PHILOSOPHICAL TRANSACTIONS A

royalsocietypublishing.org/journal/rsta

Review

Cite this article: Kashinath K *et al*. 2021 Physics-informed machine learning: case studies for weather and climate modelling. *Phil. Trans. R. Soc. A* **379**: 20200093. https://doi.org/10.1098/rsta.2020.0093

Accepted: 24 November 2020

One contribution of 13 to a theme issue 'Machine learning for weather and climate modelling'.

Subject Areas:
artificial intelligence, computational physics, atmospheric science, fluid mechanics

Keywords:
neural networks, physical constraints, turbulent flows, physics-informed machine learning, weather and climate modeling

Author for correspondence:
Karthik Kashinath
e-mail: kkashinath@lbl.gov

Physics-informed machine learning: case studies for weather and climate modelling

K. Kashinath[1], M. Mustafa[1], A. Albert[1,2], J-L. Wu[1,3], C. Jiang[1,4], S. Esmaeilzadeh[5], K. Azizzadenesheli[6], R. Wang[1,7], A. Chattopadhyay[1,8], A. Singh[1,2], A. Manepalli[1,2], D. Chirila[9], R. Yu[7], R. Walters[10], B. White[2], H. Xiao[11], H. A. Tchelepi[5], P. Marcus[4], A. Anandkumar[3,12], P. Hassanzadeh[8] and Prabhat[1]

[1]NERSC - Lawrence Berkeley National Lab, Berkeley, CA, USA
[2]Terrafuse Inc., Berkeley, CA, USA
[3]Caltech, Pasadena, CA, USA
[4]University of California, Berkeley, CA, USA
[5]Stanford University, Stanford, CA, USA
[6]Purdue University, West Lafayette, IN, USA
[7]UC San Diego, La Jolla, CA, USA
[8]Rice University, Houston, TX, USA
[9]Alfred Wegener Institute, Bremerhaven, Germany
[10]Northeastern University, Boston, MA, USA
[11]Virginia Tech, Blacksburg, VA, USA
[12]NVIDIA, Santa Clara, California, USA

KK, 0000-0002-9311-5215; BW, 0000-0002-3739-9604

Machine learning (ML) provides novel and powerful ways of accurately and efficiently recognizing complex patterns, emulating nonlinear dynamics, and predicting the spatio-temporal evolution of weather and climate processes. Off-the-shelf ML models, however, do not necessarily obey the fundamental governing laws of physical systems, nor do they generalize well to scenarios on which they have not been trained. We survey systematic approaches to incorporating physics and domain knowledge into ML models and distill these approaches into broad categories. Through 10 case studies, we show how these approaches have been used successfully for

THE ROYAL SOCIETY PUBLISHING

emulating, downscaling, and forecasting weather and climate processes. The accomplishments of these studies include greater physical consistency, reduced training time, improved data efficiency, and better generalization. Finally, we synthesize the lessons learned and identify scientific, diagnostic, computational, and resource challenges for developing truly robust and reliable physics-informed ML models for weather and climate processes.

This article is part of the theme issue 'Machine learning for weather and climate modelling'.

1. Introduction

Machine learning (ML) and deep learning (DL) are making significant inroads into the sciences as they provide powerful methods for analysing complex data, extracting nonlinear relationships within massive datasets, and building predictive models. They are even enabling novel scientific discoveries that were nearly impossible with traditional statistical methods [1,2]. In a nutshell, three key forces have contributed to the unprecedented success of ML and DL [3]: (i) access to vast quantities of data; (ii) advances in computational algorithms; and (iii) an exponential increase in computational horsepower in accordance with Moore's Law [4,5]. Process-based numerical simulations, including those for weather and climate modelling applications, are compute- and resource-intensive, requiring extensive customized engineering for encoding governing equations and other domain knowledge [6]. The key differentiator is that ML and DL models can learn complex tasks from vast quantities of data and be significantly more computationally efficient, sometimes up to billions of times faster [7]. These characteristics of ML and DL enable breakthroughs across many scientific applications [8,9]. Henceforth, for simplicity, we use ML to refer to ML *and* DL, which is a subset of ML.

While ML has many promising strengths and advantages over physical modelling and traditional statistical approaches, there are also several challenges in making it trustworthy and robust for a wide range of scientific applications so that it can be reliably adopted for decision- and policy-making. One of the foremost challenges of ML is that it does not always obey the underlying physical principles of the systems it is applied to [6]. While ML models are capable of learning the underlying relationships that exist in the data, they do not consistently respect those principles in their predictions, especially when used in situations that they are not trained on, i.e. they do not generalize well to new scenarios. Additionally, ML requires copious high-quality data to train models with larger capacity that generalize better [10].

To address these pressing challenges, researchers have attempted to develop novel and effective strategies to incorporate domain knowledge and physical principles into ML models. This has resulted in the emergence of the field of physics-informed machine learning (PIML), also referred to as knowledge-guided machine learning (KGML) [11,12]. Karpatne *et al.* [11] and Willard *et al.* [12] broadly survey research work in this field that applies across the sciences. By contrast, this article focuses on work that pertains to weather and climate modelling.

2. Physics-informed machine learning: objectives, approaches, applications

(a) Objectives of physics-informed machine learning

By incorporating physical principles, governing laws and domain knowledge into ML models, the rapidly growing field of PIML seeks to:

- Build physically consistent and scientifically sound predictive models.
- Increase data efficiency, i.e. train models with fewer data points.
- Accelerate the training process, i.e. help models converge faster to optimal solutions.
- Improve the generalizability of models to make reliable predictions for unseen scenarios, including applicability to non-stationary systems, e.g. a changing climate.
- Enhance transparency and interpretability to make models more trustworthy.

(b) Ten key approaches to incorporate physics into ML

Researchers in weather and climate science have used many ways to incorporate physics and domain knowledge into ML models. Some of their approaches draw on ideas from the applied mathematics, dynamical systems, and fluid dynamics communities. Ways to incorporate physics and domain knowledge include direct approaches such as enforcing conservation laws and indirect approaches such as using domain expertise to design models that are better suited for the physical process being modelled. Most methods to incorporate physics and domain knowledge can be broadly categorized into the 10 approaches described below, listed in approximate descending order of pervasiveness. Incorporating one or more of these elements into models strengthens their ability to achieve the PIML objectives listed in §2a. Interpretability and uncertainty quantification (UQ) are listed last as they are not ways to incorporate physics but apply to all PIML models.

(i) Custom-designed loss functions/physics-based regularizations

Custom loss functions, also referred to as physics-based regularizations, help prevent overfitting and solve ill-posed problems. During training, ML models use a loss function in their optimization process typically defined simply as a mean-square error (MSE) or a root-mean-square error (RMSE) loss between the ground truth and predictions. One of the simplest and most widely used ways to incorporate physics is via regularizations, where the loss function is augmented with additional terms that are physics-based. The relative weights of the physics-based losses are adjustable hyper-parameters. This approach is sometimes referred to as imposing 'soft' constraints, which will be contrasted with imposing 'hard' constraints in §2bii.

Using customized loss functions in PIML models has been employed by Karpatne *et al.* [13] for modelling lake temperatures; Beucler *et al.* [14] to penalize the violation of conservation laws; Raissi *et al.* [15] in developing physics-informed neural networks (NN) to solve nonlinear partial differential equations (PDE) in fluid dynamics, quantum mechanics, reaction–diffusion systems, and nonlinear wave dynamics; and Zhu *et al.* [16] for surrogate modelling of transient PDEs in turbulent flows. This approach is more effective than unconstrained ML models [17,18]. However, the imposed 'soft' constraints are not required to be strictly satisfied and their relative importance to the standard MSE loss is tunable, so there are no generalizability guarantees.

(ii) Custom-designed neural network architectures to enforce physical constraints

Custom-designed NN architectures are a powerful approach to incorporating physics because constraints can be strictly enforced, including in new scenarios [19]. The modularity of NNs offers opportunities for the design of novel neurons, layers, or blocks that encode or enforce specific physical properties.

Beucler *et al.* [14] designed conservation layers to strictly enforce conservation laws in their NN emulator of atmospheric convection. Mohan *et al.* [20] guaranteed continuity (mass conservation) from NNs for coarse-graining of three-dimensional turbulence by encoding the curl operator. Jiang *et al.* [21] designed a PDE layer to strictly enforce PDE constraints for super-resolution of turbulence. Daw *et al.* [22] modified the long short-term memory (LSTM) model architecture to introduce an intermediate variable to strictly preserve monotonicity in a NN model of lake temperatures. A key advantage of physics-constrained NN architectures is that they can be used to impose 'hard' constraints that are guaranteed to be satisfied, when compared with the 'soft' constraints described in §2bi, and hence are more generalizable.

(iii) Symmetries, invariances and equivariances

Embedding symmetries, invariances, and equivariances in ML models are powerful ways to encode physical properties. They also achieve substantially simpler models with reduced the data requirements and higher prediction accuracy [23–25]. Symmetries, invariances, and equivariances represent geometric properties of spatio-temporal dynamics and physical systems. Noether's

theorems establish a correspondence between conserved quantities of PDEs and groups of symmetries; for example, rotational symmetry corresponds to the conservation of angular momentum.

