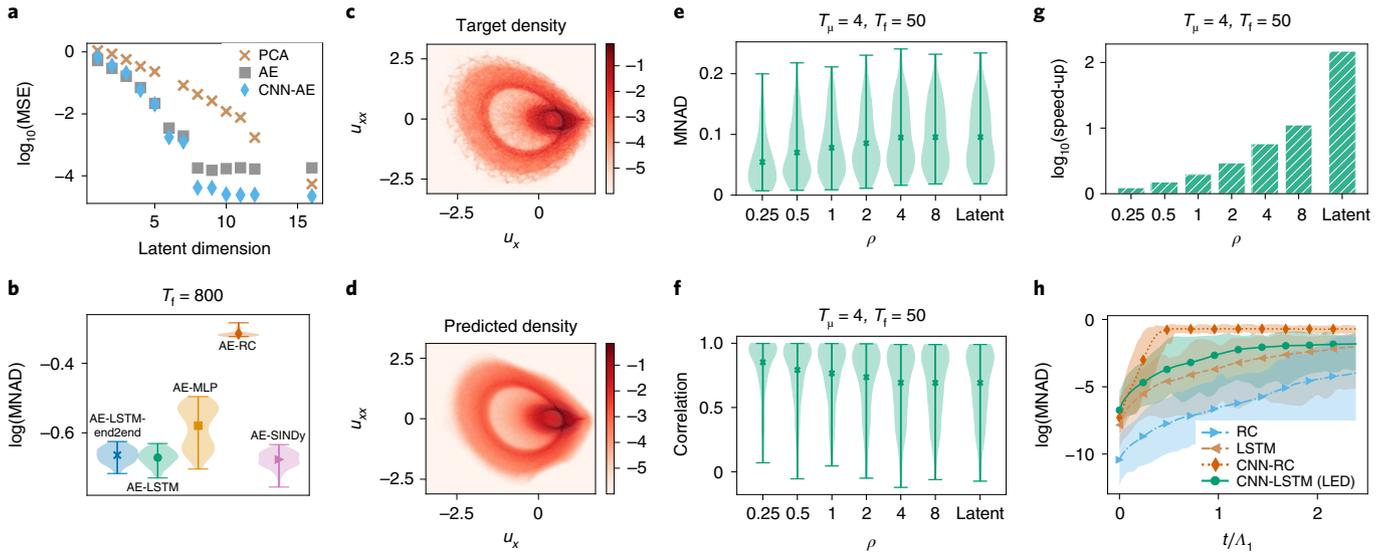


Fig. 3 | Kuramoto-Sivashinsky equation. **a**, Comparison of the reconstruction MSE in the test data in the KS dynamics between PCA, AE and convolutional AE as a function of the latent dimension. **b**, Comparison of different macrodynamics propagators in iterative latent forecasting. **c,d**, The density of values in the u_x-u_y space computed from 32,000 time-steps (**c**), which matches closely the prediction (**d**), illustrating that the LED is able to replicate characteristics of the dynamical system and remain at the attractor, even though propagating coarse dynamics. **e,f**, The MNAD (**e**) and the correlation between LED predictions and the reference data (**f**) as a function of ρ . **g**, The speed-up of LED with respect to ρ . Evolution of the latent state of LED ($T_\mu = 0$) is up to two orders of magnitude cheaper than the microscale dynamics. **h**, Comparison of CNN-LSTM (LED), CNN-RC, LSTM and RC in short-term forecasting of the KS dynamics; time is normalized with the Lyapunov time $T^{A_1} = 1/\Lambda_1 = 20.83$.

speed-up of LED is plotted against ρ . Latent-LED is able to reproduce the long-term ‘climate dynamics’²⁸ and remain at the attractor while being more than two orders of magnitude faster compared with the micro solver. As ρ is increased, the error is reduced (correlation increased), at the cost of reduced speed-up.

