Multiscale Neural Operators: Learning Fast and Grid-independent PDE Solvers

Anonymous Authors[1]

Abstract

Numerical simulations in climate, chemistry, or astrophysics are computationally too expensive for uncertainty quantification or parameter-exploration at high-resolution. Reduced-order or surrogate models are multiple orders of magnitude faster, but traditional surrogates are inflexible or inaccurate and pure machine learning (ML)-based surrogates too data-hungry. We propose a hybrid, flexible surrogate model that exploits known physics for simulating large-scale dynamics and limits learning to the hard-to-model term, which is called parametrization or closure and captures the effect of fine- onto large-scale dynamics. Leveraging neural operators, we are the first to learn grid-independent, non-local, and flexible parametrizations. Our *multiscale neural operator* is motivated by a rich literature in multiscale modeling, has quasilinear runtime complexity, is more accurate or flexible than state-of-the-art parametrizations and demonstrated on the chaotic equation multiscale Lorenz96.

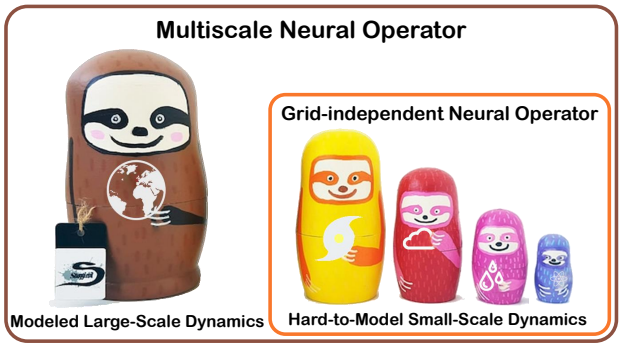


Figure 1: **Multiscale neural operator.** Explicitly modeling all scales of Earth's weather is too expensive for traditional and learning-based solvers [97]. Our multiscale neural operator dramatically reduces the computational cost by modeling the large-scale explicitly and learning the effect of fine- onto large-scale dynamics; such as turbulence slowing down a river stream. We embed grid-independent neural operators in the large-scale physical simulations as "parametrizations", conceptually similar to Matryoshka dolls. Image based on [119]

1. Introduction

Climate change increases the likelihood of storms, floods, wildfires, heat waves, biodiversity loss and air pollution [57]. Decision-makers rely on climate models to understand and plan for changes in climate, but current climate models are computationally too expensive: as a result, they are hard to access, cannot predict local changes ($< 10km$), fail to resolve local extremes (e.g., rainfall), and do not reliably quantify uncertainties [97]. For example, running a global climate model at $1km$ resolution can take ten days on a $4888\times$GPU node supercomputer, consuming the same electricity as a coal power plants generates in one hour [45]. Similarly, in molecular dynamics [3], chemistry [4], biology [139], energy [143], astrophysics or fluids [41], scientific progress is hindered by the computational cost of solving partial differential equations (PDEs) at high-resolution [63]. We are proposing the first PDE surrogate that quickly computes approximate solutions via correcting known large-scale simulations with learned, grid-independent, non-local parametrizations.

Surrogate models are fast, reduced-order, and lightweight copies of numerical simulations [107] and of significant interest in physics-informed machine learning [67, 114, 64, 47]. Machine learning (ML)-based surrogates have simulated PDEs up to $1-3$ order of magnitude faster than traditional numerical solvers and are more flexible and accurate than traditional surrogate models [63]. However, pure ML-based surrogates are too data-hungry [113]; so, hybrid ML-physics models are created, for example, via incorporating known symmetries [21, 3] or equations [133]. Most hybrid models represent the solution at the highest possible resolution, which becomes computationally infeasible in multiscale or very high-resolution physics; even with optimal runtime [103, 104].

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

As depicted in Figures 1 and 2, we simulate multiscale physics by running easy-to-acces large-scale models and focusing learning on the challenging task: *How can we model the influence of fine- onto large-scale dynamics, i.e., what is the subgrid parametrization term?* The lack of accuracy in current subgrid parametrizations, also called closure or residual terms, is one of the major sources of uncertainty in multiscale systems, such as turbulence or climate [97, 48]. Learning subgrid parametrizations can be combined with incorporating equations as soft [109] or hard [8] constraints. Various works learn subgrid parametrizations, but are either inaccurate, hard to share or inflexible because they are local [48], grid-dependent [73], or domain-specific [5], respectively as detailed in Section 2. We are the first to formulate the parametrization problem as learning neural operators [2] to represent non-local, flexible, and grid-independent parametrizations.

We propose, *multiscale neural operator* (MNO), a novel learning-based PDE surrogate for multiscale physics with the key contributions:

- A learning-based multiscale PDE surrogate that has quasilinear runtime complexity, leverages known large-scale physics, is grid-independent, flexible, and does not require autodifferentiable solvers.
- The first surrogate to approximate grid-independent, non-local parametrizations via neural operators
- Demonstration of the surrogate on the chaotic, coupled, multiscale PDE: multiscale Lorenz96

2. Related works

We embed our work in the broader field of physics-informed machine learning and surrogate modeling. We propose the first surrogate that corrects a coarse-grained simulation via learned, grid-independent, non-local parameterizations.

Direct numerical simulation. Despite significant progress in simulating physics numerically it remains prohibitively expensive to repeatedly solve high-dimensional partial differential equations (PDEs) [63]. For example, finite difference, element, volume, and (pseudo-)spectral methods have to be re-run for every choice of initial or boundary condition, grid, or parameters [43, 15]. The issue arises if the chosen method does not have optimal runtime, i.e., does not scale linearly with the number of grid points, which renders it infeasibly expensive for calculating ensembles [15]. Select methods have optimal or close-to-optimal runtime, e.g., quasi-linear $O(N \log N)$, and outperform machine learning-based methods in runtime and accuracy, but their implementation often requires significant problem-specific adaptations; for example multigrid [20] or spectral methods [15]. We acknowledge the existence of impressive resarch directions towards optimal and flexible non-ML solvers, such as the spectral solver called Dedalus [23], but advocate to simultaneously explore easy-to-adapt ML methods to create fast, accurate, and flexible surrogate models.

Surrogate modeling. Surrogate models are approximations, lightweight copies, or reduced-order models of PDE solutions, often fit to data, and used for parameter exploration or uncertainty quantificiation [118, 107]. Surrogate models via SVD/POD [31], Eigendecompositions/KLE [46], Koopman operators/DMD [135], take simplying assumptions to the dynamics, e.g., linearizing the equations, which can break down in high-dimensional or nonlinear regimes [107]. Our work leverages the expressiveness of neural operators as universal approximations [35] to learn fast high-dimensional surrogates that are accurate in nonlinear regimes [87, 141, 38, 93]. **Pure ML-based** surrogate models have shown impressive sucess in approximating dynamical systems from ground-truth simulation data – for example with neural ODEs [108, 34, 55], GNNs [16, 25], CNNs [121], neural operators [76, 2, 102, 86, 60], RNNs [62, 113], GPs [29], reservoir computing [100, 93], or transformers [32] – but, without incorporating physical knowlege become data-hungry and poor at generalization [63, 9].

Physics-informed machine learning. Two main approaches of incorporating physical knowledge into ML systems is via known symmetries [21] or equations [63]. Our approach leverages known equations for computing a coarse-grid prior; which is complementary to using known equations as soft [109, 74, 142, 137, 144, 139] or hard constraints [50, 89, 8, 39, 7, 61] as these methods can still be used to constrain the learned parametrization. In terms of symmetry, our approach exploits translational equivariance via Fourier transformations [76], but can be extended to other frameworks that exploit in- or equivariance of PDEs [95] to rotational [44, 124], Galilean [136, 105], scale [9], translational [123], reflectional [37] or permutational [145] transformations.

The field of physics-informed machine learning is very broad, as reviewed most recently in [133] and [63, 28, 65]. We focus on the task of learning fast and accurate surrogate models of fine-scale models when a fast and approximate coarse-grained simulation is availabe. This task differs from other interesting research areas in equation discovery or symbolic regression [22, 82, 83, 80, 106], downscaling or superresolution [138, 13, 72, 122, 128, 51], design space exploration or data synthesis [36, 30], controls [11] or interpretability [126, 90]. Our work is complementary to data assimilation or parameter calibra-

tion [58, 59, 66, 143, 14] which fit to observational data instead of models and differs from inverse modeling and parameter estimation [99, 53, 140, 81] which fit parametrizations that are independent of the previous state.

Correcting coarse-grid simulations via parametrizations. Problems with large domains are often solved via multiscale methods [103]. Multiscale methods simulate the dynamics on a coarse-grid and capture the effects of small-scale dynamics that occur within a grid cell via additive terms, called subgrid parametrizations, closures, or residuals [103, 91]. Existing subgrid parametrizations for many equations are still inaccurate [131] and ML outperformed them by learning parametrizations directly from high-resolution simulations; for example in turbulence [41], climate [48], chemistry [54], biology [104], materials [79], or hydrology [6]. The majority of ML-based parametrizations, however, is local [48, 94, 17, 18, 19, 141, 26, 6, 54, 79, 105, 78, 99, 136, 110], i.e., the in- and output are variables of single grid points, which assumes perfect scale separation, for example, in isotropic homogeneous turbulent flows [96]. However, local parametrizations are inaccurate; for example in the case of anisotropic nonhomogeneous dynamics [96, 129], for correcting global error for coarse spectral discretizations [15], or in large-scale climate models [40, 100]. More recent works propose non-local parametrizations, but their formulations either rely on a fixed-resolution grid [129, 12, 73, 33], an autodifferentiable solver [127, 117], or are formulated for a specific domain [5]. A single work proposes non-local and grid-independent parametrizations [101], but requires the explicit representation of a high-resolution state which is computationally infeasible for large domains, such as in climate modeling. We are the first to propose grid-independent and non-local parametrizations via neural operators to create fast and accurate surrogate models of fine-scale simulations.