Thomas *et al.* [26] embedded equivariances via tensor field networks to improve robustness and generalization of NNs in applications in physics and chemistry. Ling *et al.* [27] used an invariant tensor basis to embed Galilean invariance into a NN to learn a model for the Reynolds stress anisotropy tensor in turbulent flows and showed improved prediction accuracy compared to a generic NN that did not embed this invariance property. Cohen *et al.* [28] developed spherical convolutional NNs (CNN) using rotation equivariance to operate on data from spherical domains, such as global weather and climate data, and demonstrated computational efficiency, numerical accuracy, and effectiveness. Jiang *et al.* [29] extended this to unstructured grids using parametric differential operators to develop efficient and compact networks with high accuracy on climate pattern segmentation. Wang *et al.* [30] incorporated symmetries of translation, rotation, uniform motion, and scaling, and showed improved predictions of the time evolution of oceanic flows. Chattopadhyay *et al.* [31] used an equivariant spatial transformer network to predict geophysical turbulence and showed that the equivariance preserving property improves prediction accuracy. Beucler *et al.* [32] used non-dimensionalization and scaling relationships leveraging the Clausius–Clapeyron equation to improve the generalizability of models for a data-driven convective parameterization. In effect, non-dimensionalization results in variables that have similar distributions in both climates, i.e. they are climate-invariant, thus changing a hard extrapolation problem to an interpolation problem due to the exponential dependence of humidity on temperature.

(iv) Stochasticity

Stochastic methods are essential for accurately representing the inherently chaotic and turbulent nature of weather and climate systems, and the uncertainties in subgrid-scale processes, but are often absent from weather and climate models. Even if resolved-scale initial conditions were known perfectly, stochasticity is needed as it is impossible to represent subgrid-scale processes as a function of resolved-scale variables without error. Although the PDEs that describe the physical climate system are deterministic, there are important reasons why the computational representations of these equations should be stochastic: such representations better respect the scaling symmetries of the underlying PDEs, improve forecasting skill, and reduce systematic model errors [33]. Representing subgrid-scale processes with an ensemble of predictions instead of a single prediction is more accurate when initial conditions are uncertain, because errors in the initial condition exceed errors in the model that generates the ensembles. A natural way to model distributions and incorporate stochasticity and uncertainty in ML is through probabilistic models such as Bayesian NNs and generative models [34].

Krasnopolsky *et al.* [35] use an ensemble of NNs to learn stochastic convection parameterizations for climate and numerical weather prediction models from data simulated by a cloud resolving model. Recent work by Gagne *et al.* [36] develops a stochastic parameterization using generative adversarial networks (GAN), a class of deep generative models. Groenke *et al.* [37] develop a novel unsupervised statistical downscaling model using normalizing flows, a type of generative model that allows for both conditional and unconditional sampling from the joint distribution over high- and low-resolution spatial fields.

(v) Stability

Stable models are essential in at least two PIML applications in weather and climate modelling: (i) if instabilities tend to develop and grow over time after embedding a PIML emulator in a weather or climate model; and (ii) when the PIML model is used for forecasting and has long-term drift that can lead to unphysical behaviour. Stable PIML models can be achieved by careful design of the ML model and/or by careful consideration of the physics in choosing appropriate inputs and

outputs of the model. Furthermore, using tools from stability theory and spectral theory can shed light on the causes and modes of instability.

Researchers have devoted tremendous effort to developing stable physics-based numerical models that simulate dynamical systems [38]. Designing stable PIML models is in its infancy [39], though there are several relevant examples. Erichson *et al.* [40] encode Lyapunov stability into an autoencoder model for predicting fluid flow and sea surface temperature. They show improved generalizability and reduced prediction uncertainty for NNs that preserve Lyapunov stability. Lusch *et al.* [41] used DL to discover physically interpretable universal linear embeddings of nonlinear dynamics from data, a powerful approach that was extended by Mamakoukas *et al.* [42] for imposing stability constraints on the data-driven model to improve the prediction of nonlinear systems over long horizons.

In climate modelling, Brenowitz *et al.* [43] used stability theory to identify the causes and conditions for instabilities in ML parameterizations of moist convection when coupled with atmospheric fluid dynamics. In the development of a Fortran–Keras bridge for DL applications in atmospheric dynamics, Ott *et al.* [44] identified a strong relationship between offline validation error and online performance, in which the choice of optimizer proves unexpectedly critical. They used a hyperparameter search to find NN architectures that produce considerable improvements in stability, including some with reduced error. Rasp [45] proposed a coupled learning approach where a pretrained NN parameterization is run in parallel with a high-resolution simulation that is kept in sync with the NN-driven Earth System Model (ESM) through constant nudging. This approach attempts to curb the instabilities and biases previously experienced in embedding NN parameterizations into ESMs. Yuval *et al.* [46] achieved stable ML parameterizations of convection using random forests. In more recent work Yuval *et al.* [47] achieved stability with NNs by using a novel structure, careful coarse-graining and calculation of subgrid terms, and conditions to conserve mass and energy exactly. In forecasting, Chattopadhyay *et al.* [31] achieved a stable long-term forecast of geophysical turbulence using a novel equivariance-preserving spatial transformer architecture with custom losses.

(vi) Multi-scale properties and spectral methods

Weather and climate are complex systems with the Kolmogorov dissipation scale of millimetres up to the planetary scale of thousands of kilometres, and on all time scales from seconds to decades and longer. There are many nonlinear interactions across these scales that can lead to self-organization and emergent behaviour. It has for long been recognized that an important requirement for developing accurate weather and climate models is the ability to characterize accurately the multi-scale nature of these systems, in which small-scale, high-frequency (hours to days) variations play a key role in determining the large-scale, low-frequency (months to years) evolution of the system [48]. Spectral methods provide novel ways of incorporating the multi-scale properties of weather and climate systems in ML.

A common means of representing the scale distribution of a turbulent field is through its spectrum and covariance function. The spectrum and covariance function are related by the Fourier transform and its inverse. Wu *et al.* [18] captured the scale distribution and correlations across scales in a GAN model of turbulent convection by enforcing covariance constraints. These constraints also help preserve the spectral properties of the PDEs underlying the data. Mohan *et al.* [49] used the wavelet transform to inject physics-based features with a compact representation, the wavelet coefficients, for predicting a turbulent flow. Li *et al.* [50] developed a new neural operator in Fourier space, allowing for an expressive and efficient architecture that solved the Navier–Stokes equations up to three orders of magnitude faster compared to traditional PDE solvers. Tancik *et al.* [51] show that Fourier features help NNs learn high-frequency functions in low-dimensional domains, thus overcoming spectral biases. Recent novel approaches in computer vision have used spectral methods coupled with deep NNs, such as Fourier CNNs [52] and SpecNet [53], which could have implications for PIML model development.

(vii) Spatio-temporal coherence

Atmospheric and oceanic variability are characterized by strong spatio-temporal coherence across scales [54]. Spatio-temporal coherent structures are ubiquitous in atmospheric and ocean flows. They are considered the hidden skeletons that organize the rest of the flow into ordered patterns and modulate mixing, transport, and energetics [55]. Furthermore, they have important implications for extreme events [56]. In addition to capturing the multi-scale nature described in §2bvi, accurately representing the physics of the atmosphere and ocean requires that PIML models capture the spatio-temporal coherence that exists in them.

Simulating these systems using the governing equations naturally captures these coherent structures, such as atmospheric blocking, tropical cyclones, and the El Niño-Southern Oscillation. Although some ML models such as CNN use filters that learn coherent spatial patterns, there are no guarantees that ML models capture the underlying spatio-temporal coherent structures present in the data obtained from physical systems. Xie *et al.* [57] developed tempoGAN, a GAN augmented with an additional discriminator network that preserves temporal coherence for super-resolution of fluid flow. Li *et al.* [58] developed graph NNs that capture long-range interactions in NN solutions to PDEs. De Bezenac *et al.* [59] use a warping scheme based on the advection-diffusion equation that preserves spatial coherence in conjunction with a CNN to predict the evolution of sea surface temperatures. Recent developments in computer vision and pattern recognition have leveraged spatio-temporal coherence [60], which could have implications for PIML model development.