Finally, in Fig. 3h, we compare the performance of Latent-LED (CNN-LSTM) with previous studies^{28,44}, which forecast directly on the high-dimensional space. Specifically, the Latent-LED matches the performance of an LSTM (no dimensionality reduction) but shows inferior short-term forecasting ability compared with an RC (no dimensionality reduction) forecasting on the high-dimensional space. This is expected, as the RC and the LSTM have full information about the state. In turn, when the RC is employed on the latent space of LED as a macrodynamics propagator, the error grows considerably, and the performance is inferior to the CNN-LSTM case.

A forecast of Latent-LED is provided in Supplementary Fig. 11.

Viscous flow past a cylinder. The flow past a cylinder is a widely studied problem in fluids⁵⁰, which exhibits a rich range of dynamical phenomena such as the transition from laminar to turbulent flow at high Reynolds numbers and is used as a benchmark for reduced-order modelling approaches. The flow past a cylinder in the two-dimensional space is simulated by solving the incompressible Navier–Stokes equations with Brinkman penalization to enforce the no-slip boundary conditions on the surface of the cylinder^{51,52}. More details on the simulation are provided in Supplementary Section 3D. We consider the application of LED to two Reynolds numbers, $Re \in \{100, 1000\}$. The definition of Re is provided in Supplementary equation (22).

The flow is simulated in a cluster with 12 CPU cores, up to $T=200$, after discarding the initial transients. 250 time-steps distanced $\Delta t=0.2$ in time (total time $T=50$) are used for training, 250 for validation and the rest for testing purposes. The vortex shedding period is $T \approx 2.86$ for $Re=100$ and $T \approx 2.22$ for $Re=1,000$.

The state of LED is $\mathbf{s}_t \equiv \{p, u_x, u_y, \omega\} \in \mathbb{R}^{4 \times 512 \times 1024}$, where ω is the vorticity field. For the autoencoding part, LED employs

CNNs that take advantage of spatial correlations. The architecture of the CNN is given in Supplementary Table 14 and depicted in Supplementary Fig. 13. The dimension of the latent space is tuned on the basis of the performance on the validation dataset to $d_z=4$ for $Re=100$ and $d_z=10$ for $Re=1,000$.

A prediction of ω by Latent-LED at lead time $T=4$ is given in Fig. 4. LED captures the flow for both $Re \in \{100, 1,000\}$. The error concentrates mainly around the cylinder, rendering the accurate prediction of the drag coefficient challenging. In Fig. 4d,h, the latent space of Latent-LED is compared with the transformation of the data to the latent space. The predictions stay close to the attractor even for a very large horizon ($T=20$). The Strouhal number St (defined in Supplementary equation (23)) describes the periodic vortex shedding at the wake of the cylinder. By estimating the dominant frequency of the latent state using a Fourier analysis, we find that LED reproduces exactly the St of the system dynamics for both $Re \in \{100, 1,000\}$ cases.

The LSTM propagator of LED is benchmarked against SINDy and RC in predicting the dynamics, starting from 10 initial conditions randomly sampled from the test data for a prediction horizon of $T=20$ (100 time-steps). The hyperparameters (reported in Supplementary Tables 15–17, along with the training times) are tuned on the basis of the MNAD on the validation data. The logarithm of the MNAD is given in Fig. 5a for $Re=100$ and Fig. 5e for $Re=1,000$. For the $Re=100$ case, the LSTM exhibits lower MNAD and lower variance than RC and SINDy. For the challenging $Re=1,000$ scenario, LSTM and RC exhibit lower MNAD than SINDy, with the LSTM being more robust (lower variance).

In the $Re=100$ case, Latent-LED recovers a periodic nonlinear mode in the latent space and can forecast the dynamics accurately, as illustrated in Fig. 4. In this case, approaches based on the Galerkin method or dynamic mode decomposition construct reduced-order models with six to eight degrees of freedom⁵³ that capture the most energetic spatiotemporal modes. In contrast, the latent space of LED in the $Re=100$ case has a dimensionality of $d_z=4$. In the challenging $Re=1,000$ scenario, LED with $d_z=10$ can capture accurately