Neural operators for grid-independent, non-local parametrizations. Most current learning-based non-local parametrizations rely on FCNNs, CNNs [73], or RNNs [33], which are mappings between finite-dimensional spaces and thus grid-dependent. In comparison, neural operators learn mappings in between infinite-dimensional function spaces [71] such as the Laplacian, Hessian, gradient, or Jacobian. Typically, neural operators lift the input into a grid-independent state such as Fourier [76], Eigen- [10], graph kernel [75, 2] or other latent [86] modes and learn weights in the lifted domain. We are the first to formulate neural operators for learning parametrizations.

3. Approach

We propose *multiscale neural operator* (MNO): a surrogate model with quasilinear runtime complexity that exploits know coarse-grained simulations and learns a grid-independent, non-local parametrization.

3.1. Multiscale neural operator

Partial differential equations. We focus on partial differential equations (PDEs) that can be written as initial value problem (IVP) via the method of lines [134]. The PDEs in focus have one temporal dimension, $t \in [0, T] =: D_t$, and (multiple) spatial dimensions, $x = [x_1, ..., x_d]^T \in D_x$, and can be written in the iterative, explicit, symbolic form [43]:

$$\begin{aligned} \frac{\delta u}{\delta t} - \mathcal{N}(u) = 0 \text{ with } t, x \in [0, T] \times D_x \\ u(0, x) = u^0(x),\ \mathcal{B}[u](t, x) = 0 \text{ with } x \in D_x, \\ (t, x) \in [0, T] \times \delta D_x \end{aligned} \tag{1}$$

In our case, the (non-)linear operator, $\mathcal{N}$, encodes the **known** physical equations; for example a combination of Laplacian, integral, differential, etc. operators. Further, $u : D_t \times D_x \to D_u$ is the solution to the initial values, $u^0 : D_x \to D_u$, and Dirichlet, $\mathcal{B}_D[u] = u - b_D$, or Neumann boundary conditions, $\mathcal{B}_N[u] = n^T \delta_x u - b_N$, with outward facing normal on the boundary, $n \perp \delta B$.

Scale separation. We transfer a concept from the rich and mathematical literature in multiscale modeling [103] to consider a filter kernel operator, $\mathcal{G}*$, that creates the large-scale solution, $\bar{u}(x) = u(x) + u'(x)$, where u' are the small-scale deviations and $\bar{\cdot}$ denotes the filtered variable, $\bar{\phi}(x) = \mathcal{G} * \phi = \int_{D_x} G(x, x')\phi(x')dx'$. Assuming the kernel, G, preserves constant fields, $\bar{a} = a$, commutes with differentiation, $[\mathcal{G}*, \frac{\delta}{\delta s}], s = x, t$, is linear, $\overline{\phi + \psi} = \bar{\phi} + \bar{\psi}$ [96], we can rewrite (1) to:

$$\begin{aligned} \mathcal{G} * \frac{\delta u}{\delta t} = \frac{\delta \bar{u}}{\delta t} &= \mathcal{G} * \mathcal{N}(u) \\ &= \mathcal{N}(\bar{u}) + [\mathcal{G}*, \mathcal{N}](u) \end{aligned} \tag{2}$$

where $[\mathcal{G}*, \mathcal{N}](u) = \mathcal{G}*\mathcal{N}(u) - \mathcal{N}(\mathcal{G}*u)$ is the filter subgrid parametrization, closure term, or commutation error, i.e., the error introduced through propagating the coarse-grained solution.

Approximations of the subgrid parametrization as an operator that acts on $\bar{u}$ require significant domain expertise and are derived on a problem-specific basis. In the case of isotropic homogeneous turbulence, for example, the subgrid parametrization can be approximated as the spatial derivative of the subgrid stress tensor, $[\mathcal{G}*, \mathcal{N}](\bar{u})_{\text{turbulence}} \approx \frac{\delta \tau_{ij}}{\delta x_j} = \frac{\delta \overline{u_i' u_j'}}{\delta x_j}$ [96]. Many works approximate the subgrid stress tensor with physics-informed ML [105, 78, 99, 136], but are domain-specific, local, or require a differentiable

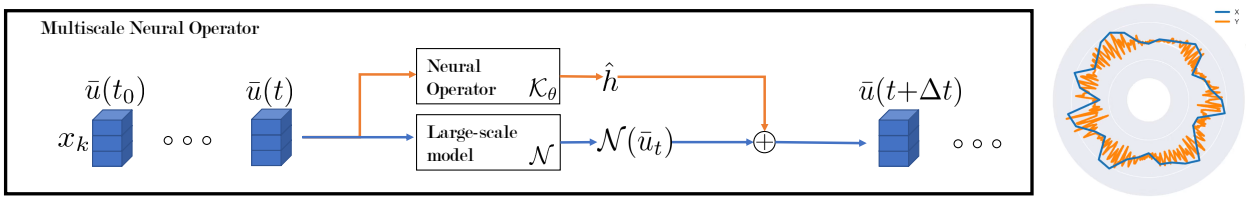


Figure 2: Left: **Model Architecture.** A physics-based model, $\mathcal{N}$, can quickly propagate the state, $\bar{u}_t$, at a large-scale, but will accumulate the error, $h = \overline{\mathcal{N}(u)} - \mathcal{N}\bar{u}$. A neural operator, $\mathcal{K}_\theta$, wraps the computational and implementation complexities of unmodeled fine-scale dynamics into a non-local and grid-independent term, $\hat{h}$, that iteratively corrects the large-scale model. Right: **Multiscale Lorenz96.** We demonstrate multiscale neural operator (MNO) on the multiscale Lorenz96 equation, a model for chaotic atmospheric dynamics. Image: [110]

solver or fixed-grid. We propose a general purpose method to approximating the subgrid parametrization, independent of the grid, domain, isotropy, and underlying solver.

Multiscale neural operator. We aim to approximate the filter commutation error, $[\mathcal{G}*, \mathcal{N}] \approx: h$, via learning a neural operator on high-resolution training data. Let $\mathcal{K}_\theta$ be a neural operator that approximates the commutation error:

$$[\mathcal{G}*, \mathcal{N}] \approx \mathcal{K}_\theta : \bar{U}(D_x; \mathbb{R}^{d_u}) \to H(D_x; \mathbb{R}^{d_u}) \quad (3)$$

where θ are the learned parameters and $\bar{U}, H$ are separable Banach spaces of all continuous functions taking values, $\mathbb{R}^{d_u}$, defined on the bounded, open set, $D_x \subset \mathbb{R}^{d_x}$, with norm $||f||_{\bar{U}} = ||f||_H = \max_{x \in D_x} |f(x)|$. We embed the neural operator as an autoregressive model with fixed time-discretization, Δt, such that the final *multiscale neural operator* (MNO) model is:

$$\bar{u}(t + \Delta t) = f(t, \bar{u}, \frac{\delta \bar{u}}{\delta x}, \frac{\delta^2 \bar{u}}{\delta x^2}, \dots) + \mathcal{K}_\theta(\bar{u}) \quad (4)$$

where $f(t, \bar{u}, \frac{\delta \bar{u}}{\delta x}, \frac{\delta^2 \bar{u}}{\delta x^2}) = \int_t^{t+\Delta t} \mathcal{N}(\bar{u}) d\tau$ is the known large-scale tendency, i.e. one-step solution. MNO is fit using MSE with the loss function:

$$L = \mathbb{E}_t \mathbb{E}_{\bar{u}|u(t) \sim p(t)} \left(\mathcal{L}(\mathcal{K}_\theta(\bar{u}(t)), [\mathcal{G}*, \mathcal{N}](u(t)) \right) \quad (5)$$

where the ground-truth data, $u(t) \sim p(t)$, is generated by integrating a high-resolution simulation with varying parameters, initial or boundary conditions and uniformly sampling time snippets according to the distribution $p(t)$. Similar to problems in superresolution, there exist multiple realizations of the learned commutation error, $[\mathcal{G}*, \mathcal{N}](\bar{u})$, for a given ground-truth, $[\mathcal{G}*, \mathcal{N}](u)$; using MSE will learn a smooth average and future work will explore adversarial losses [49] or an intersection between neural operators and normalizing flows [115] or diffusion-based models [120] to account for the stochasticity [132]. During training, the model input is generated via $\bar{u}(t) = \mathcal{G} * (u(t))$ and the target via

$$h_{\text{target}} = \overline{\mathcal{N}(u)} - \mathcal{N}(\bar{u}). \quad (6)$$

During inference MNO is initialized with a large-scale state and integrates the dynamics in time via coupling the neural operator and a large-scale simulation.

Our approach does not need access to the high-resolution simulator or equations; it only requires a precomputed high-resolution dataset, which are increasingly available [56, 24], and allows the user to incorporate existing easy-to-access solvers of large-scale equations. There is no requirement for the large-scale solver to be autodifferentiable which significantly simplifies the implementation for large-scale models, such as in climate. If desired, our loss function can easily be augmented with a physics-informed loss [109] on the large-scale dynamics or parametrization term.

Choice of neural operator. Our formulation is general enough to allow the use of many operators, such as Fourier [76], PCA-based [10], low-rank [69], Graph [75] operators, or DeepOnet [130, 86]. Because DeepONet [86] focuses on interpolation and assumes fixed-grid sensor data, we decided to modify Fourier Neural Operator (FNO) [76] for our purpose. FNO is a universal approximator of nonlinear operators [71, 35], grid-independent and can be formulated as autoregressive model [76]. As there exist significant knowledge on symmetries and conservation properties of the commutation error [96], MNO's explicit formulation increases interpretability and ease of incorporating symmetries and constraints. With FNO, we exploit approximate translational symmetries in the data and leave novel opportunities for neural operators that exploit the full range of known equi- and invariances of the subgrid parametrization term, such as Galilean invariance [105], for future work.