(viii) Physics-based modelling frameworks

Building upon the existing frameworks of physics-based models allows for integrating well-understood and scientifically sound model structures with data-driven learned components. Although many powerful methods based on theoretical reasoning have been used to develop weather and climate models, they often have simplifying assumptions and/or parameters that need to be determined empirically. ML offers novel approaches to replacing approximations or empirical parameters with data-driven learned counterparts while maintaining the original structure of the physics-based model. For example, the large-eddy simulation (LES) approach resolves the large-scale flow and models the impact of subgrid-scale (SGS) turbulence as a function of the resolved flow. The dynamic Smagorinsky model [61], a widely used model for atmospheric and oceanic turbulence, is a first-order closure model that relies on the eddy-viscosity assumption, which is only valid when there is sufficient scale separation. Recent work shows that ML models that learn the SGS stresses from data without invoking the eddy-viscosity assumption are more accurate and faster [62,63]. Numerical solutions to PDEs use discretizations, approximate coarse-grained representations, that are often ad hoc. Bar-Sinai *et al.* [64] use ML to learn discretizations based on actual solutions to the known underlying equations that are significantly more accurate and faster than existing methods. Several researchers have used ML for estimating and correcting errors in physics-based models, especially for forecasting and data assimilation. Pathak *et al.* [65] and Watson [66] used ML coupled with a dynamical system model for improved accuracy and forecasting horizons of chaotic systems. Bonavita & Laloyaux [67] used ML to extend current data assimilation capabilities in operational state-of-the-art forecasting systems. Farchi *et al.* [68] used ML to correct model error in data assimilation and forecasting.

During development, global climate models have their properties adjusted or tuned in various ways to best match the known state of the Earth's climate system. Developers typically perform this calibration by adjusting uncertain, or even non-observable, parameters related to processes not explicitly represented at the model grid resolution [69]. Key model properties, such as climate sensitivity, depend on frequently used tuning parameters. Mauritsen *et al.* [69] explain: 'The model tuning process at our institute is artisanal in character, in that both the adjustment of parameters at each tuning iteration and the evaluation of the resulting candidate models are done by hand, as is done at most other modelling centres. It is, however, at least conceptually possible to automate this process and find optimal sets of parameters with respect to certain targets.' ML

could be employed to find the optimal set of critical parameters in weather and climate models, as has been done with rigorous statistical inference to determine model coefficients in turbulence modelling [70] and using inverse methods and other statistical parameter estimation methods in weather and climate modelling [71–73]. However, uncertainties and limitations from the choice of physics-based model structure cannot be improved by ML [74].

(ix) Interpretability

Interpretable models provide transparency and are necessary to make PIML reliable and trustworthy. While there exists a large body of literature in interpretable ML [75–77], Rudin [78] argues that because interpretability needs to be defined in a domain-specific way, some of the most important technical challenges for the future of interpretable ML models are tied to the needs of specific domains.

Much work remains to be done in making PIML models for weather and climate science truly interpretable, however, initial progress shows great promise. McGovern *et al.* [79] and Ebert–Uphoff *et al.* [80] showed ways to interpret, visualize and evaluate ML models in meteorological applications. Gagne *et al.* [81] used feature importance and feature optimization to interpret their CNN model for predicting the probability of severe hailstorms and found that the model synthesized information about the environment and storm morphology that is consistent with our current understanding of the physics of hailstorms. Toms *et al.* [82] developed interpretable NNs for the geosciences and showed their usefulness and reliability in improving our understanding of the Madden-Julian oscillation [83]. Brenowitz *et al.* [43] developed an interpretability framework specialized for analysis of the relationship between offline skill versus online coupled prognostic performance for ML parameterizations of convection.

(x) Uncertainty quantification

Models that have their uncertainties characterized and quantified are critical for reliable decision- and policy-making for climate change mitigation and adaptation. Given the large number of assumptions, components and parameters of ML models, and uncertainties in the training data either from noise or data quality issues, UQ is a requirement for increasing the reliability of predictions, especially under distributional shifts and in out-of-sample scenarios. Though there is no PIML-specific method to employ, several UQ methods in ML could be employed.

Caldeira *et al.* [84] compare three popular UQ methods in ML applications in the sciences: Bayesian NNs, Concrete Dropout, and Deep Ensembles. In Bayesian NNs, the parameters are modelled as full probability distributions resulting in better calibrated confidence estimates and more robustness to adversarial and out-of-distribution examples [85]. However, modelling the full posterior distribution for the model's parameters given the data is usually computationally intractable. This high computational cost can be circumvented by dropout, where an approximate posterior distribution is obtained using variational inference [86]. Model uncertainty is estimated using dropout and predicting multiple times to obtain a spread of the different predictions. Deep ensembles, i.e. ensembles of NNs, can be used to obtain well-calibrated uncertainty estimates that are comparable to those obtained by Bayesian NNs [87].

Probabilistic ML models are amenable for efficient UQ, as shown by Zhu *et al.* [16] in using conditional generative models to model uncertainties in solving PDEs. Yang *et al.* [88] use adversarial UQ in NNs to construct a probabilistic ML model for a system governed by PDEs and use the model to characterize uncertainty to due noisy inputs. Daw *et al.* [22] use Monte Carlo dropout for UQ in lake temperature modelling. Vandal *et al.* [89] use Bayesian DL to downscale climate data with quantified uncertainties. Gagne *et al.* [36] demonstrate that GANs can be used as explicit stochastic parameterizations to model the uncertainties in subgrid processes directly from data. Schneider *et al.* [90] propose a blueprint for using ML to integrate observations and high-resolution simulations in Earth system modelling that systematically learn from both and quantifies uncertainties.

(c) Applications for physics-informed machine learning in weather and climate modelling

PIML is becoming increasingly important in at least three major applications in weather and climate modelling:

(i) emulating complex physical processes that are either poorly understood or not sufficiently well represented by existing models [46,91–95];
(ii) downscaling coarse data to produce high-fidelity high-resolution data [37,89,96,97]; and
(iii) forecasting the spatio-temporal dynamics of the atmosphere and ocean [31,59,98,99].

Other under-explored but promising applications include PIML-augmented PDE solvers [70,100] and discovery of governing equations [101].

3. Physics-informed machine learning: case studies in emulation, downscaling and forecasting

In this section, we introduce 10 case studies representing the three application areas in §2c that use the key PIML approaches described in §2b to address critical challenges in weather and climate modelling.

Table 1 characterizes the 10 case studies by PIML application/modelling task from §2c, physical processes modelled, datasets used, relevant PIML approaches used from §2b, ML model type, and PIML objectives achieved from §2a. The physical processes examined span a range of complexities from fundamental turbulent flows such as Rayleigh–Bénard convection to complex weather and climate phenomena such as clouds, precipitation, and melting of snowpack.

Section 3a contains four case studies on emulation, §3b contains three case studies on downscaling/super-resolution, and §3c contains three case studies on forecasting. In each case study, we emphasize the significance of the modelling challenge, identify the motivations for using PIML, describe how physics and domain knowledge is incorporated into the ML model, explain how uncertainties are quantified, highlight the key results, and summarize the implications of the study for the broader weather and climate science community.

(a) Emulating complex physical processes

Earth's weather and climate are characterized by a wide range of spatial and temporal scales with interactions across all scales. Resolving all the scales of weather and climate systems in simulations is prohibitively expensive. In practice, simulating these systems often involves closures, or parameterizations, to model unresolved processes (subgrid-scale physics) such as convection, clouds, and turbulence. However, these parameterizations also account for major sources of uncertainties in simulation results, partly due to neglecting high-order statistics of nonlinear subgrid-scale processes and their effect on the resolved scales [91]. PIML offers novel ways of leveraging existing high-fidelity simulation datasets to build models that can emulate all or a part of complex multi-scale processes, which can be used to augment or replace existing parameterizations in weather and climate models [36,46,47,91–95,106].

PIML also offers novel ways of emulating a chain of coupled processes, for example, the hydrological cycle, or even entire weather and climate models altogether. These emulators can offer ways to rapidly explore different scenarios with wide ranges of parameter values, to test different potential values of effective parameters, and for estimating parametric uncertainties [90].

(i) Constrained GANs to emulate turbulent Rayleigh–Bénard convection

Here, we review an approach that used constrained GANs to emulate turbulent convection, an important subgrid-scale process parameterized in weather and climate models. We begin with a

Table 1. Characteristics of PIML case studies examined in §3.