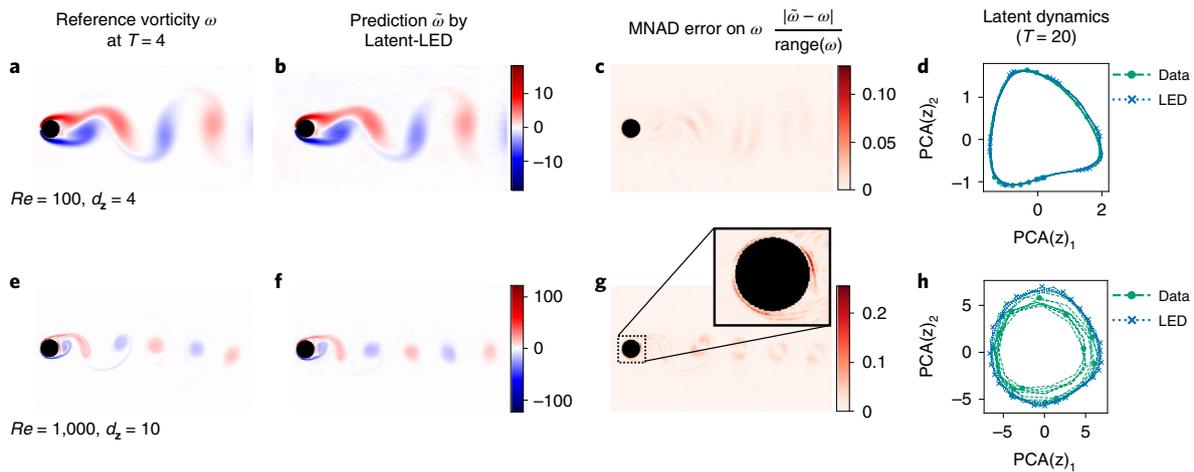


Fig. 4 | Iterative latent forecasting in flow past a cylinder. **a,e**, ω at lead time $T = 4$ for $Re = 100$ (**a**) and for $Re = 1,000$ (**e**). **b,f**, The vorticity field $\hat{\omega}$ predicted by Latent-LED at final time $T = 4$ for $Re = 100$ (**b**) and for $Re = 1,000$ (**f**). **c,g**, The MNAD for $Re = 100$ (**c**) and for $Re = 1,000$ (**g**). **d,h**, The LED latent-state dynamics compared against the attractor of the test data embedded in the latent space for $Re = 100$ (**d**) and for $Re = 1,000$ (**h**).