3.2. Illustration of MNO via multiscale Lorenz96

We illustrate the idea of MNO on a canonical model of atmospheric dynamics, the multiscale Lorenz96 equation [84, 125]. This PDE is multiscale, chaotic, time-

continuous, space-discretized, 2D (space+time), nonlinear, displayed in Figure 2-right and detailed in Appendix A.3. Most importantly, the large- and small-scale solutions, $X_k \in \mathbb{R}, Y_{j,k} \in \mathbb{R}\ \forall\ j \in \{0, ..., J\}, k \in \{0, ..., K\}$, demonstrate the *curse of dimensionality*: the number of the small-scale states grows exponentially with scale and explicit modeling becomes computationally expensive, for example, quadratic for two-scales: $O(N^2) = O(JK)$. The PDE writes:

$$\frac{\delta X_k}{\delta t} = X_{k-1}(X_{k+1} - X_{k-2}) - X_k + F - \frac{h_s c}{b} \sum_{j=0}^{J-1} Y_{j,k}(X_k),$$
$$\frac{\delta Y_{j,k}}{\delta t} = -cbY_{j+1,k}(Y_{j+2,k} - Y_{j-1,k}) - cY_{j,k} + \frac{h_s c}{b} X_k. \tag{7}$$

where F is the forcing, h_s the coupling strength, b the relative magnitude of scales, and c the evolution speed. With the multiscale framework from Section 3.1, we define:

$$u(x) = [X_0, Y_{0,0}, Y_{1,0}, ..., Y_{J,0}, X_1, Y_{0,1}, ... \\ , X_K, ..., Y_{J,K}]_x\ \forall x \in D_x = \{0, ..., K(J+1)\}$$
$$\mathcal{N}(u)(x) = \begin{cases} \frac{\delta X_k}{\delta t} & \text{if } x = k(J+1)\ \forall k \in \{0, \dots, K\} \\ \frac{\delta Y_{j,k}}{\delta t} & \text{otherwise,} \end{cases}$$
$$G(x, x') = \begin{cases} 1 \text{ if } x' = k(J+1)\ \forall k \in \{0, \dots, K\} \\ 0 \text{ otherwise,} \end{cases}$$

with the solution, u, operator, $\mathcal{N}$, and kernel, G.

MNO learns the parametrization term via a neural operator, $\mathcal{K}_\theta = \hat{h} \approx h$, and then models:

$$\frac{\delta \hat{X}_k}{\delta t} = \frac{\delta \overline{\hat{X}}_k}{\delta t} + \mathcal{K}_\theta(\hat{X}_{0:K})(k) \tag{8}$$

where the known large-scale dynamics are abbreviated with $\frac{\delta \overline{\hat{X}}_k}{\delta t} = \hat{X}_{k-1}(\hat{X}_{k+1} - \hat{X}_{k-2}) - \hat{X}_k + F$ and ground-truth parametrization is $h(x) = \{-\frac{h_s c}{b} \sum_{j=0}^{J-1} Y_{j,k}(X_k)$ if $x = k(J+1)\ \forall k \in \{0, \dots, K\}$ and 0 otherwise$\}$. See Appendix A.4 for all terms.

The parametrization, $\mathcal{K}_\theta$, accepts inputs that are sampled anywhere inside the spatial domain, which differs from previous local [110] or grid-dependent [33] Lorenz96 parametrizations.

We create the ground-truth data via randomly sampled initial conditions, periodic boundary conditions, and integrating the coupled equation with a 4th-order Runge-Kutta solver. After a Lyapunov timescale the state is independent of initial conditions and we extract 4K snippets with $T/\Delta t = 400$steps length for 1-step training. This model is run autoregressively on 1K test samples of length $T/\Delta t = 400$steps, which correspond to 10 Earth days, as detailed in Appendix A.3.

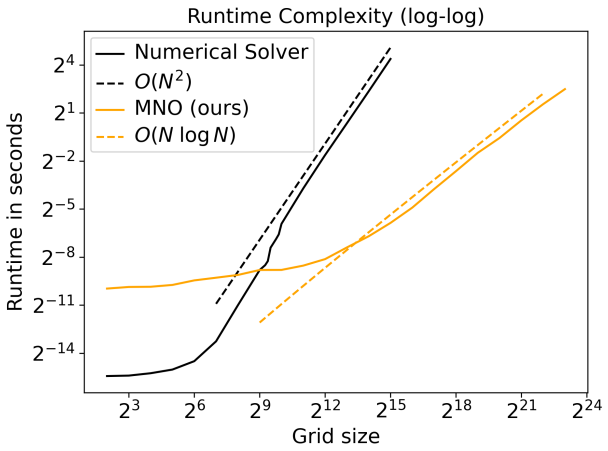


Figure 3: **MNO is faster than direct numerical simulation.** Our proposed multiscale neural operator (orange) can propagate multiscale PDE dynamics in quasilinear complexity, $O(N\ \log N)$. For a grid with $K = 2^{15}$, MNO is $\sim$ 1000-times faster than direct numerical simulation (black) which has quadratic complexity, $O(N^2)$

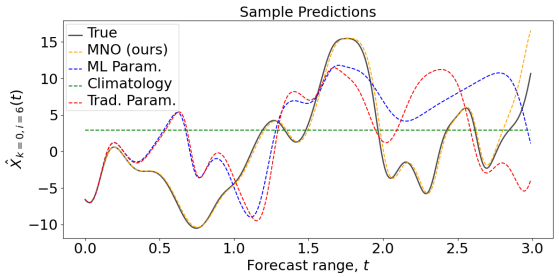
4. Results

Our results demonstrate that multiscale neural operator (MNO) is faster than direct numerical simulation, generates stable solutions, and is more accurate than current parametrizations. We now proceed to discussing each of these in more detail.

4.1. Runtime Complexity: MNO is faster than traditional PDE solvers

MNO (orange in Figure 3) has quasilinear, $O(N\ \log N)$, runtime complexity in the number of large-scale grid points, $N{=}K$, in the multiscale Lorenz96 equation. The runtime is dominated by a lifting operation, here a fast Fourier transform (FFT), which is necessary to learn spatial correlations in a grid-independent space. In comparison, the direct numerical simulation (black) has quadratic runtime complexity, $O(N^2)$, because of the explicit representation of $N^2{=}JK$ small-scale states. Both models are linear in time, $O(T)$. Local parametrizations can achieve optimal runtime, $O(N)$, but it is an open question if there exists a decomposition that replaces FFT to yield an optimal, non-local, grid-independent model.

We ran MNO up to a resolution of $K = 2^{24}$, which would equal $75cm/px$ in a global 1D (space) climate model and only took $\approx\ 2s$ on a single CPU. MNO is three orders of magnitude (1000-times) faster than DNS, at a resolution of $K = 2^{15}$ or $200m/px$. For 2D or 3D simulations the gains of using MNO vs. DNS are even higher with $O(N^2\ \log N)$ vs. $O(N^4)$ and $O(N^3\ \log N)$ vs. $O(N^6)$, respectively [68].



Method	RMSE
Climatology	6.902
Traditional parametrizations	2.326
ML-based parametrization [112]	2.053
MNO (ours)	0.5067

Figure 4: **Left: MNO is more accurate than traditional parametrizations.** A sample plot shows, that our proposed multiscale neural operator (yellow/orange-dotted) can accurately forecast the large-scale physics (black-solid), $X_{k=0}(t)$. In comparison, ML-based blue-dotted) and traditional (red-dotted) parametrizations quickly start to diverge. Note that the system is chaotic and small deviations are rapidly amplified; even inserting the exact parametrizations in float32 instead of float64 quickly diverges. **Right: Accuracy.** MNO is more accurate than traditional parametrizations as measured by the root mean-square error (RMSE).

The runtimes have been calculated by choosing the best of 1-100k runs depending on grid size on a single-threaded Intel Xeon Gold 6248 CPU@2.50GHz with 164Gb RAM. We time a one step update which, for DNS, is the calculation of (7) and for MNO the calculation of (8), i.e., the sum of a large-scale step and a pass through the neural operator.

In Figure 3, MNO and DNS plateau at low-resolution ($K < 2^9$), because runtime measurement is dominated by grid-independent operations. DNS plateaus at a lower runtime, because MNO contains several fixed-cost matrix transformations. The runtime of DNS has a slight discontinuity at $K \approx 2^9$ due to extending from cache to RAM memory. We focus on a runtime comparison, but MNO also has significant savings in memory: representing the state at $K = 2^{17}$ in double precision occupies 64GB RAM for DNS and 0.5MB for MNO.

4.2. MNO is more accurate than traditional parametrizations

Figure 4-left shows a forecasted trajectory of a sample at the left boundary, $k = 0$, where MNO (orange-dotted) accurately forecasts the large-scale dynamics, $X_0(t)$, (black-solid) while current ML-based (blue-dotted) [48] and traditional parametrizations (red-dotted) quickly diverge. The quantitive comparison of RMSE and a mean/std plot Figure 7 over $1K$ samples and 200steps or 10days ($\Delta t = 0.005 = 36$min) confirms that MNO is the most accurate in comparison to ML-based parametrizations, traditional parametrizations, and a mean forecast (climatology). Note, the difficulty of the task: when forecasting *chaotic* dynamics even numerical errors rapidly amplify [96].

ML-based parametrizations is a state-of-the-art (SoA) model in learning parametrizations and trains a ResNet to forecast a local, grid-independent parametrization, $h_k = \text{NN}(X_k)$, similar to [48]. The **traditional parametrizations** (trad. param.) are often used in practice and use linear regression to learn a local, grid-independent parametrization [91]. It was suggested that multiscale Lorenz96 is too easy as a test-case for comparing offline models because traditional parametrizations already perform well [111], but the significant difference between MNO and Trad. Params. during online evaluation suggests otherwise. The **climatology** forecasts the mean of the training dataset, $X_k(t) = 1/T \sum_{t=0}^{T} 1/N \sum_{i=0}^{N} X_{k,i}(t)$. The full list of hyperparameters and model parameters can be found in Appendix A.5.2. For fairness, we only compare against grid-independent methods that do not require an autodifferentiable solver; models with soft or hard constraints, e.g., PINNs [109] or DC3 [39], are complementary to MNO.