Section and case study reference	PIML application/ modelling task from §2c	Physical processes	Datasets	PIML approaches from §2b	ML model	PIML objectives achieved from §2a
§3ai: Wu *et al.* [18]	emulation	Rayleigh-Bénard convection	direct numerical simulation (DNS)	custom loss, stochasticity, multi-scale, spectral	GAN	physically consistent, accelerated training
§3aii: Manepalli *et al.* [102]	emulation	mountain snowpack melting (hydro-climate)	hydro- meteorological observational product	custom loss, stochasticity, UQ	conditional GAN	physically consistent
§3aiii: Daw *et al.* [22]	emulation	lake temperature dynamics	observational lake characteristics data	custom architecture, UQ	LSTM	physically consistent, data efficient, interpretable
§3aiv: Beucler *et al.* [14]	emulation	atmospheric convection and clouds	super-parametrized community atmosphere model	custom loss, custom architecture	NN	physically consistent, generalizable
§3bi: Singh *et al.* [103]	downscaling/ super-resolution	atmospheric winds	weather research and forecasting model	custom loss, stochasticity, spectral	GAN	physically consistent, generalizable
§3bii: Vandal *et al.* [89]	downscaling/ super-resolution	precipitation	reanalysis product (PRISM)	custom architecture, stochasticity, UQ	Bayesian NN and CNN	physically consistent, data efficient
§3biii: Jiang *et al.* [104]	downscaling/ super-resolution	Rayleigh-Bénard convection	DNS	custom loss, custom architecture, multi-scale, spatio-temporal coherence	encoder-decoder NN	physically consistent, generalizable, scalable
§3ci: Wang *et al.* [105]	forecasting	Rayleigh-Bénard convection	DNS	custom loss, custom architecture, multi-scale, physics-based structure	encoder-decoder NN	physically consistent, generalizable, interpretable
§3cii: Wang *et al.* [30]	forecasting	Rayleigh-Bénard convection, Ocean currents	DNS, ocean reanalysis product (ORAS5)	custom architecture, equivariant, spatio-temporal coherence	residual network	physically consistent, data efficient, generalizable
§3ciii: Chattopadhyay *et al.* [31]	forecasting	geophysical turbulence	DNS	custom loss, custom architecture, equivariant, spatio-temporal coherence	spatial transformer network	physically consistent, data efficient, stable

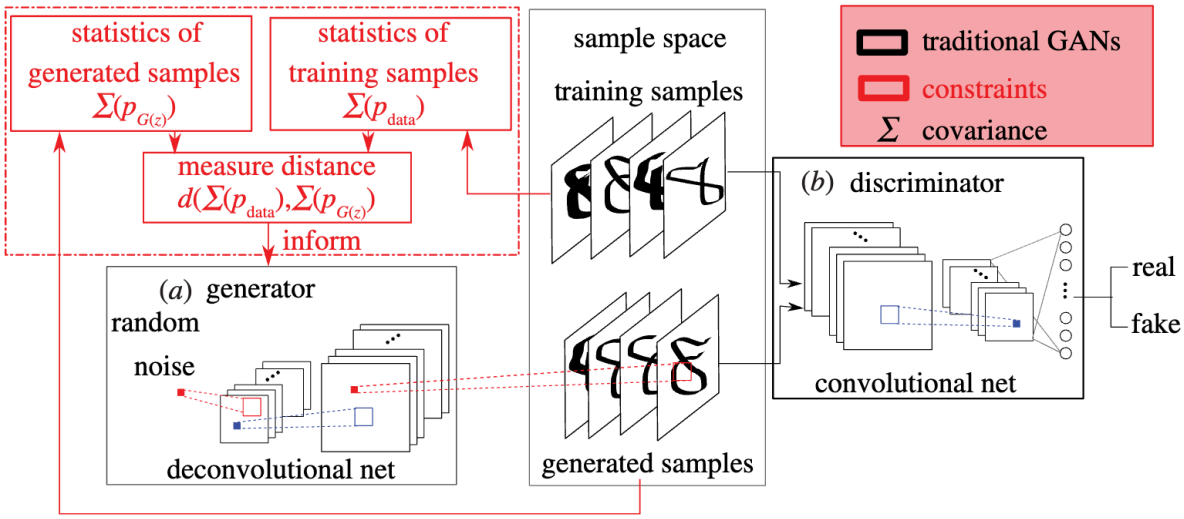

Figure 1. The architecture of a constrained GAN, including the architecture of a standard GAN (black) and the modification to help preserve high-order statistics incorporated via a custom-designed loss function (red). Figure reproduced from [18]. (Online version in colour.)

brief introduction to GANs. GANs can emulate the behaviour of a complex system by mimicking the data distribution it is trained on [107]. GANs are formulated as a zero-sum game between two deep NNs, a generator G and a discriminator D. Figure 1 shows a schematic of the DL architecture. In the standard setting, G receives a random noise vector z drawn from a simple distribution (such as a uniform or Gaussian) as input, which it passes through a succession of deconvolutional layers and nonlinear transforms to output a sample $G(z)$. The role of D is to act as a classifier, deciding if a sample it receives is either real or fake (generated by G). After training, G is ideally able to produce 'fake' samples that are implicitly drawn from the data distribution that G seeks to emulate.

Recent research has shown how GANs can be used to generate new solutions of PDE-governed systems by training on simulation datasets and can capture several desirable physical and statistical properties of turbulent flows [108]. GANs can, however, be notoriously difficult to train because of instabilities in training from sensitivity to hyper-parameters, challenges in convergence, generation of noisy samples, and can suffer from mode collapse, where they only generate samples from a single mode of the true multimodal data distribution [109,110]. Several approaches have been proposed that incorporate domain knowledge to alleviate some of the above challenges and improve the performance of GANs for physical problems, using customized loss functions and modified architectures, e.g., Gagne *et al.* [36] uses GANs with temporal coherence for stochastic emulation of subgrid scale dynamics, Xie *et al.* [57] incorporate temporal coherence to GANs to generate realizations of turbulent flows, Yang *et al.* [111] encode the governing physical laws in the form of stochastic differential equations into the architecture of GANs, and Stinis *et al.* [112] incorporate constraints to enhance the interpolation and extrapolation capabilities of GANs.

How is physics incorporated? In this study, Wu *et al.* [18] incorporate high-order statistical constraints as a novel regularizer in their GAN-based emulator [18]. More precisely, the covariance structure, i.e. the second-order moment of the training data distribution, is enforced by introducing a penalty term in the loss function. The introduced penalty term accounts for the difference between the covariance structures of the generated samples and the training data. As described in §2bvi, covariance constraints help capture the scale distribution and correlations across scales, and preserve the spectral properties of the PDE underlying the data.

The physical system investigated in this study is Rayleigh–Bénard convection (RBC), a canonical buoyancy-driven turbulent flow which is an idealized model for atmospheric convection. Details on the simulations and datasets used for training are in [18]. Figure 2 shows

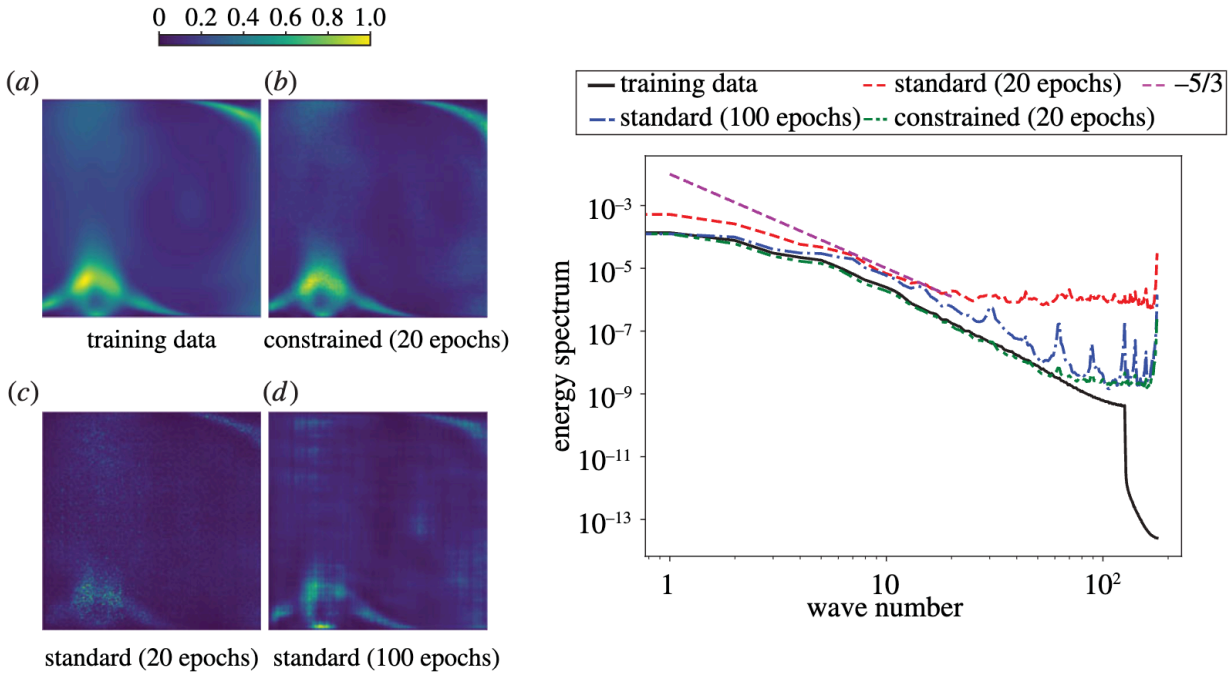

Figure 2. A comparison between the training data (truth), a standard GAN trained up to 20 epochs and 100 epochs, and the constrained GAN trained up to 20 epochs. Left: (*a–d*) time-averaged turbulent kinetic energy fields over a square spatial domain of size 256 × 256. Right: turbulent kinetic energy spectra. The −5/3 line is predicted by theory. The constrained GAN captures the spectrum at all except the highest wavenumbers, i.e. the finest scales of the flow. Figure reproduced from [18]. (Online version in colour.)

a comparison between the training data (truth), a standard GAN trained up to 20 epochs and 100 epochs, and the constrained GAN trained up to 20 epochs. On the left are turbulent kinetic energy fields and on the right are power spectral densities (PSD) of turbulent kinetic energy, a metric that incorporates information across all spatial scales. The constrained GAN outperforms its unconstrained counterpart by generating more accurate turbulent kinetic energy fields and energy spectra. The PSD shows that the entire range of scales is accurately captured, thus achieving physical consistency.