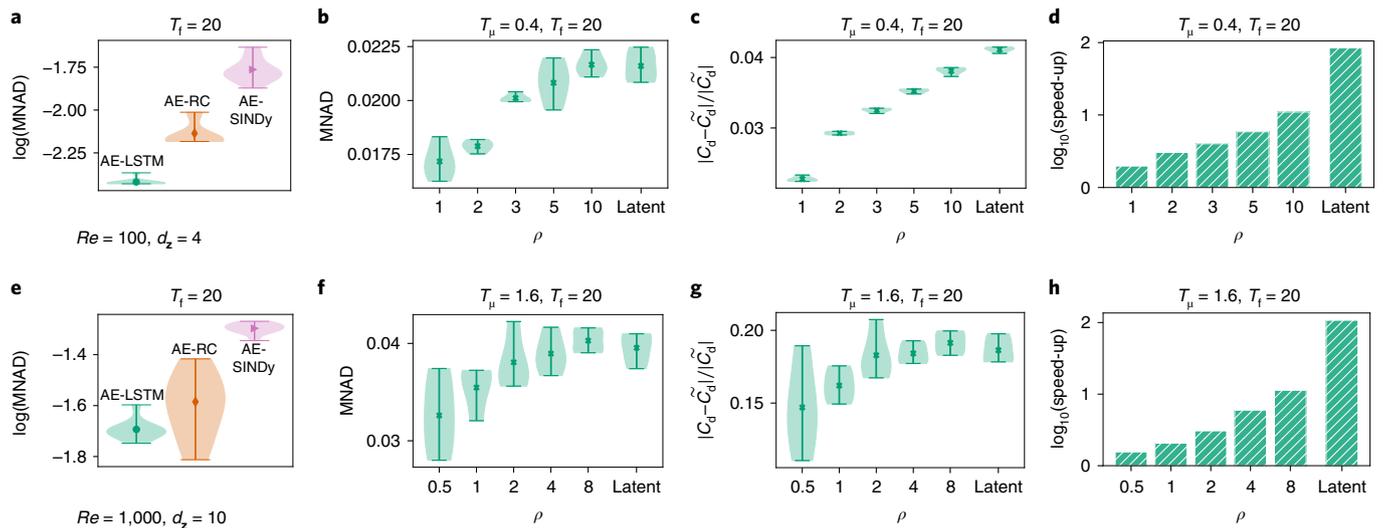


Fig. 5 | Multiscale LED in flow past a cylinder. **a,e**, Comparison of different macrodynamics propagators for $Re = 100$ (**a**) and for $Re = 1,000$ (**e**). **b,c**, The MNAD (**b**) and the relative error on the drag between predictions by LED and the reference data (**c**) as a function of ρ for $Re = 100$. **f,g**, The same for $Re = 1,000$. **d,h**, The speed-up of LED compared with the flow solver with respect to ρ for $Re = 100$ (**d**) and for $Re = 1,000$ (**h**).

the characteristic vortex street, and long-term dynamics. Reduced-order models for flows past a cylinder have been so far limited to periodic laminar flows of the order of $Re = 100$, whereas this study exceeds these flows by an order of magnitude.

Starting from four initial conditions randomly sampled from the test data, six LED variants (Latent-LED, Multiscale-LED with $T_\mu = 0.4$, $T_m \in \{0.4, 0.8, 1.2, 2, 4\}$ for $Re = 100$, and Latent-LED, Multiscale-LED with $T_\mu = 1.6$, $T_m \in \{0.8, 1.6, 3.2, 6.4, 12.8\}$ for $Re = 1,000$) are tested on predicting the dynamics of the flow up to $T_f = 20$, after $T_{\text{warm}} = 2$. The MNAD is plotted in Fig. 5b for $Re = 100$, and Fig. 5f for $Re = 1,000$. The speed-up is plotted in Fig. 5d for $Re = 100$, and Fig. 5h for $Re = 1,000$. The Latent-LED is two orders of magnitude faster than the flow solver while exhibiting MNAD errors of 0.02 and 0.04 for $Re = 100$ and $Re = 1,000$, respectively. By alternating between macro and micro, the error is reduced at the cost of decreased speed-up.

In Fig. 5c,g, the relative error on the drag coefficient C_d (defined in Supplementary equation (28)) is plotted as a function of ρ . Latent-LED exhibits a relative error of 0.04, which is reduced to approximately 0.02 for $\rho = 1$. For $Re = 1,000$, as we observe in Fig. 4, the prediction error of LED concentrates around the cylinder, which leads to an inaccurate computation of the drag. Even though Multiscale-LED reduces this error, it still remains of the order of 0.15.

Discussion

We have presented a novel framework for LED and accelerating the simulations of multiscale (stochastic or deterministic) complex dynamical systems. Our work relies on augmenting the equation-free formalism with state-of-the-art ML methods.

The LED framework is tested on a number of benchmark problems. In systems where evolving the high-dimensional state

dynamics is computationally expensive, LED accelerates the simulation by propagating on the latent space and upscaling to the high-dimensional states with the probabilistic, generative MD or deterministic convolutional decoder. This comes at the cost of training the networks, a process that is performed once offline. The trained model can forecast the dynamics starting from any arbitrary initial condition.