4.3. MNO is stable

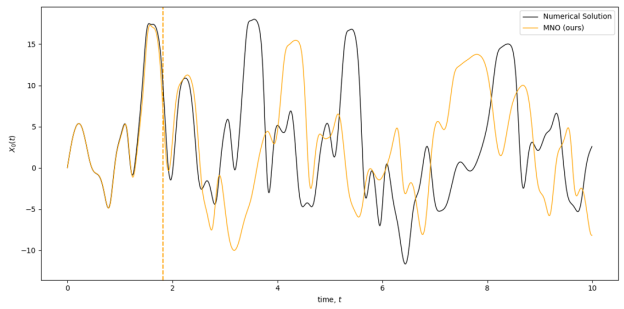
Figure 5 shows that predicting large-scale dynamics with MNO is stable. We first plot a randomly selected sample of the first large-scale state, $X_{k=0}(t)$ (left-black), to illustrate that the prediction is bounded. The MNO prediction (left-yellow) follows the ground-truth up to an approximate horizon of, $t = 1.8$ or 9 days, then diverges from the ground-truth solution, but stays within the bounds of the ground-truth prediction and does not diverge to infinity. The RMSE over time in Figure 5 shows that MNO (yellow) is approximately more accurate than current ML-based (blue) and traditional (red) parametrizations for $\approx 100\%$-longer time, measuring the time to intersect with climatology. Despite the difficulty in predicting chaotic dynamics, the RMSE of MNO reaches a plateau, which is slightly above the optimal plateau given by the climatology (black).

The RMSE over time is calculated as:

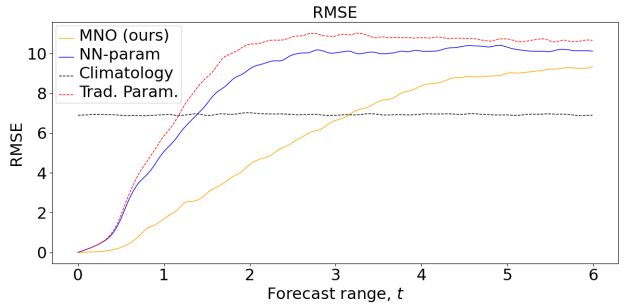
$$\text{RMSE}(t) = \frac{1}{K}\sum_{k=0}^{K}\sqrt{(\frac{1}{N}\sum_{i=0}^{N}(\hat{X}_{k,i}(t) - X_{k,i}(t))^2)}. \tag{9}$$

5. Limitations and Future Work

We demonstrated the accuracy, speed, and stability of MNO on the chaotic multiscale Lorenz96 equation. Future work, can extend MNO towards higher-dimensional or



(a) Long-term sample forecast, $X_0(t)$



(b) Error over time

Figure 5: **MNO is stable.** MNO can propagate a sample state, $X_{k=0}(t)$, over a long time horizon without diverging to infinity (left). The right plot shows that the RMSE of MNO plateaus for long-term forecasts, further confirming stability. Further, MNO (yellow) maintains accuracy longer than ML-based parametrizations (blue) and a climatology (black).

time-irregular systems and further integrate symmetries or constraints:

The results show promise to extend MNO to higher-dimensional, chaotic, multiscale, multiphysics problems and improve parametrizations in anisotropic turbulence predictions [96], Rayleigh-Bénard Convection (see Appenix A.1.) or clouds of global atmospheric models [129, 97]. Lightweight climate surrogate models could dramatically improve uncertainties [88] or decision-exploration [116] in climate.

MNO is grid-independent in space but not in time which could be alleviated via integrations with Neural ODEs [34]. MNO is a myopic model which might suffice for chaotic dynamics [77], but could be combined with LSTMs [92] or reservoir computing [100] to contain a memory. Further, we leveraged global Fourier decompositions to exploit grid-independent periodic spatial correlations, but future work could also capture local discontinuities, e.g., along coastlines [60] with multiwavelets [52], or incorporate non-periodic boundaries via Chebyshev polynomials.

Lastly, MNO can be combined with Geometric deep learning, PINNs, or hard constraint models. This avenue of research is particularly exciting with MNO as there exist many known symmetries for the paramtrization term [105].

6. Conclusion

We proposed a hybrid physics-ML surrogate of multiscale PDEs that is quasilinear, accurate, and stable. The surrogate limits learning to the influence of fine- onto large-scale dynamics and is the first to use neural operators for a grid-independent, non-local corrective term of large-scale simulations. We demonstrated that multiscale neural operator (MNO) is faster than direct numerical simulation ($O(N \log N)$ vs. $O(N^2)$ and more accurate ($\approx 100\%$ longer prediction horizon) than state-of-the-art parametrizations on the chaotic, multiscale equations multiscale Lorenz96. With the dramatic reduction in runtime, MNO could enable rapid parameter exploration and robust uncertainty quantification in complex climate models.

7. Ethical and Societal Implications of the proposed work

Climate change is the defining challenge of our time. Environmental disasters will become more frequent: from storms, floods, wildfires and heat waves to biodiversity loss and air pollution [57]. The impacts of climate change will not only be severe, but also unjustly distributed: island states, minority populations, and the Global South are already facing the most severe consequences of climate change, while the Global North is responsible for the most emissions since the industrial revolution [1]. Decision-makers require better tools to understand and plan for changes in climate and limit the economic, human, and environmental impact [97]. We propose a faster differential equation solver to improve the underlying climate models. Because fast differential equations can be leveraged in ethically questionable fields, such as missile development, we are applying our methods to climate modeling to demonstrate our work towards positive impact.

References

[1] Glenn Althor, James E. M. Watson, and Richard A. Fuller. Global mismatch between greenhouse gas emissions and the burden of climate change. *Scientific Reports*, 6, 2016.

[2] Anima Anandkumar, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Nikola Kovachki, Zongyi Li, Burigede Liu, and Andrew Stuart. Neural operator: Graph kernel network for partial differential equations. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.

[3] Simon Batzner, Albert Musaelian, Lixin Sun, Mario

Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13, 2022.

[4] Jörg Behler. Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations. *Phys. Chem. Chem. Phys.*, 13, 2011.

[5] Parrinello M. Behler J. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys Rev Lett.*, 14, 2007.

[6] Andrew Bennett and Bart Nijssen. Deep learned process parameterizations provide better representations of turbulent heat fluxes in hydrologic models. *Earth and Space Science Open Archive*, page 20, 2020.

[7] Tom Beucler, Stephan Rasp, Michael Pritchard, and Pierre Gentine. Achieving Conservation of Energy in Neural Network Emulators for Climate Modeling. jun 2019.

[8] Tom Beucler, Michael Pritchard, Stephan Rasp, Jordan Ott, Pierre Baldi, and Pierre Gentine. Enforcing analytic constraints in neural networks emulating physical systems. *Phys. Rev. Lett.*, 126:098302, Mar 2021.

[9] Tom Beucler, Michael S. Pritchard, Janni Yuval, Ankitesh Gupta, Liran Peng, Stephan Rasp, Fiaz Ahmed, Paul A. O'Gorman, J. David Neelin, Nicholas J. Lutsko, and Pierre Gentine. Climate-invariant machine learning. *CoRR*, 2021.

[10] Kaushik Bhattacharya, Bamdad Hosseini, Nikola B. Kovachki, and Andrew M. Stuart. Model reduction and neural networks for parametric pdes, 2020.

[11] Katharina Bieker, Sebastian Peitz, Steven L. Brunton, J. Nathan Kutz, and Michael Dellnitz. Deep model predictive flow control with limited sensor data and online learning, 2020.

[12] Sindre Stenen Blakseth, Adil Rasheed, Trond Kvamsdal, and Omer San. Deep neural network enabled corrective source term approach to hybrid analysis and modeling. *Neural Networks*, 146:181–199, 2022.

[13] Mathis Bode, Michael Gauding, Zeyu Lian, Dominik Denker, Marco Davidovic, Konstantin Kleinheinz, Jenia Jitsev, and Heinz Pitsch. Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows. *Proceedings of the Combustion Institute*, 38 (2):2617–2625, 2021.

[14] Massimo Bonavita and Patrick Laloyaux. Machine learning for model error inference and correction. *Journal of Advances in Modeling Earth Systems*, 12 (12), 2020.

[15] J.P. Boyd. *Chebyshev and Fourier Spectral Methods: Second Revised Edition*. Dover Books on Mathematics. Dover Publications, 2013.

[16] Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations (ICLR)*, 2022.

[17] N. D. Brenowitz and C. S. Bretherton. Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, 45(12): 6289–6298, 2018.

[18] Noah D. Brenowitz, Tom Beucler, Michael Pritchard, and Christopher S. Bretherton. Interpreting and stabilizing machine-learning parametrizations of convection. *Journal of the Atmospheric Sciences*, 77(12): 4357 – 4375, 2020.

[19] Christopher S. Bretherton, Brian Henn, Anna Kwa, Noah D. Brenowitz, Oliver Watt-Meyer, Jeremy McGibbon, W. Andre Perkins, Spencer K. Clark, and Lucas Harris. Correcting coarse-grid weather and climate models by machine learning from global storm-resolving simulations. *Journal of Advances in Modeling Earth Systems*, 14(2), 2022.

[20] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A Multigrid Tutorial (2nd Ed.)*. Society for Industrial and Applied Mathematics, USA, 2000. ISBN 0898714621.

[21] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, 2021.

[22] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[23] Keaton J. Burns, Geoffrey M. Vasil, Jeffrey S. Oishi, Daniel Lecoanet, and Benjamin P. Brown. Dedalus: A flexible framework for numerical simulations with spectral methods. *Physical Review Research*, 2(2), April 2020.

[24] Randal Burns, Gregory Eyink, Charles Meneveau, Alex Szalay, Tamer Zaki, Ethan Vishniac, Akshat Gupta, Mengze Wang, Yue Hao, Zhao Wu, and Gerard Lemson. Johns hopkins turbulence database, 2022. last accessed May, 2022.

[25] Salva Rühling Cachay, Emma Erickson, Arthur Fender C. Bucker, Ernest Pokropek, Willa Potosnak, Suyash Bire, Salomey Osei, and Björn Lütjens. The world as a graph: Improving el niño forecasts with graph neural networks, 2021.

[26] Salva Rühling Cachay, Venkatesh Ramesh, Jason N. S. Cole, Howard Barker, and David Rolnick. ClimART: A benchmark dataset for emulating atmospheric radiative transfer in weather and climate models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[27] J. Campin, C. Hill, H. Jones, and J. Marshall. Super-parameterization in ocean modeling: Application to deep convection. *Ocean Modelling*, 36:90–101, 2011.

[28] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91, Dec 2019.

[29] S. Chakraborty, S. Adhikari, and R. Ganguli. The role of surrogate models in the development of digital twins of dynamic systems. *Applied Mathematical Modelling*, 90:662–681, 2021.