How is uncertainty quantified? Although UQ was not performed in this study, the stochastic nature of GANs allows for several ways of estimating uncertainty. Gagne *et al.* [36] provide an insightful discussion on how GANs can be used to model uncertainties in SGS processes. Furthermore, in §3aii, we discuss approaches for UQ using generated samples or injecting noise into intermediate layers of G to calculate distributions for the application of statistical tests and to calculate confidence intervals.

What are the key implications? First, the results show that constrained GANs emulate the statistics of the training data better than their unconstrained counterparts, indicating that the statistical constraint leads to better convergence towards the global minimum, where all statistics of the training data can be captured. Second, constrained GANs achieve greater accuracy at significantly lower computational cost (up to 80% reduction of computational cost in model training) compared with the unconstrained model. In effect, the statistical constraint reduces the space of allowable solutions, forcing the training procedure to explore only a reduced solution space where the second-order moment of generated samples is similar to that of the training data.

With the growth of high-fidelity simulation databases of turbulence, weather, and climate, this work shows that physics-constrained GANs that preserve high-order statistics emulate complex multi-scale systems well, can model stochasticity and uncertainties from data, and could be promising alternatives to subgrid-scale closure models or parameterizations for unresolved physics.

(ii) Conditional GANs to emulate numerical hydro-climate models

Here, we review the potential of conditional GANs (cGAN) to emulate a physics-based model of the spatial distribution of the water content of mountain snowpack, or snow water equivalent (SWE). Snowpack and SWE are key indicative variables for investigating the changing water cycle and its impact on nature, society and the economy. Acquiring SWE data via direct observations is extremely difficult in mountainous regions and very expensive. Furthermore, for process-based models the uncertainties in the meteorological variables that influence SWE, such as temperature, wind velocities, humidity, net radiation, and precipitation, are known to result in SWE predictions with extremely large uncertainties. Hence, ML-based alternatives to SWE prediction that characterize and estimate uncertainties reliably are attractive.

Manepalli *et al.* [102] formulate the emulation problem as an image-to-image translation task where the goal is to transform an image from domain X, gridded meteorological variables, to domain Y, SWE grids based on pix2pix, a general-purpose solution to image-to-image translation problems [113]. In this setting, training samples from the two domains X and Y are assumed paired. Furthermore, G takes a noise vector as an additional input, and can generate distributions of realistic and plausible SWE maps for individual days through sampling. Details on the training datasets are in [102].

How is physics incorporated? Domain knowledge is incorporated into the cGAN via a custom-designed loss function as follows: (i) areas of higher elevation typically have larger amounts of snow (and therefore SWE), and the cGAN is penalized for large errors in such areas accordingly; (ii) as a significant portion of the data covers water bodies such as the Pacific ocean, where no snowpack can exist, the model is penalized for placing SWE values in these areas; and (iii) the difference in total SWE between cGAN solutions and physics model output is also penalized, to ensure that total stored water mass is properly estimated. All three custom losses are weighted equally. Figure 3 shows the model architecture of the cGAN.

Histograms of normalized pixel values in figure 3*a* show that the generated data distribution matches the real data distribution well. The PSD plot in figure 3*b* shows that the large scales are captured well by the cGAN, with a small discrepancy at the small scales (high-spatial frequency).

An ablation study was performed, where custom losses were removed systematically, one at a time, in order to understand their relative importance in accuracy and convergence of the cGAN. The authors found that the custom loss for higher elevation was the most crucial; RMSE increased by 40% when that loss was removed and extreme events, i.e. the tails of the distribution, were not captured.

How is uncertainty quantified? Injecting noise into several layers of G allows for the generation of diverse but realistic and physically plausible SWE grids. Furthermore, sampling can be performed at individual test points, allowing for the creation of SWE distributions for the application of statistical tests and confidence intervals. For one test location, figure 3*c* shows that the resulting SWE distributions formed by sampling from stochastic G (blue) are centred around the prediction of the physics-based numerical model, Livneh (green). Large deviations from the average SWE value of SNOTEL (observation) are to be expected, as the numerical model is gridded at a 4 km resolution, whereas SNOTEL is measured with a single small pressure sensor. A more complete description of uncertainty and reliability of predicted distributions would require a reliability diagram, as discussed in §3aiii.

Finally, during inference, the cGAN has a 250× speedup over the numerical model enabling previously intractable studies such as probabilistic risk assessment and sensitivity analysis.

What are the key implications? These results indicate that the physics-constrained cGAN model is able to effectively learn diverse mappings between meteorological forcings and SWE output, thus providing a means for fast and accurate SWE modelling that can have a significant impact in a variety of applications such as hydropower forecasting, agriculture and water supply management [114]. The massive speedups, diverse sampling, and sensitivity/saliency modelling that cGANs can bring to process emulation, along with methods for UQ, show promise for investigating the impacts of climate change using cGANs-based emulators.

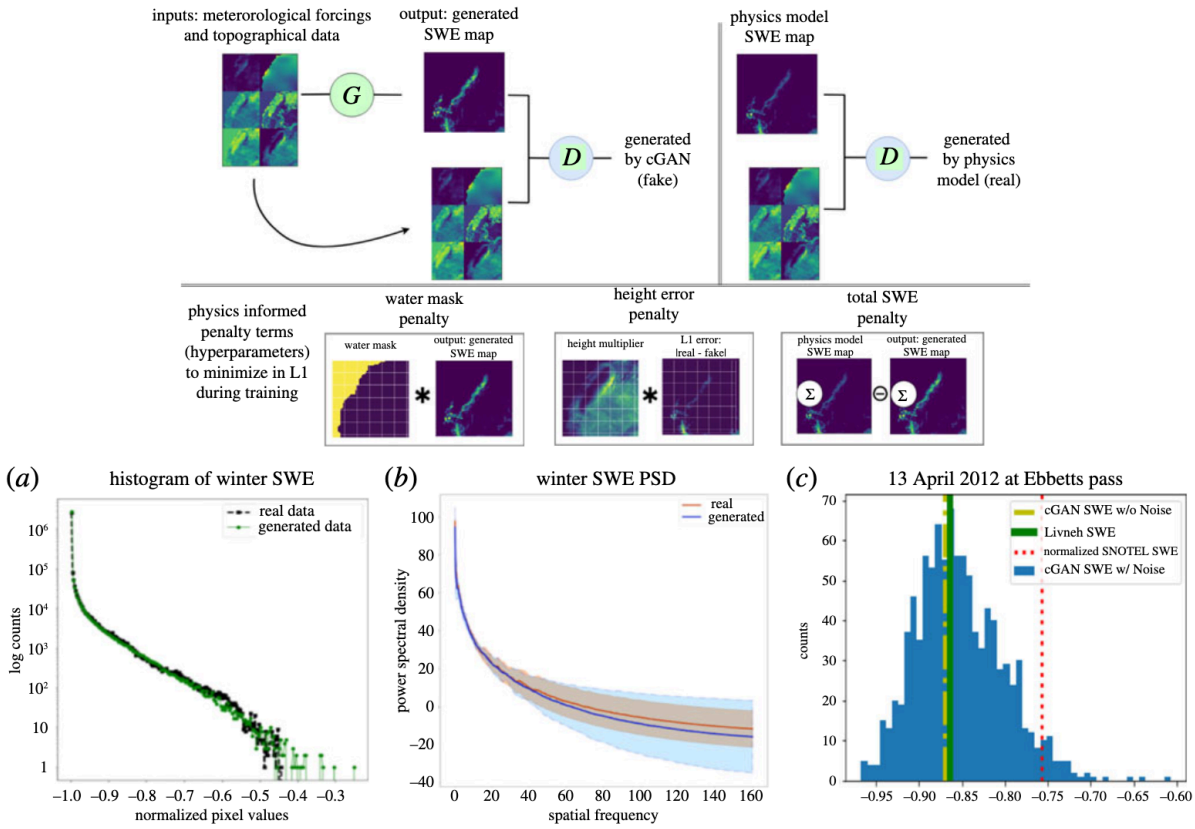

Figure 3. Top: Architecture of the cGAN. Bottom: (*a*) Histograms (densities on log scale) of normalized pixel values comparing cGAN (green) and physics model (black), normalized by the winter range of SWE, i.e. (min–max); (*b*) Power spectral density of cGAN and physics model; (*c*) Distributions of SWE (Blue) formed by sampling from stochastic G compared with sample from non-stochastic G (yellow), Livneh training data (green) and observational data from nearby SNOTEL Station (Red) on 13 April 2012 (a single sample time). *X*-axis is normalized SWE pixel values. Large deviations from the average SWE value of the pixel are to be expected, as the numerical model is gridded at a 4 km resolution, whereas SNOTEL is measured with a single small pressure sensor. Figure reproduced from [102]. (Online version in colour.)