The efficiency of LED was evaluated in forecasting the FHN dynamics, achieving an order of magnitude lower approximation error compared with other equation-free approaches while being two orders of magnitude faster than the LB solver. We demonstrated that the proposed framework identifies the effective dynamics of the KS equation with $L=22$, capturing the long-term behaviour (climate dynamics), achieving a speed-up of $S \approx 100$. Furthermore, LED captures the long-term dynamics of a flow past a cylinder in $Re=100$ and $Re=1,000$ accurately while being two orders of magnitude faster than a flow solver. In Supplementary Information, we demonstrate that LED can unravel and forecast the stochastic collective dynamics of 1,000 particles following Brownian motion subject to advection and diffusion in the three-dimensional space (Supplementary Section 3A). In our recent work⁴⁰ (briefly described in Supplementary Section 3E), we show that LED can be applied to learn the stochastic dynamics of molecular systems. We note that the present method is readily applicable to all problems where equation-free, HMM and FLAVOR methodologies have been applied.

In summary, LED identifies and propagates the effective dynamics of dynamical systems with multiple spatiotemporal scales, providing considerable computational savings. Moreover, LED provides a systematic way of trading between speed-up and accuracy for a multiscale system by switching between the propagation of the latent dynamics and evolution of the original equations, iteratively correcting the statistical error at the cost of reduced speed-up.

The LED does not currently contain any mechanism to decide when to upscale the latent space dynamics. This is an active area of investigation. We do not expect LED to generalize to dynamical regions markedly different from those represented in the training data. Further research efforts will address this issue by adapting the training procedure.

The present methodology can be deployed in problems described by first principles as well as in problems where data are available only for either the macro- or microscale descriptions of the system. Ongoing work is concerned with the extension of the LED methodology to experimental settings and real-world data. We envision how experimental snapshots may inform the latent-space dynamics through AEs, but the challenge remains on how to reinitialize the experiments from the decoded microscale description. Moreover, real-world applications often entail noisy data. We note that LED can handle stochastic systems efficiently, as has been demonstrated in its application of accelerating molecular simulations⁴⁰.

In summary, LED creates unique algorithmic alloys between data-driven and first-principles models and opens new horizons for the accurate and efficient prediction of complex multiscale systems.

Methods

The methods for this paper are described in Supplementary Section I.

Data availability

All the data analysed in this paper were produced with open-source software described in the code availability statement. Reference data and the scripts used to produce the data figures, as well as instructions to launch training and inference (evaluation of trained models) scripts for LED, are available on the GitHub repository: <https://github.com/cselab/LED>.

Code availability

Simulations of the KS equation have been performed with a spectral fourth-order solver for stiff PDEs developed in Python. Simulation of the FHN equation has been performed with an LB method developed in Python. The LED software is

implemented in Python, utilizing the PyTorch library for the neural networks. All codes are made readily available in the GitHub repository: <https://github.com/cselab/LED>. Direct numerical simulations were performed with the flow solver CubismUP 2D: https://github.com/cselab/CubismUP_2D.

Received: 30 March 2021; Accepted: 16 February 2022;