[30] Shing Chan and Ahmed H. Elsheikh. Parametric generation of conditional geological realizations using generative neural networks, 2019.

[31] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7): 808–817, 2000.

[32] Ashesh Chattopadhyay, Mustafa Mustafa, Pedram Hassanzadeh, and Karthik Kashinath. Deep spatial transformers for autoregressive data-driven forecasting of geophysical turbulence. In *Proceedings of the 10th International Conference on Climate Informatics*, CI2020, page 106–112, New York, NY, USA, 2020. Association for Computing Machinery.

[33] Ashesh Chattopadhyay, Adam Subel, and Pedram Hassanzadeh. Data-driven super-parameterization using deep learning: Experimentation with multiscale lorenz 96 systems and transfer learning. *Journal of Advances in Modeling Earth Systems*, 12(11), 2020.

[34] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems 31*, pages 6571–6583. Curran Associates, Inc., 2018.

[35] Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.

[36] Wei Chen and Faez Ahmed. Padgan: Learning to generate high-quality novel designs. *Journal of Mechanical Design*, 143(3).

[37] Taco S. Cohen and Max Welling. Steerable cnns. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France*, 2017.

[38] Alberto Costa Nogueira, João Lucas de Sousa Almeida, Guillaume Auger, and Campbell D. Watson. Reduced order modeling of dynamical systems using artificial neural networks applied to water circulation. In Heike Jagode, Hartwig Anzt, Guido Juckeland, and Hatem Ltaief, editors, *High Performance Computing*, pages 116–136, Cham, 2020. Springer International Publishing.

[39] Priya L. Donti, David Rolnick, and J Zico Kolter. DC3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations (ICLR)*, 2021.

[40] P. D. Dueben and P. Bauer. Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11(10):3999–4009, 2018.

[41] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51(1):357–377, 2019.

[42] A. Dutt and V. Rokhlin. Fast fourier transforms for nonequispaced data. *SIAM Journal on Scientific Computing*, 14(6):1368–1393, 1993.

[43] S.J. Farlow. *Partial Differential Equations for Scientists and Engineers*. Dover books on advanced mathematics. Dover Publications, 1993.

[44] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020.

[45] O. Fuhrer, T. Chadha, T. Hoefler, G. Kwasniewski, X. Lapillonne, D. Leutwyler, D. Lüthi, C. Osuna, C. Schär, T. C. Schulthess, and H. Vogt. Near-global climate simulation at 1 km resolution: establishing a performance baseline on 4888 gpus with cosmo 5.0. *Geosci. Model Dev.*, 11:1665 – 1681, 2018.

[46] K. Fukunaga and W.L.G. Koontz. Application of the karhunen-loève expansion to feature selection and ordering. *IEEE Transactions on Computers*, C-19(4): 311–318, 1970.

[47] A. R. Ganguly, E. A. Kodra, A. Agrawal, A. Banerjee, S. Boriah, Sn. Chatterjee, So. Chatterjee, A. Choudhary, D. Das, J. Faghmous, P. Ganguli, S. Ghosh, K. Hayhoe, C. Hays, W. Hendrix, Q. Fu, J. Kawale, D. Kumar, V. Kumar, W. Liao, S. Liess, R. Mawalagedara, V. Mithal, R. Oglesby, K. Salvi, P. K. Snyder, K. Steinhaeuser, D. Wang, and D. Wuebbles. Toward enhanced understanding and projections of climate extremes using physics-guided data mining techniques. *Nonlinear Processes in Geophysics*, 21(4):777–795, 2014.

[48] P. Gentine, M. Pritchard, S. Rasp, G. Reinaudi, and G. Yacalis. Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, 45(11):5742–5751, 2018.

[49] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[50] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 15379–15389. Curran Associates, Inc., 2019.

[51] Brian Groenke, Luke Madaus, and Claire Monteleoni. Climalign: Unsupervised statistical downscaling of climate variables via normalizing flows. In *Proceedings of the 10th International Conference on Climate Informatics*, CI2020, page 60–66, New York, NY, USA, 2020. Association for Computing Machinery.

[52] Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[53] Franz Hamilton, Alun L. Lloyd, and Kevin B. Flores. Hybrid modeling and prediction of dynamical systems. *PLOS Computational Biology*, 13(7):1–20, 07 2017.

[54] Katja Hansen, Grégoire Montavon, Franziska Biegler, Siamac Fazli, Matthias Rupp, Matthias Scheffler, O. Anatole von Lilienfeld, Alexandre Tkatchenko, and Klaus-Robert Müller. Assessment and validation of machine learning methods for predicting molecular atomization energies. *Journal of Chemical Theory and Computation*, 9(8):3404–3419, 2013.

[55] Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. Liquid time-constant networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):7657–7666, May 2021.

[56] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730): 1999–2049, 2020.

[57] IPCC. Global warming of 1.5c. an ipcc special report on the impacts of global warming of 1.5c above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty, 2018.

[58] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan Read, Jacob Zwart, Michael S Steinbach, and Vipin Kumar. Physics guided rnns for modeling dynamical systems: A case study in simulating lake temperature profiles. In *SIAM International Conference on Data Mining, SDM 2019*, SIAM International Conference on Data Mining, SDM 2019, pages 558–566. Society for Industrial and Applied Mathematics Publications, 2019.

[59] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan S. Read, Jacob A. Zwart, Michael Steinbach, and Vipin Kumar. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *ACM/IMS Trans. Data Sci.*, 2 (3), 2021.

[60] Peishi Jiang, Nis Meinert, Helga Jordão, Constantin Weisser, Simon Holgate, Alexander Lavin, Björn Lütjens, Dava Newman, Haruko Wainwright, Catherine Walker, and Patrick Barnard. Digital Twin Earth

– Coasts: Developing a fast and physics-informed surrogate model for coastal floods via neural operators. *2021 NeurIPS Workshop on Machine Learning for the Physical Sciences (ML4PS)*, 2021.

[61] Pengzhan Jin, Zhen Zhang, Aiqing Zhu, Yifa Tang, and George Em Karniadakis. Sympnets: Intrinsic structure-preserving symplectic networks for identifying hamiltonian systems. *Neural Networks*, 132, 12 2020.

[62] J. Nagoor Kani and Ahmed H. Elsheikh. DR-RNN: A deep residual recurrent neural network for model reduction. *CoRR*, abs/1709.00939, 2017.

[63] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440, June 2021.

[64] A. Karpatne, I. Ebert-Uphoff, S. Ravela, H. A. Babaie, and V. Kumar. Machine learning for the geosciences: Challenges and opportunities. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1544–1554, Aug 2019.

[65] Anuj Karpatne, Gowtham Atluri, James H. Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2318–2331, 2017.

[66] Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling. *arXiv e-prints*, page arXiv:1710.11431, October 2017.

[67] K. Kashinath, M. Mustafa, A. Albert, J-L. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, A. Manepalli, D. Chirila, R. Yu, R. Walters, B. White, H. Xiao, H. A. Tchelepi, P. Marcus, A. Anandkumar, P. Hassanzadeh, and null Prabhat. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194), 2021.

[68] Marat Khairoutdinov, David Randall, and Charlotte DeMott. Simulations of the atmospheric general circulation using a cloud-resolving model as a superparameterization of physical processes. *Journal of the Atmospheric Sciences*, 62(7 I):2136–2154, jul 2005.

[69] Yuehaw Khoo and Lexing Ying. Switchnet: A neural network model for forward and inverse scattering problems. *SIAM Journal on Scientific Computing*, 41 (5), 2019.

[70] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[71] Nikola B. Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Neural operator learning maps between function spaces. *CoRR*, abs/2108.08481, 2021.

[72] Rupa Kurinchi-Vendhan, Björn Lütjens, Ritwik Gupta, Lucien Werner, and Dava Newman. Wisosuper: Benchmarking super-resolution methods on wind and solar data. *Conference on Neural Information Processing Systems (NeurIPS) Workshop on Tackling Climate Change with Machine Learning (CCML)*, 2021.

[73] Corentin J. Lapeyre, Antony Misdariis, Nicolas Cazard, Denis Veynante, and Thierry Poinsot. Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *Combustion and Flame*, 203:255–264, 2019.

[74] Kookjin Lee and Kevin T. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020. ISSN 0021-9991.

[75] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. In *Advances in Neural Information Processing Systems*, volume 33, pages 6755–6766. Curran Associates, Inc., 2020.

[76] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *ICML*, 2021.

[77] Zongyi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Markov neural operators for learning chaotic systems. *CoRR*, abs/2106.06898, 2021.

[78] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[79] Burigede Liu, Nikola Kovachki, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, Andrew M. Stuart, and Kaushik Bhattacharya. A learning-based multiscale method and its application to inelastic impact problems. *Journal of the Mechanics and Physics of Solids*, 158:104668, 2022. ISSN 0022-5096. doi: https://doi.org/10.1016/j.jmps.2021.104668. URL https://www.sciencedirect.com/science/article/pii/S0022509621002982.

[80] Ziming Liu, Bohan Wang, Qi Meng, Wei Chen, Max Tegmark, and Tie-Yan Liu. Machine-learning nonconservative dynamics for new-physics detection. *Phys. Rev. E*, 104:055302, Nov 2021. doi: 10.1103/PhysRevE.104.055302. URL https://link.aps.org/doi/10.1103/PhysRevE.104.055302.

[81] Yun Long, Xueyuan She, and Saibal Mukhopadhyay. Hybridnet: Integrating model-based and data-driven learning to predict evolution of dynamical systems. In *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 551–560. PMLR, 29–31 Oct 2018.

[82] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. PDE-net: Learning PDEs from data. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3208–3216. PMLR, 10–15 Jul 2018.

[83] Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925, 2019.

[84] Edward Lorenz. Predictability - a problem partly solved. In Tim Palmer and Renate Hagedorn, editors, *Predictability of Weather and Climate*. Cambridge University Press, Cambridge, 2006.

[85] Edward N. Lorenz and Kerry A. Emanuel. Optimal sites for supplementary weather observations: Simulation with a small model. *Journal of the Atmospheric Sciences*, 55(3):399 – 414, 1998.

[86] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021.