(iii) Physics-guided neural network for lake temperature modelling

Predicting spatial and temporal characteristics of lake temperatures is critical for understanding ecological, aquatic, and biogeochemical processes and the impact of climate change on fresh water [115]. Accurate physics-based models of lake temperature, which require modelling the many complex processes that are coupled to each other, are too expensive. Observations of water temperatures are difficult or impossible at broad spatial and temporal scales. Hence ML models that could offer faster and potentially more accurate solutions are essential.

Here, we review a novel physics-guided architecture (PGA) of NN proposed by Daw *et al.* [22] to model lake temperatures and integrate UQ. They formulate the problem of lake temperature modelling as a spatio-temporal sequential prediction problem. Their PGA has three components: an autoencoder to extract temporal features, a physics-based LSTM model, and a multi-layer perceptron to predict the new spatio-temporal sequence of temperature.

How is physics incorporated? The physics-based LSTM predicts an intermediate variable, the density. Since the density can only increase with depth, it must be monotonic. The LSTM model is constrained to only predict positive density increments with increasing depth. The details of the implementation are in [22].

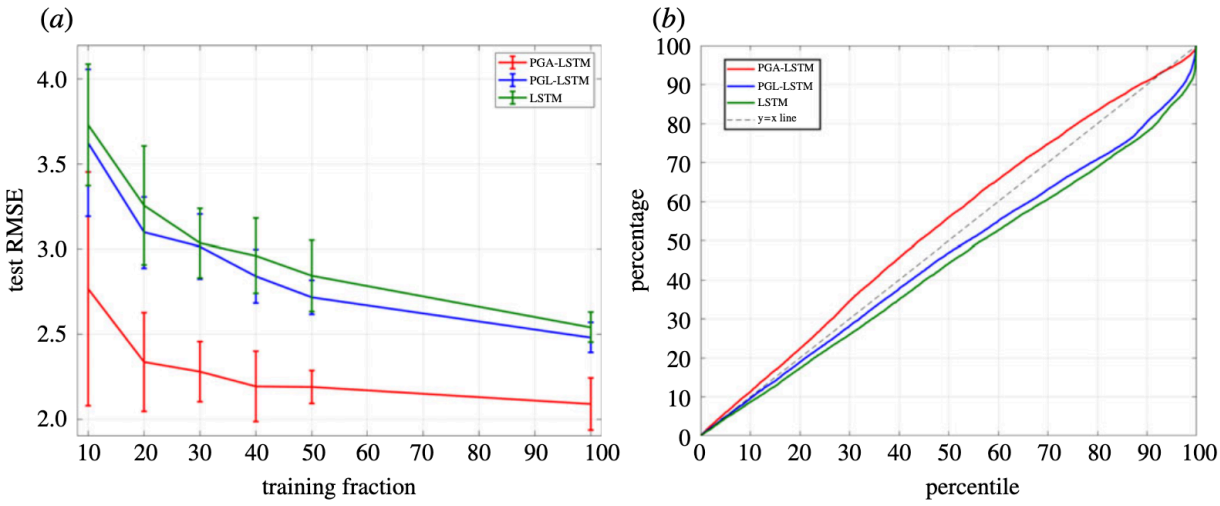

Figure 4. (*a*) Test RMSE (per sample) on varying training sizes; (*b*) Cumulative percentage of observations within a certain percentile of samples of comparative models. Figure reproduced from [22]. (Online version in colour.)

Table 2. Test RMSE and physical inconsistency (the fraction of times the Monte Carlo sample predictions at consecutive depths are physically inconsistent, i.e. they violate the density-depth relationship) using 40% of training data. Table reproduced from Daw *et al.* [22].

	test RMSE (in °C)		physical inconsistency	
	per sample	mean	per sample	mean
LSTM	2.96 ± 0.22	2.27 ± 0.17	0.28 ± 0.02	0.07 ± 0.03
PGL	2.84 ± 0.16	2.12 ± 0.13	0.27 ± 0.02	0.08 ± 0.03
PGA	2.19 ± 0.21	1.88 ± 0.12	0.00 ± 0.01	0.00 ± 0.00

How is uncertainty quantified? Uncertainty is quantified by using dropout in the testing phase to produce Monte Carlo samples of the target variable for every test instance—a technique called Monte Carlo dropout [86].

The baseline models used for comparisons are a standard LSTM model and an LSTM with a physics-guided loss (PGL) that penalizes non-monotonic density changes [13]. Shown in table 2 are comparisons of the performance of the proposed PGA versus baselines, a standard LSTM and PGL. Standard architectures produce physically inconsistent solutions even with a physics-guided loss. The authors argue that the randomness injected into the trained weights of the NN during dropout is sufficient to unlearn the physical consistency introduced by the physics-guided loss during training. In contrast to the baselines, the proposed PGA shows the smallest RMSE per test sample while always preserving physical consistency, even after performing Monte Carlo dropout.

Furthermore, by training on varying sample sizes, they show that the PGA-LSTM has the lowest RMSE across all values of training fractions. The goal of this is to simulate realistic scenarios on lakes where little or no observational data exists. As seen in figure 4*a*, the PGA achieves RMSE values comparable to the standard LSTM with about an order of magnitude smaller amount of training data. This shows that the novel PGA is highly data efficient.

To assess uncertainty estimates using the Monte Carlo dropout method, the cumulative percentage of ground-truth observations that fall within a certain percentile of samples generated by comparative models are shown in figure 4*b* (a reliability diagram). The ideal model is represented by the diagonal line $y = x$, where the percentage of ground-truth points within a percentile is equal to the percentile value. Models that are over-confident would have fewer

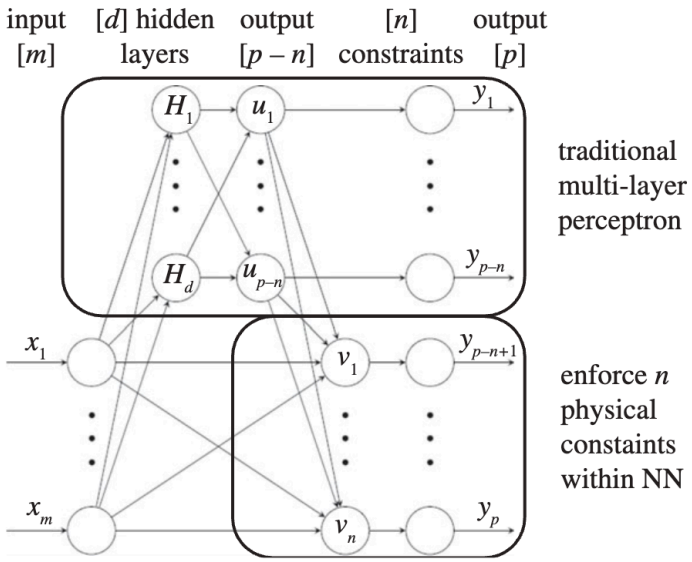

Figure 5. Architecture-constrained configuration: NN augmented with *n* conservation layers. Figure reproduced from [14].

ground-truth points within a certain percentile and hence would lie below the diagonal. PGA tends to be slightly under-confident in its uncertainty estimates whereas the baselines are over-confident in their uncertainty estimates, i.e. the distribution of ground truth points sometimes falls outside the distribution of Monte Carlo samples. The larger uncertainty estimates of the proposed architecture are desirable, especially for unseen scenarios.

What are the key implications? The results above show that custom-designed architectures are powerful ways to enforce constraints that achieve physical consistency and data efficiency. Furthermore, this work shows how introducing intermediate variables that are physical and interpretable helps make the PIML model more transparent. Finally, Monte Carlo dropout is shown to provide a simple yet effective means of constructing a more robust model with uncertainties quantified.

(iv) Enforcing conservation laws in neural networks for climate modelling

Conservation of mass, momentum, and energy are critical for climate change studies because the climate system is highly sensitive to mass, momentum, and energy imbalances. Conservation laws are often expressed as PDEs, including nonlinear PDEs such as the Navier–Stokes equations. Conservation laws are also fundamental for a variety of applications of ML in other physical systems, yet there do not exist general approaches for enforcing these laws in ML models.

Here, we review a novel and systematic method developed by Beucler *et al.* [14] to strictly enforce conservation laws.

How is physics incorporated? Beucler *et al.* [14] augment a standard NN with *n* conservation layers to enforce the conservation laws *C* to machine precision. The feed-forward network outputs an 'unconstrained' vector of size $p - n$, where p is the size of final output vector required. The remaining component of the output vector is calculated using the *n* constraints imposed via the *n* conservation layers. Figure 5 shows their architecture. The MSE loss is calculated over the entire output that concatenates the output of the original NN with the output from the conservation layers, which are exact residuals from the constraints. Because the full output vector is used in the NN training process and the gradients of the loss function are passed through the conservation layers during optimization, this approach is fundamentally different from simply calculating a part of the output as a post-processing step using the conservation laws as constraints.