Published online: 07 April 2022

References

- Rackovsky, S. & Scheraga, H. A. The structure of protein dynamic space. *Proc. Natl Acad. Sci. USA* **117**, 19938–19942 (2020).
- Gilmour, D., Rembold, M. & Leptin, M. From morphogen to morphogenesis and back. *Nature* **541**, 311–320 (2017).
- Robinson, P. A., Rennie, C. J., Rowe, D. L., O'Connor, S. C. & Gordon, E. Multiscale brain modelling. *Philos. Trans. R. Soc. B* **360**, 1043–1050 (2005).
- Council, N. R. *A National Strategy for Advancing Climate Modeling* (National Academies Press, 2012).
- Mahadevan, A. The impact of submesoscale physics on primary productivity of plankton. *Annu. Rev. Mar. Sci.* **8**, 161–184 (2016).
- Bellomo, N. & Dogbe, C. On the modeling of traffic and crowds: a survey of models, speculations, and perspectives. *SIAM Rev.* **53**, 409–463 (2011).
- Lee, E. H., Hsin, J., Sotomayor, M., Comellas, G. & Schulten, K. Discovery through the computational microscope. *Structure* **17**, 1295–1306 (2009).
- Springel, V. et al. Simulations of the formation, evolution and clustering of galaxies and quasars. *Nature* **435**, 629–636 (2005).
- Car, R. & Parrinello, M. Unified approach for molecular dynamics and density-functional theory. *Phys. Rev. Lett.* **55**, 2471–2474 (1985).
- Kevrekidis, I. G. et al. Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis. *Commun. Math. Sci.* **1**, 715–762 (2003).
- Weinan, E. & Engquist, B. et al. The heterogeneous multiscale methods. *Commun. Math. Sci.* **1**, 87–132 (2003).
- Kevrekidis, I. G., Gear, C. W. & Hummer, G. Equation-free: the computer-aided analysis of complex multiscale systems. *AIChE J.* **50**, 1346–1355 (2004).
- Laing, C. R., Frewen, T. & Kevrekidis, I. G. Reduced models for binocular rivalry. *J. Comput. Neurosci.* **28**, 459–476 (2010).
- Bar-Sinai, Y., Hoyer, S., Hickey, J. & Brenner, M. P. Learning data-driven discretizations for partial differential equations. *Proc. Natl Acad. Sci. USA* **116**, 15344–15349 (2019).
- Weinan, E., Li, X. & Vanden-Eijnden, E. in *Multiscale Modelling and Simulation* (eds Attinger, S. & Koumoutsakos, P.) 3–21 (Springer, 2004).
- Weinan, E., Engquist, B., Li, X., Ren, W. & Vanden-Eijnden, E. Heterogeneous multiscale methods: a review. *Commun. Comput. Phys.* **2**, 367–450 (2007).
- Tao, M., Owahdi, H. & Marsden, J. E. Nonintrusive and structure preserving multiscale integration of stiff ODEs, SDEs, and Hamiltonian systems with hidden slow dynamics via flow averaging. *Multiscale Model. Simul.* **8**, 1269–1324 (2010).
- Linot, A. J. & Graham, M. D. Deep learning to discover and predict dynamics on an inertial manifold. *Phys. Rev. E* **101**, 062209 (2020).
- Robinson, J. C. Inertial manifolds for the Kuramoto–Sivashinsky equation. *Phys. Lett. A* **184**, 190–193 (1994).
- Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
- Brunton, S. L., Noack, B. R. & Koumoutsakos, P. Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52**, 477–508 (2019).
- Lusch, B., Kutz, J. N. & Brunton, S. L. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.* **9**, 4950 (2018).
- Geneva, N. & Zabarar, N. Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. *J. Comput. Phys.* **403**, 109056 (2020).
- Milano, M. & Koumoutsakos, P. Neural network modeling for near wall turbulent flow. *J. Comput. Phys.* **182**, 1–26 (2002).
- Wehmeyer, C. & Noé, F. Time-lagged autoencoders: deep learning of slow collective variables for molecular kinetics. *J. Chem. Phys.* **148**, 241703 (2018).
- Bhatia, H. et al. Machine-learning-based dynamic-importance sampling for adaptive multiscale simulations. *Nat. Mach. Intell.* **3**, 401–409 (2021).
- Chung, J. et al. A recurrent latent variable model for sequential data. *Adv. Neural Inf. Process. Syst.* **28**, 2980–2988 (2015).
- Vlachas, P. R. et al. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Netw.* **126**, 191–217 (2020).
- Gonzalez, F. J. & Balajewicz, M. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. Preprint at <https://arxiv.org/abs/1808.01346> (2018).
- Maulik, R., Lusch, B. & Balaprakash, P. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Phys. Fluids* **33**, 037106 (2021).