[87] Björn Lütjens, Catherine H. Crawford, Mark Veillette, and Dava Newman. Pce-pinns: Physics-informed neural networks for uncertainty propagation in ocean modeling. *International Conference on Learning Representations (ICLR) Workshop on AI for Modeling Oceans and Climate Change*, May 2021.

[88] Björn Lütjens, Catherine H Crawford, Mark Veillette, and Dava Newman. Spectral PINNs: Fast uncertainty propagation with physics-informed neural networks. In *Advances in Neural Information Processing Systems (NeurIPS) Workshop on The Symbiosis of Deep Learning and Differential Equations (DLDE)*, 2021.

[89] Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[90] Marie C. McGraw and Elizabeth A. Barnes. Memory matters: A case for granger causality in climate variability studies. *Journal of Climate*, 31(8):3289 – 3300, 2018.

[91] Kendal McGuffie and Ann Henderson-Sellers. *A Climate Modeling Primer, Third Edition*. John Wiley and Sons, Ltd., Jan. 2005.

[92] Arvind Mohan, Don Daniel, Michael Chertkov, and Daniel Livescu. Compressed convolutional lstm: An efficient deep learning framework to model high fidelity 3d turbulence, 2019.

[93] Alberto C. Nogueira Jr., Felipe C. T. Carvalho, João Lucas S. Almeida, Andres Codas, Eloisa Bentivegna, and Campbell D. Watson. Reservoir computing in reduced order modeling for chaotic dynamical systems. In Heike Jagode, Hartwig Anzt, Hatem Ltaief, and Piotr Luszczek, editors, *High Performance Computing*, pages 56–72, Cham, 2021. Springer International Publishing.

[94] Paul A. O'Gorman and John G. Dwyer. Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. *Journal of Advances in Modeling Earth Systems*, 10(10):2548–2563, 2018.

[95] Peter J. Olver. *Symmetry Groups of Differential Equations*, pages 77–185. Springer New York, New York, NY, 1986.

[96] Sagaut P. *Large Eddy Simulation for Incompressible Flows: An Introduction*. Scientific Computation. Springer-Verlag Berlin Heidelberg, 3 edition, 2006.

[97] Tim Palmer, Bjorn Stevens, and Peter Bauer. We need an international center for climate modeling, 2019. URL https://blogs.scientificamerican.com/observations/we-need-an-international-center-for-climate-modeling/. last accessed 04/13/20.

[98] Ambrish Pandey, Janet D. Scheel, and Jörg Schumacher. Turbulent superstructures in rayleigh-bénard convection. *Nature Communications*, 9, 2018.

[99] Eric J. Parish and Karthik Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*, 305:758–774, 2016. ISSN 0021-9991.

[100] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Lett.*, 120, Jan 2018.

[101] Jaideep Pathak, Mustafa Mustafa, Karthik Kashinath, Emmanuel Motheau, Thorsten Kurth, and Marcus Day. Using machine learning to augment coarse-grid computational fluid dynamics simulations, 2020.

[102] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators. February 2022.

[103] Greg Pavliotis and Andrew Stuart. *Multiscale Methods Averaging and Homogenization*, volume 53 of *Texts in Applied Mathematics*. Springer-Verlag New York, 1 edition, 2008. ISBN 978-0-387-73829-1.

[104] Grace C. Y. Peng, Mark Alber, Adrian Buganza Tepole, William R. Cannon, Suvranu De, Savador Dura-Bernal, Krishna Garikipati, George Karniadakis, William W. Lytton, Paris Perdikaris, Linda Petzold, and Ellen Kuhl. Multiscale modeling meets machine learning: What can we learn? *Archives of Computational Methods in Engineering*, 28:1017–1037, 2021.

[105] Aviral Prakash, Kenneth E. Jansen, and John A. Evans. Invariant Data-Driven Subgrid Stress Modeling in the Strain-Rate Eigenframe for Large Eddy Simulation. *arXiv e-prints*, June 2021.

[106] Zhaozhi Qian, Krzysztof Kacprzyk, and Mihaela van der Schaar. D-CODE: Discovering closed-form ODEs from observed trajectories. In *International Conference on Learning Representations*, 2022.

[107] Alfio Quarteroni and Gianluigi Rozza. Reduced order methods for modeling and computational reduction, 2014.

[108] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, and Ali Ramadhan. Universal differential equations for scientific machine learning. *ArXiv*, abs/2001.04385, 2020.

[109] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686 – 707, 2019.

[110] S. Rasp. Coupled online learning as a way to tackle instabilities and biases in neural network parameterizations: general algorithms and lorenz 96 case study (v1.0). *Geoscientific Model Development*, 13 (5):2185–2196, 2020.

[111] Stephan Rasp. Lorenz 96 is too easy! machine learning research needs a more realistic toy model, July 2019. URL https://raspstephan.github.io/blog/lorenz-96-is-too-easy/. last accessed May 2022.

[112] Stephan Rasp, Michael S. Pritchard, and Pierre Gentine. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39):9684–9689, 2018.

[113] Stephan Rasp, Peter D. Dueben, Sebastian Scher, Jonathan A. Weyn, Soukayna Mouatadid, and Nils Thuerey. Weatherbench: A benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11), 2020.

[114] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 566:195 – 204, 2019.

[115] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1530–1538. JMLR.org, 2015.

[116] Juliette N. Rooney-Varga, Florian Kapmeier, John D. Sterman, Andrew P. Jones, Michele Putko, and Kenneth Rath. The climate action simulation. *Simulation & Gaming*, 51(2):114–140, 2020. URL https://en-roads.climateinteractive.org/.

[117] Justin Sirignano, Jonathan F. MacArt, and Jonathan B. Freund. Dpm: A deep learning pde augmentation method with application to large-eddy simulation. *Journal of Computational Physics*, 423, 2020.

[118] Ralph C. Smith. Uncertainty quantification: Theory, implementation, and applications. In *Computational science and engineering*, page 382. SIAM, 2013.

[119] SnagglebitInkArt. Sloth nesting dolls - hand painted modern russian matryoshka doll set, 2022. URL https://www.etsy.com/ie/listing/690540535/sloth-nesting-dolls-hand-painted-modern. last accessed 01/22.

[120] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

[121] Kim Stachenfeld, Drummond Buschman Fielding, Dmitrii Kochkov, Miles Cranmer, Tobias Pfaff, Jonathan Godwin, Can Cui, Shirley Ho, Peter Battaglia, and Alvaro Sanchez-Gonzalez. Learned simulators for turbulence. In *International Conference on Learning Representations*, 2022.

[122] Karen Stengel, Andrew Glaws, Dylan Hettinger, and Ryan N. King. Adversarial super-resolution of climatological wind and solar data. *Proceedings of the National Academy of Sciences*, 117(29):16805–16815, 2020.

[123] Adam Subel, Ashesh Chattopadhyay, Yifei Guan, and Pedram Hassanzadeh. Data-driven subgrid-scale modeling of forced burgers turbulence using deep learning with generalization to higher reynolds numbers via transfer learning. *Physics of Fluids*, 33(3): 031702, 2021.

[124] Nathaniel Thomas, Tess E. Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *CoRR*, abs/1802.08219, 2018.

[125] Tobias Thornes, Peter Duben, and Tim Palmer. On the use of scale-dependent precision in earth system modelling. *Quarterly Journal of the Royal Meteorological Society*, 143(703):897–908, 2017.

[126] Benjamin A. Toms, Elizabeth A. Barnes, and Imme Ebert-Uphoff. Physically interpretable neural networks for the geosciences: Applications to earth system variability. *Journal of Advances in Modeling Earth Systems*, 12(9), 2020.

[127] Kiwon Um, Robert Brand, Yun Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-Loop: Learning from Differentiable Physics to Interact with Iterative PDE-Solvers. *Advances in Neural Information Processing Systems*, 2020.

[128] Thomas Vandal, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R. Ganguly. Deepsd: Generating high resolution climate change projections through single image super-resolution. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1663–1672, New York, NY, USA, 2017. Association for Computing Machinery.

[129] Peidong Wang, Janni Yuval, and Paul A. O'Gorman. Non-local parameterization of atmospheric subgrid processes with neural networks, 2022.

[130] Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science Advances*, 7(40), 2021.

[131] Mark J. Webb, Adrian P. Lock, Christopher S. Bretherton, Sandrine Bony, Jason N.S. Cole, Abderrahmane Idelkadi, Sarah M. Kang, Tsuyoshi Koshiro, Hideaki Kawai, Tomoo Ogura, Romain Roehrig, Yechul Shin, Thorsten Mauritsen, Steven C. Sherwood, Jessica Vial, Masahiro Watanabe, Matthew D. Woelfle, and Ming Zhao. The impact of parametrized convection on cloud feedback. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 373(2054), nov 2015.

[132] Daniel S. Wilks. Effects of stochastic parametrizations in the lorenz '96 system. *Quarterly Journal of the Royal Meteorological Society*, 131(606):389–407, 2005.

[133] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Comput. Surv.*, jan 2022.

[134] Schiesser William. *The Numerical Method of Lines: Integration of Partial Differential Equations*. Elsevier, 1991.

[135] Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25: 1432–1467, 2015.

[136] Jin-Long Wu, Heng Xiao, and Eric Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Phys. Rev. Fluids*, 3:074602, Jul 2018.

[137] Jin-Long Wu, Karthik Kashinath, Adrian Albert, Dragos Chirila, Prabhat, and Heng Xiao. Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. *Journal of Computational Physics*, 406, 2020.

[138] You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. Tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Trans. Graph.*, 37(4), jul 2018.

[139] Alireza Yazdani, Lu Lu, Maziar Raissi, and George Em Karniadakis. Systems biology informed deep learning for inferring parameters and hidden dynamics. *PLOS Computational Biology*, 16(11):1–19, 11 2020.

[140] Yuan Yin, Vincent Le Guen, Jérémie Dona, Emmanuel de Bézenac, Ibrahim Ayed, Nicolas Thome, and Patrick Gallinari. Augmenting physical models with deep networks for complex dynamics forecasting. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124012, dec 2021. doi: 10.1088/1742-5468/ac3ae5. URL https://doi.org/10.1088/1742-5468/ac3ae5.

[141] Janni Yuval, Paul A. O'Gorman, and Chris N. Hill. Use of neural networks for stable, accurate and physically consistent parameterization of subgrid atmospheric processes with good performance at reduced precision. *Geophysical Research Letter*, 48: e2020GL091363, 2021.