Beucler *et al.* [14] apply this approach to a NN emulator of convection for climate modelling. The goal of the NN is to predict the effect of cloud processes on climate, i.e. the radiative and convective tendencies, based on inputs that represent the climate state, i.e. the large-scale

Table 3. Mean-squared error (skill) and physical constraints penalty P (violation of energy/mass/radiation conservation laws) for different models in units W^2/m^4 in the format (mean $\pm$ s.d.). Table reproduced from Beucler *et al.* [14].

validation	metric	MLR	NNU	NNL	NNA
baseline	MSE	$295 \pm 1.7 \times 10^3$	$156 \pm 1.0 \times 10^3$	$177 \pm 1.1 \times 10^3$	$169 \pm 1.0 \times 10^3$
(+0 K)	P	$28 \pm 2 \times 10^1$	$458 \pm 5 \times 10^2$	5.0 ± 5	$7 \times 10^{-10} \pm 1 \times 10^{-9}$
Cl. change	MSE	$747 \pm 1 \times 10^5$	$633 \pm 7 \times 10^3$	$496 \pm 8 \times 10^3$	$567 \pm 8 \times 10^3$
(+4 K)	P	$265 \pm 2 \times 10^3$	$3 \times 10^5 \pm 1 \times 10^6$	$470 \pm 2 \times 10^3$	$2 \times 10^{-9} \pm 5 \times 10^{-9}$

thermodynamic variables. The conservation laws imposed are: conservation of mass, enthalpy, terrestrial radiation, and solar radiation. They compare results against three baseline models: a multiple-linear regression model (MLR), an unconstrained NN (NNU), and NN with an additional penalty in the loss function equal to the residual from the constraints, i.e. an NN with 'soft' constraints (NNL). Details of the training data and procedure are in [14].

Tests are performed on the present day's climate, similar to the training data, and on a climate change scenario with 4K warming. Table 3 compares the performance measured by MSE and physical inconsistency, defined as the degree to which the conservation laws are violated, measured by the penalty, P. α is the relative weight of the penalty P to the standard MSE loss. NNU has low MSE but strongly violates the conservation laws. NNL performs reasonably well with a lower penalty than its unconstrained counterpart. The architecturally constrained NN satisfies the conservation laws to machine precision. Importantly, table 3 shows that NNL and NNA perform better than NNU on a climate change scenario, suggesting that physically constrained NNs ('soft' or 'hard') generalize better to unseen scenarios. Although NNA has higher MSE than NNL in the climate change scenario, in subsequent work the authors show that NNA can achieve conservation without degrading performance [17] (table 3).

What are the key implications? This work shows that enforcing strict constraints via custom-designed NN architectures for the conservation of physical quantities, a critical requirement in weather and climate modelling, guarantees physical consistency and improves generalizability. In subsequent work, the authors extend their approach to more general and broader classes of analytic constraints, including nonlinear constraints and inequality constraints [17].

(b) Downscaling (super-resolving) coarse data

Accurate and reliable high-resolution weather and climate data are essential for understanding scientific phenomena better and for a wide range of climate impact studies, planning and policy-making under climate change. This is especially important in the event of highly localized phenomena such as weather and climate extremes, in urban areas, and in regions with high topographic complexity and sharp gradients like mountains or coastal regions. However, fully resolving these complex systems in conventional numerical weather and climate models is intractable and most observational datasets do not contain reliable information at the fine scales. Therefore, there is a pressing need for efficient and accurate methods to enhance the resolution of weather and climate data.

Enhancing the resolution of weather and climate data, so-called downscaling, can be done using dynamical or statistical approaches. Dynamical downscaling techniques use high-resolution regional models, where coarse data are used as boundary and initial conditions, for dynamically predicting the effects of large-scale climate processes on regional or local scales of interest. Dynamical downscaling techniques are generally more reliable because they are physics-based, but are computationally too expensive. By contrast, statistical downscaling is cheap and fast, but suffers from poor generalizability. Some traditional statistical downscaling methods also tend to smooth out small-scale features [97].

Super-resolution (SR) is the process of taking a low-resolution (LR) image and producing an enhanced image that approximates the true high-resolution (HR) version of it [116]. SR includes bilinear or bicubic interpolation, which are simple but tend to significantly smooth out small-scale features and sharp gradients. ML-based SR approaches offer novel ways of resolution enhancement by learning complex mappings between pairs of LR/HR images [117]. Criteria for successful SR include realistic small-scale features at the HR, both perceptually and physically; and rigorous quantitative validation of an HR image using physically relevant metrics.

Vandal *et al.* [118] develop DeepSD, a generalized stacked SR convolutional neural network (SRCNN) framework, for statistical downscaling of climate data that outperforms several traditional statistical downscaling methods. Stengel *et al.* [97] develop an adversarial DL approach for SR and show promising results on enhancing the resolution of climate data by a factor of 50. SR is a one-to-many problem, since one LR image could be mapped to many HR images. Hence probabilistic ML models with UQ are preferred. Groenke *et al.* [37] develop a novel unsupervised statistical downscaling model using normalizing flows, a type of generative ML model, that allows for both conditional and unconditional sampling from the joint distribution over high and low resolution spatial fields.

The holy grail of SR is spatio-temporal SR, where both the spatial and temporal resolutions are enhanced to produce physically accurate HR data that satisfies the governing laws of the system, has physically accurate and realistic small-scale features, and is coherent in space and time.

(i) Physics-constrained GAN for super-resolution of weather data

Here, we review a DL-based SR method that produces high-fidelity output fields by using a physical constraint that encodes the multi-scale features of the system. Singh *et al.* [103] use a modification of the enhanced SR GAN (ESRGAN) architecture. ESRGAN is a conditional GAN designed for SR. It contains three losses: an adversarial loss, a 'content loss' between the generated data and true HR data, and a 'perceptual loss' [119]. The 'content loss' is an MSE loss computed using an L2 norm between the generated data and true HR data. The 'perceptual loss', motivated by work in image processing and computer vision, is a feature-based loss constructed from a previously trained auxiliary network that identifies critical features in image data from intermediate layers of the network to improve the perceptual quality of the enhanced output.

How is physics incorporated? ESRGAN is modified by replacing the adversarial loss with a PSD loss. The PSD loss penalizes errors in the energy spectrum of the generated images by comparing against the spectrum of the ground truth data. As discussed in §3ai and §3aii, capturing the energy spectrum accurately implies that the range of spatial scales is characterized accurately. Furthermore, because PSD is a differentiable function, it allows for optimization using back-propagation [103]. The authors refer to this model as PSD-Net. Furthermore, direct optimization of the spectra accelerates training as it is stable with larger batch sizes and does not require training a discriminator.

SR is performed on 15 years of wind velocity fields from a numerical simulation of the Weather Research and Forecasting (WRF) model over southern California. The spatial resolution of the data is 1.5 km and temporal resolution is hourly. SR enhances the spatial resolution by 4× in each dimension (see [103] for more details on the dataset and the training procedure). The proposed physics-based SR method, PSD-Net, is compared against three baselines: (i) the standard ESRGAN; (ii) SR-CNN, a CNN architecture used by Vandal *et al.* [118]; and (iii) upsampling using bicubic interpolation. Figure 6*a* shows a schematic of a GAN for SR.

Table 4 compares performance on the validation set. The metrics for comparison are peak signal to noise ratio (PSNR), MSE, mean absolute error (MAE) and Kullback–Leibler (KL) divergence between the empirical distributions of the generated images and the ground truth. PSNR, MSE and MAE are averaged over all images in the validation set. PSD-Net performs best on all metrics.

The PSD plot in figure 6*b* shows that both the standard ESRGAN and PSD-Net capture the range of scales accurately, whereas SR-CNN and bicubic interpolation drop significantly

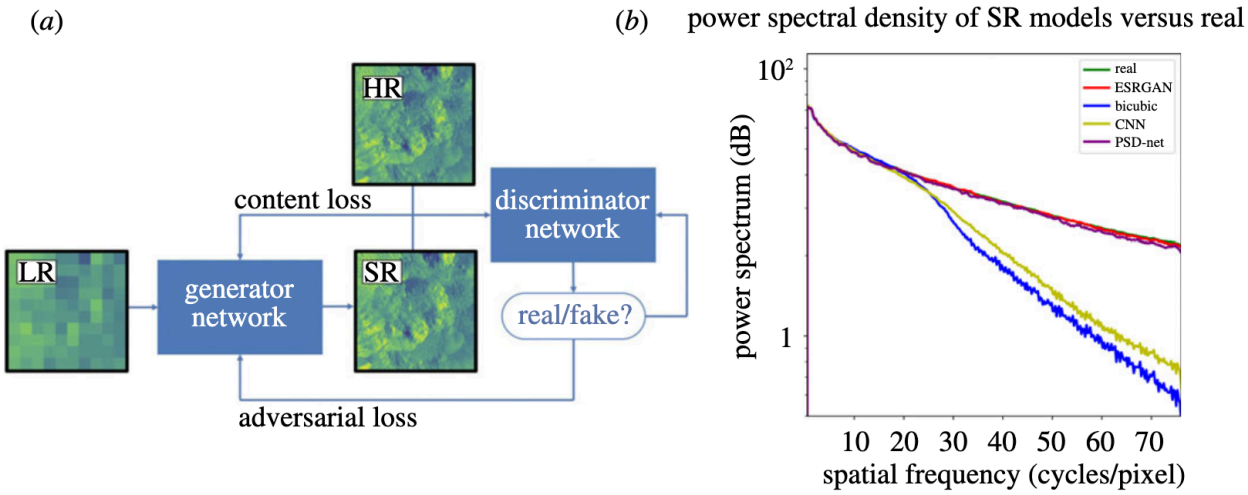

Figure 6. (*a*) Schematic of GAN for SR. Figure reproduced from [97]; (*b*) Power spectral density (PSD) plot of SR methods compared. Figure reproduced from [103]. (Online version in colour.)