31. Hasegawa, K., Fukami, K., Murata, T. & Fukagata, K. Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes. *Theor. Comput. Fluid Dyn.* **34**, 367–383 (2020).
32. Lee, S., Kooshkbaghi, M., Spiliotis, K., Siettos, C. I. & Kevrekidis, I. G. Coarse-scale PDEs from fine-scale observations via machine learning. *Chaos* **30**, 013141 (2020).
33. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
34. Bishop, C. M. *Mixture Density Networks* Technical Report NCRG/97/004 (Neural Computing Research Group, Aston University, 1994).
35. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
36. Werbos, P. J. Generalization of backpropagation with application to a recurrent gas market model. *Neural Netw.* **1**, 339–356 (1988).
37. Hernández, C. X., Wayment-Steele, H. K., Sultan, M. M., Husic, B. E. & Pande, V. S. Variational encoding of complex dynamics. *Phys. Rev. E* **97**, 062412 (2018).
38. Sultan, M. M., Wayment-Steele, H. K. & Pande, V. S. Transferable neural networks for enhanced sampling of protein dynamics. *J. Chem. Theory Comput.* **14**, 1887–1894 (2018).
39. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proc.* (eds Bengio, Y. & LeCun, Y.) 1–15 (2015).
40. Vlachas, P. R., Zavadlav, J., Praprotnik, M. & Koumoutsakos, P. Accelerated simulations of molecular systems through learning of their effective dynamics. *J. Chem. Theory Comput.* **18**, 538–549 (2021).
41. FitzHugh, R. Impulses and physiological states in theoretical models of nerve membrane. *Biophys. J.* **1**, 445 (1961).
42. Nagumo, J., Arimoto, S. & Yoshizawa, S. An active pulse transmission line simulating nerve axon. *Proc. IRE* **50**, 2061–2070 (1962).
43. Karlin, I. V., Ansumali, S., Frouzakis, C. E. & Chikatamarla, S. S. Elements of the lattice Boltzmann method I: Linear advection equation. *Commun. Comput. Phys.* **1**, 616–655 (2006).
44. Pathak, J., Hunt, B., Girvan, M., Lu, Z. & Ott, E. Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. *Phys. Rev. Lett.* **120**, 024102 (2018).
45. Brunton, S. L., Proctor, J. L. & Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **113**, 3932–3937 (2016).
46. Kuramoto, Y. Diffusion-induced chaos in reaction systems. *Prog. Theor. Phys. Suppl.* **64**, 346–367 (1978).
47. Sivashinsky, G. I. Nonlinear analysis of hydrodynamic instability in laminar flames—I. Derivation of basic equations. *Acta Astronaut.* **4**, 1177–1206 (1977).
48. Cvitanović, P., Davidchack, R. L. & Siminos, E. On the state space geometry of the Kuramoto–Sivashinsky flow in a periodic domain. *SIAM J. Appl. Dyn. Syst.* **9**, 1–33 (2010).
49. Kassam, A. & Trefethen, L. Fourth-order time-stepping for stiff PDEs. *SIAM J. Sci. Comput.* **26**, 1214–1233 (2005).
50. Zdravkovich, M. *Flow Around Circular Cylinders Volume 1: Fundamentals* (Oxford University Press, 1997).
51. Rossinelli, D. et al. MRAG-I2D: multi-resolution adapted grids for remeshed vortex methods on multicore architectures. *J. Comput. Phys.* **288**, 1–18 (2015).
52. Bost, C., Cottet, G.-H. & Maitre, E. Convergence analysis of a penalization method for the three-dimensional motion of a rigid body in an incompressible viscous fluid. *SIAM J. Numer. Anal.* **48**, 1313–1337 (2010).
53. Taira, K. et al. Modal analysis of fluid flows: applications and outlook. *AIAA J.* **58**, 998–1022 (2020).

Acknowledgements

We thank N.Kallikounis (ETH Zurich) for helpful discussions on the LB method, P. Weber and M. Chatzimanolakis (ETH Zurich) for help with the simulations of the flow past a cylinder and Y. Kevrekidis (Johns Hopkins University) and K. Spiliotis (University of Rostock) for providing code to reproduce data for the FHN equation. We acknowledge the infrastructure and support of CSCS, providing the necessary computational resources under project s929.

Author contributions

P.K. conceived the project; P.R.V., G.A., C.U. and P.K. designed and performed research; P.R.V. and G.A. contributed new analytic tools; P.R.V., G.A. and P.K. analysed data and P.R.V., G.A. and P.K. wrote the paper.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s42256-022-00464-w>.

Correspondence and requests for materials should be addressed to Petros Koumoutsakos.

Peer review information *Nature Machine Intelligence* thanks Harsh Bhatia and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature Limited 2022