[142] Yang Zeng, Jin-Long Wu, and Heng Xiao. Enforcing imprecise constraints on generative adversarial networks for emulating physical systems. *Communications in Computational Physics*, 30(3):635–665, 2021.

[143] Liang Zhang, Gang Wang, and Georgios B. Giannakis. Real-time power system state estimation and forecasting via deep unrolled neural networks. *IEEE Transactions on Signal Processing*, 67(15): 4069–4077, 2019.

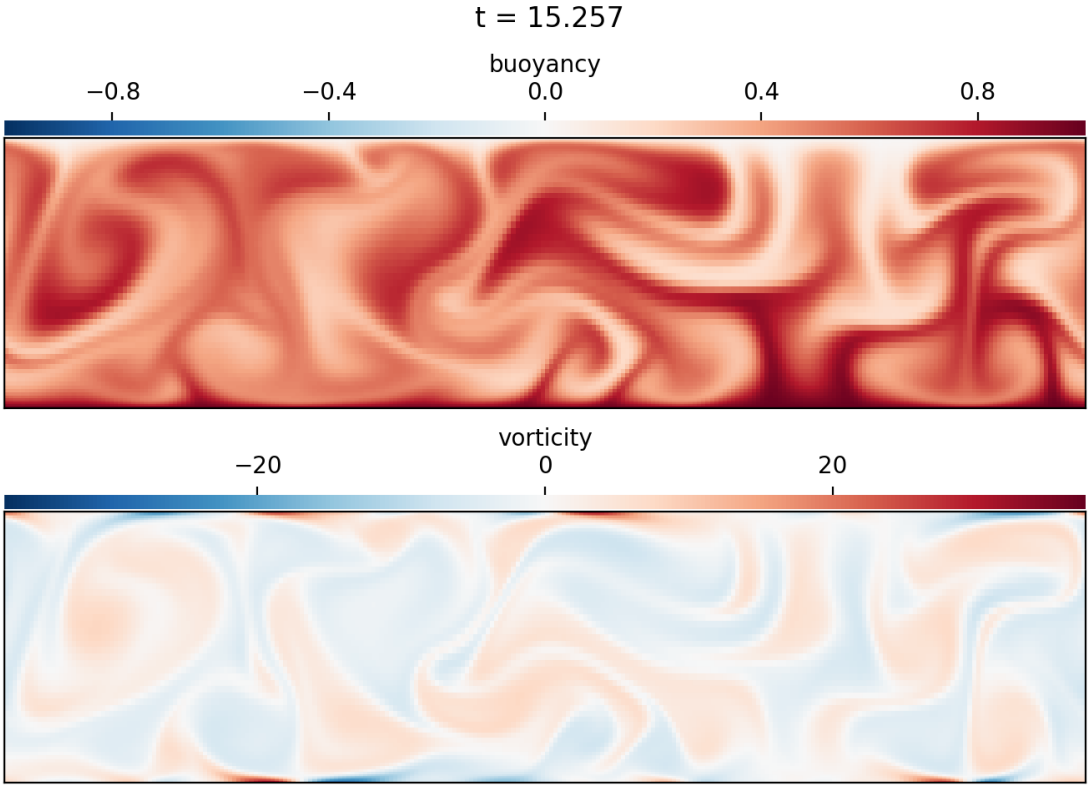
[144] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.*, 120, Apr 2018.

[145] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

A. Appendix

A.1. Rayleigh-Bénard Convection

We plan to extend the multiscale neural operator to higher-dimensional systems; starting with the Rayleigh-Bénard Convectione equations, as displayed in Figure 6.



(a) Ground-truth

Figure 6: **We are planning to extend MNO to Rayleigh-Bénard Convection.** We depicted a sample plot for ground-truth training data of the 2D RBC.

A.1.1. Details and Interpretation

Rayleigh-Bénard Convection (RBC) is one of the simplest turbulent, chaotic, convection-dominated flows. The equation finds applications in fluid dynamics, atmospheric dynamics, radiation, phase changes, magnetic fields, and more [98].

So far, we have generated a ground-truth dataset that we implemented with the 2D turbulent Rayleigh-Bénard Convenction equations with Dedalus spectral solver [23] similar to [98]:

$$\begin{aligned} \frac{\delta u}{\delta t} + u \cdot \nabla u &= \sqrt{\frac{\mathrm{Pr}}{\mathrm{Ra}}} \nabla^2 u - \nabla p + b \\ \frac{\delta T}{\delta t} + u \cdot \nabla T &= \frac{1}{\sqrt{RaPr}} \nabla^2 T \\ \nabla \cdot u &= 0 \end{aligned} \tag{10}$$

with temperature/buoyancy, T, Rayleigh number, $\mathrm{Ra} = g\alpha\Delta T H^3/(\nu\kappa)$, Prandtl number, $\mathrm{Pr} = \nu/\kappa$, thermal expansion coefficient, α, kinematic viscosity, ν, thermal diffusivity, $\kappa = \frac{1}{\sqrt{\mathrm{RaPr}}}$, acceleration due to gravity, g, temperature difference, ΔT, unit vector, e, pressure, p, Nusselt number, $\mathrm{Nu} = \sqrt{\frac{\mathrm{Pr}}{\mathrm{Ra}}}$, Reynolds number, $Re = \sqrt{\langle \nabla^2 u \rangle_{V,t} \frac{\mathrm{Ra}}{\mathrm{Pr}}}$, and full volume-time

average, $\langle\cdot\rangle_{V,t}$, cell length, L_x. The equations have been non-dimensionalized with the free-fall velocity, $U_f = \sqrt{g\alpha\Delta H}$, and cell height, H. In the horizontal direction, x, we use periodic boundary conditions and in the vertical direction, z, we use no-slip boundary conditions for the velocity, $u(z=0) = u(z=L_z) = 0$, and fixed temperatures, $T(z=0) = L_z,\ T(z = L_z) = 0$. The inital conditions are sampled randomly, $b(z, t=0) = L_z + z + z(L_z - z)\omega$, with $\omega \sim \mathcal{N}(0, 1 \times 10^{-3}$.

We chose: Ra $= 2 \times 10^6$, Pr $= 1$, $L_x = 4$, $H = 1$.

A.2. Fourier Neural Operator

Our neural operator for learning subgrid parametrizations is based on Fourier neural operators [76]. Intuitively, the neural operator learns a parameter-to-solution mapping by learning a global convolution kernel. In detail, it learns the operator to transforms the current large-scale state, $\underline{X}(x_{0:K}, t) \in \mathbb{R}^{K\times d_X}$ to the subgrid parametrization, $\underline{\hat{f}}_x(x_{0:K}, t) := \underline{X}_{0:K} \in \mathbb{R}^{K\times d_X}$ with number of grid points, K, and input dimensionality, d_X, according to the following equations:

$$\begin{aligned} \underline{v}_0 &= \underline{X}_{0:K} P^T + 1^{K\times 1} b_P \\ \underline{v}_{i+1} &= \sigma\left(\underline{v}_i W^T + \int_{D_x} \kappa_\phi(x, x') v_i(x') dx'\right) \\ &\approx \sigma\left(\underline{v}_i W^T + 1^{n_v\times 1} b_W + \mathcal{F}^{-1}(R_\phi \cdot \mathcal{F}\underline{v}_i)\right) \\ \hat{f}_{x,0:K} &= \underline{v}_{n_d} Q^T + 1^{K\times 1} b_Q \end{aligned} \tag{11}$$

First, MNO lifts the input via a linear transform with matrix, $P \in \mathbb{R}^{n_v\times d_X}$, bias, $b_P \in \mathbb{R}^{1\times n_v}$, vector of ones, $1^{K\times 1}$, and number of channels, n_v. The linear transform is local in space, i.e., the same transform is applied to each grid point.

Second, multiple nonlinear "Fourier layers" are applied to the encoded/lifted state. The encoded/lifted state's, $\underline{v}_i \in \mathbb{R}^{K\times n_v}$, spatial dimension is transformed into the Fourier domain via a fast Fourier transform. We implement the FFT as a multiplication with the pre-built forward and inverse Type-I DST matrix, $\mathcal{F} \in \mathbb{C}^{k_{\max}\times K}$ and $\mathcal{F}^{-1} \in \mathbb{C}^{K\times k_{\max}}$, respectively, returning the vector, $\mathcal{F}\underline{v}_i \in \mathbb{C}^{k_{\max}\times n_v}$.

The dynamics are learned via convoluting the encoded state with a weight matrix. In Fourier space, convolution is a multiplication, hence each frequency is multiplied with a complex weight matrix across the channels, such that $R \in \mathbb{C}^{k_{\max}\times n_v\times n_v}$. In parallel to the convolution with R, the encoded state is multiplied with the linear transform, $W \in \mathbb{R}^{n_v\times n_v}$, and bias, $b_W \in \mathbb{R}^{1\times n_v}$. From a representation learning-perspective, the Fourier decomposition as a fast and interpretable feature extraction method that extracts smooth, periodic, and global features. The linear transform can be interpreted as residual term concisely capturing nonlinear residuals.

So far, we have only applied linear transformations. To introduce nonlinearities, we apply a nonlinear activation function, σ, at the end of each Fourier layer. While the non-smoothness of the activation function ReLu, $\sigma(z) = \max(0, z)$, could introduce unwanted discontinuities in the solution, we choose it resulted in more accurate models than smoother activation functions such as tanh or sigmoid.

Finally, the transformed state, v_{n_d}, is projected back onto solution space via another linear transform, $Q \in \mathbb{R}^{d_X\times n_v}$, and bias, b_Q.

The values of all trainable parameters, P, R, W, Q, b_*, are found by using a nonlinear optimization algorithm, such as stochastic gradient descent or, here, Adam [70]. We have used MSE between the predicted, $\hat{f}_x$, and ground-truth, f_x, subgrid parametrizations as loss. The neural operator is implemented in pytorch, but does not require an autodifferentiable PDE solver to generate training data. During implementation, we used the DFT which assumes a uniformly spaced grids, but can be exchanged with non-uniform DFTs (NUDFT) to transform non-uniform grids [42].

A.3. Multiscale Lorenz96

A.3.1. Details and Interpretation

The equation contains K variables, $X_k \in \mathbb{R}$, and JK small-scale variables, $Y_{j,k} \in \mathbb{R}$ that represent large-scale or small-scale atmospheric dynamics such as the movement of storms or formation of clouds, respectively. At every time-step each large-scale variable, X_k, influences and is influenced by J small-scale variables, $Y_{0:J,k}$. The coupling could be interpreted

as X_k causing static instability and $Y_{j,k}$ causing drag from turbulence or latent heat fluxes from cloud formation. The indices k, j are both interpreted as latitude, while $k \in \{0, ..., K-1\}$ indexes boxes of latitude and $j \in \{0, ..., J-1\}$ indexes elements inside the box. Illustrated on a 1D Earth with a circumference of $360°$ that is discretized with $K = 36, J = 10$, one a spatial step in k, j would equal $10°, 1°$, respectively [84]; we choose $K = J = 4$. A time step with $\Delta t = 0.005$ would equal 36 minutes [84].

We choose a large forcing, $F > 10$, for which the equation becomes chaotic. The last terms in each equation capture the interaction between small- and large-scale, $f_{x,k} = -\frac{hc}{b}\sum_{j=0}^{J} Y_{j,k}(X_k), f_y$. The scale interaction is defined by the parameters where $h = 0.5$ is the coupling strength between spatial scales (with no coupling if h would be zero), $b = 10$ is the relative magnitude, and $c = 8$ the evolution speed of $X - Y$. The linear, $-X_k$, and quadratic terms, X_*^2, model dissipative and advective (e.g., moving) dynamics, respectively.

The equation assumes perfect "scale separation" which means that small-scale variables of different grid boxes, k, are independent of each other at a given timestep, $Y_{j_1,k_2}(t) \perp Y_{j_2,k_1}(t)\ \forall t, j_1, j_2, k_1 \neq k_2$. The separation of small- and large-scale variables can be along the same or different domain and the discretized variables would then be $y \in [0, \Delta x]$ or $y \in [y_0, y_{\text{end}}]$, respectively. The equation wraps around the full large- or small-scale domain by using periodic boundaries, $X_{-k}{:=}X_{K-k}$, $X_{K+k}{:=}X_k$, $Y_{-j,k}{:=}Y_{J-j,k}$, $Y_{J+j,k}{:=}Y_{j,k}$. Note that having periodic boundary conditions in the small-scale domanin allows for superparametrization, i.e., independent simulation of the small-scale dynamics [27] and differs from the three-tier Lorenz96 where variables at the borders of the small-scale domain depend on small-scale variables of the neighbouring k [125].

A.3.2. Simulation

The initial conditions are sampled uniformly from a set of integers, $X(t_0) \sim U(-5, -4, ..., 5, 6)$, as a mean-zero unit-variance Gaussian $Y(t_0) \sim \mathcal{N}(0, 1)$, and lower scale Gaussian $Z(t_0) \sim 0.05\mathcal{N}(0, 1)$. The train and test set contains 4k and 1k samples, respectively. Each sample is $T = 1$ model time unit (MTU) or 200 (=$T/\Delta t$) time-steps long, which corresponds to 5 Earth days (= $T/\Delta t * 36$min with $\Delta t = 0.005$) [84]. Hence, our results test the generalization towards different initial conditions, but not robustness to extrapolation or different choices of parameters, c, b, h, F. The sampling starts after $T = 10$. warmup time. The dataset uses double precision.

We solve the equation by fourth order Runge-Kutta in time with step size $\Delta t = 0.005$, similar to [85]. For a PDE that is discretized with fixed time step, Δt, the ground-truth train and test data, $h_{x,0:K}(t)$, is constructed by integrating the coupled large- and small-scale dynamics.

Note, that the neural operator only takes in the current state of the large-scale dynamics. Hence, , i.e., it uses the full large-scale spatial domain as input, which exploits spatial correlations and learns parametrizations that are independent of the large-scale spatial discretization.

Our method can be queried for infinite time-steps into the future as it does not use time as input.

We are incorporating the prior knowledge from physics by calculating the large-scale dynamics, $dX_{LS,0:K}$. Note that the small-scale physics do not need to be known. Hence, MNO could be applied to any fixed time-step dataset for which an approximate model is known.

A.4. Appendix to Illustration of MNO via multiscale Lorenz96

The other large-scale (LS) and fine-scale (FS) terms are

$$
\begin{aligned}
&\text{filtered FS dynamics, } \overline{\mathcal{N}(u)}(x) = \begin{cases} \frac{\delta X_k}{\delta t} & \text{if } x = k(J+1)\ \forall k \in \{0, \dots, K\} \\ 0 & \text{otherwise} \end{cases} \\
&\text{LS dynamics, } \mathcal{N}(\bar{u})(x) = \begin{cases} \frac{\delta \bar{X}_k}{\delta t} & \text{if } x = k(J+1)\ \forall k \in \{0, \dots, K\} \\ 0 & \text{otherwise} \end{cases} \\
&\text{with abbreviation, } \frac{\delta \bar{X}_k}{\delta t} := X_{k-1}(X_{k+1} - X_{k-2}) - X_k + F \\
&\text{LS state, } \bar{u}(x) = \mathcal{G} * u(x) = [X_0, 0, ..., 0, X_1, 0, ..., X_K]
\end{aligned}
\tag{12}
$$

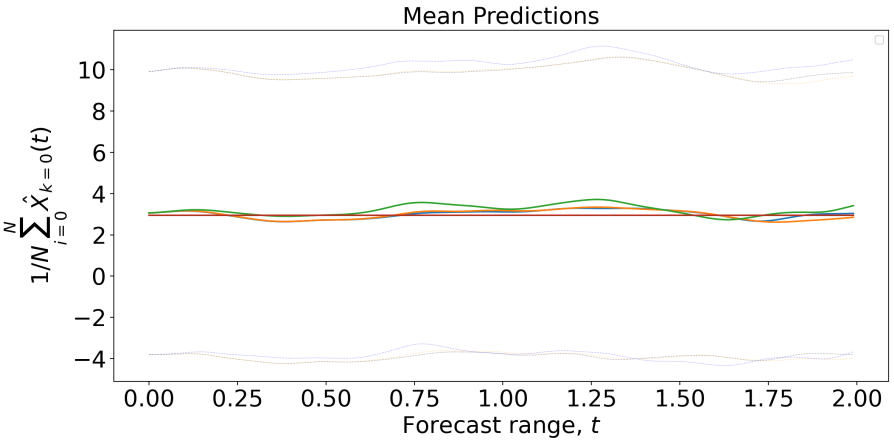


Figure 7: **Mean accuracy.** MNO (orange) most accuracy forecasts the mean (solid) and standard deviation (dotted) of the ground-truth DNS (blue) in comparison to ML-based parametrizations (green) and climatology (red).

.

A.5. Appendix to Results

A.5.1. Accuracy

Figure 7 shows that the predicted mean and standard deviation of MNO (orange) closely follows the ground-truth (blue). The ML-based parametrization (green) follows the ground-truth only for a few time steps (until $\sim t = 0.125$). The climatology (red) depicts the average prediction in the training dataset.

A.5.2. Model configuration

Multiscale Lorenz96: MNO As hyperparameters we chose the number of channels, $n_v = 64$, number of retained modes, $k_{\max} = 3$, number of Fourier layers, $n_d = 3$, and no batch norm layer. The time-series modeling task uses a history of only one time step to learn chaotic dynamics [77]. We are using ADAM optimizer with learning rate, $\lambda = 0.001$, step size, 20, number of epochs, $n_e = 2$, and an exponential learning rate scheduler with gamma, $\gamma = 0.9$ [70]. Training took $1:50$min on a single core Intel i7-7500U CPU@2.70GHz.

Multiscale Lorenz96: ML-based parametrization The ML-based parametrizations uses a ResNet with $n_{\text{layers}} = 2$ residual layers that contain a fully connected network with $n_{\text{units}} = 32$ units. The model is optimized with Adam [70] with learning rate 0.01, $\beta = (0.9, 0.999)$, $\epsilon = 1 * 10^{-8}$, trained for $20 n_{\text{epochs}} = 20$.

Multiscale Lorenz96: Traditional parametrization The traditional parametrization uses least-squares to find the best linear fit. The weight matrix is computed with $A = (X^T X)^{-1} X^T Y$, where X and Y are the concatenation of input large-scale features and target parametrizations, respectively. Inference is conducted with $\hat{y} = Ax$.

A.6. Neural networks vs. neural operators

Most work in physics-informed machine learning relies on fully-connected neural networks (FCNNs) or convolutional neural networks [63]. FCNNs however are mappings between finite-dimensional spaces and learn mappings for single equation instances rather than learning the PDE solver. In our case FCNNs only learn mappings on fixed spatial grids. We leverage the recently formulated neural operators to extend the formulation to arbitrary grids. The key distinction is that the FCNN learns a parameter-dependent set of weights, Φ_{a_y}, that has to be retrained for every new parameter setting. The neural operator is a learned function mapping with parameter-independent weights, Θ, that takes parameter settings as input and returns a function over the spatial domain, $G_\Theta(a_y)$. In comparison, the forcing term is approximated by an FCNN as

$\hat{f}_{x,\Phi}(x_k; a_y) = g_{\Phi_{a_y}}(x_k)$ and by a neural operator as $\hat{f}_{x,\Theta}(x_k; a_y) = G_\Theta(a_y)(x_k)$. The mappings are given by:

$$\begin{aligned} &\text{FCNN: } g_{\Phi_{a_y}} : \ D_x \to \mathbb{R}^{d_X}, \\ &\text{NO: } G_\Theta : \ H_{a_y}(D_x; \mathbb{R}^{d_{a_y}}) \to H_X(D_x; \mathbb{R}^{d_X}). \end{aligned} \tag{13}$$

H_{a_y} is a function space (Banach) of PDE parameter functions, a_y, that map the spatial domain, D_y, onto d_{a_y} dimensional parameters, such as ICs, BCs, parameters, or forcing terms. H_X is the function space of residuals that map the spatial domain, D_x, onto the space of d_X-dimensional residuals, $\mathbb{R}^{d_X}$.