Table 4. Overview of final performance on the validation set: PSNR (peak signal to noise ratio), MSE, MAE (mean absolute error) and KL, Kullback-Leibler divergence between the empirical distributions of the generated images and ground truth. PSNR, MSE and MAE are averaged over all the images in the validation set. Table reproduced from Singh *et al.* [103].

model	PSNR	MSE	MAE	KL
ESRGAN	32.74	5.3×10^{-4}	0.0148	0.008
SR-CNN	36.06	2.4×10^{-4}	0.0091	0.015
Bicubic	35.52	2.7×10^{-4}	0.0097	0.006
PSD-Net	39.3	1.1×10^{-4}	0.0066	0.005

at intermediate and high spatial frequencies, i.e. the fine scales are smoothed out. ESRGAN learns the data distribution at all scales because the adversarial training preserves physically relevant characteristics. Direct optimization of the spectra in PSD-Net helps reproduce the spectra faithfully.

Figure 7 compares images of the LR input, high-resolution ground truth (HR), and generated SR outputs from PSD-Net, ESRGAN, SRCNN and bicubic upsampling. Although ESRGAN performs poorly on PSNR, MSE and MAE, the generated images reveal that both PSD-Net and ESRGAN produce sharper images that have more realistic small-scale features and are less prone to artefacts.

How is uncertainty quantified? Although UQ was not performed in this study, as discussed in §3ai and §3aii, the stochastic nature of GANs allows for several ways of estimating uncertainty. Furthermore, performance on the tails of the distribution, i.e. extreme events, for generated SR images can be characterized using various statistical tests, including the reliability diagram as shown in §33aiii.

What are the key implications? This work shows that a novel physics-constrained DL SR model derived from ESRGAN is able to efficiently and effectively learn to produce hi-resolution, hi-fidelity data with fine-scale features that are realistic and physically consistent. This approach shows a means for fast and accurate SR that can have significant impact in a variety of weather and climate applications.

(ii) Bayesian deep learning with UQ for downscaling precipitation

UQ is essential for the development of robust and reliable PIML models. Here we review work by Vandal *et al.* [89] that builds upon their DL-based super-resolution model for downscaling

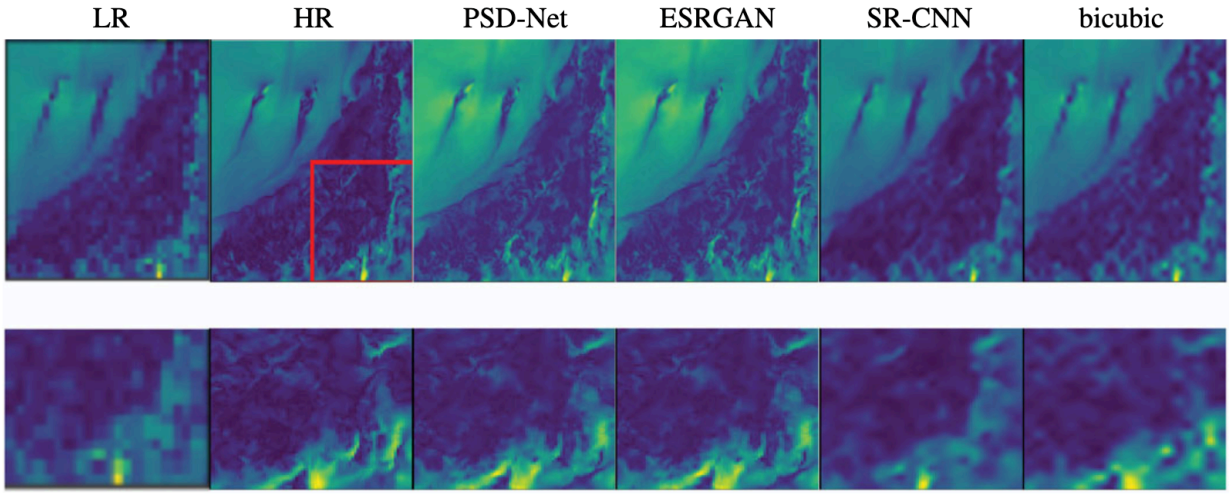

Figure 7. Comparison of low-resolution (LR), high-resolution ground truth (HR), and generated SR outputs from PSD-Net, ESRGAN, SRCNN and bicubic upsampling. The lower panel corresponds to the area of the red box in the upper panel. Figure reproduced from [103]. (Online version in colour.)

precipitation [118]. They use Bayesian deep learning (BDL) models to systematically characterize and estimate uncertainties.

How is physics incorporated? Like many weather and climate phenomena that follow non-normal distributions, the distribution of precipitation is highly skewed with fat tails because most days have no precipitation at all and few rainy days have large or even extreme precipitation. Furthermore, precipitation exhibits extreme space–time variability as well as intermittence. The authors use a discrete-continuous BDL model with lognormal likelihoods to model the highly skewed distribution of precipitation.

DeepSD is an adaptation of a CNN-based SR model, SRCNN, which performs pixel-wise regression [118]. DeepSD uses skip connections and an auxiliary variable, elevation, to correct for biases. Furthermore, by stacking multiple SRCNNs, DeepSD can achieve resolution enhancements as large as 16×. They formalize the use of BDL within the DeepSD architecture assuming a normal predictive distribution and a conditional discrete-continuous (DC) model with Gaussian and lognormal likelihoods. The DC models condition the amount of precipitation given an occurrence of precipitation. They also derive the corresponding losses and unbiased parameter estimates for their BDL models. Details of the mathematical derivation and implementation in their DL framework are in [89].

Precipitation data is obtained from the PRISM dataset, a reanalysis product at 4 km resolution, and coarsened to lower resolutions to generate training data. The SR problem is to enhance the spatial resolution from 64 km to 16 km across the contiguous USA. Three BDL models are used for comparisons: (i) BDL with a normal distribution; (ii) a discrete-continuous (DC) model with a Guassian distribution; and (iii) a DC model with a lognormal distribution and log-likelihood. For further details on the training procedure refer [89]. The models are evaluated on several metrics: RMSE, bias, and two extremes indices: heavy wet days with rainfall greater than 20 mm/day (R20) and daily intensity index defined as the annual total rainfall divided by the number of days with rainfall over 0.5 mm/day (SDII). Table 5 shows that DC models perform better, and in particular, DC-Lognormal shows the lowest bias, RMSE, and R20 error while DC-Gaussian has slightly higher errors but performs marginally better at estimating the SD II index.

How is uncertainty quantified? UQ is especially important for downscaling/SR as it is a one-to-many problem. Recent developments in BDL provide ways to capture uncertainties from noisy observations and from unknown model parameters. Vandal *et al.* [89] use a practical variational approach to approximate the posterior distribution in DeepSD using dropout and Monte Carlo

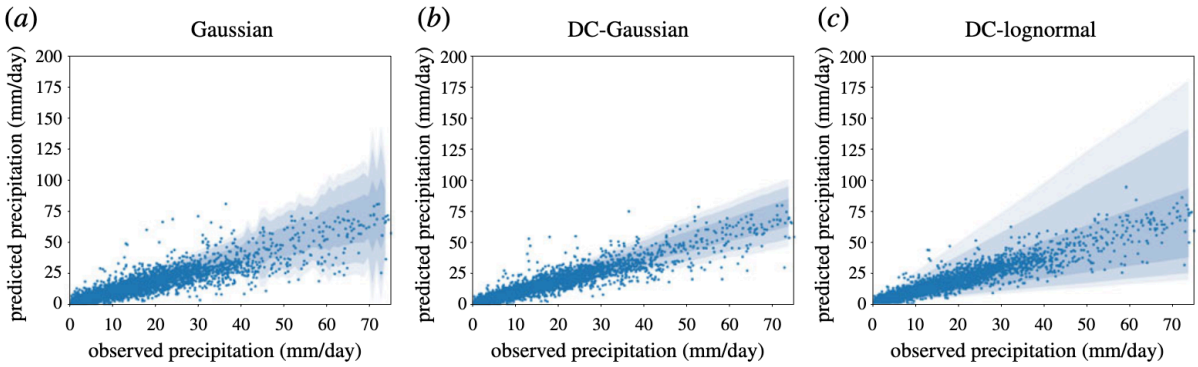

Figure 8. Uncertainty widths based on quantiles from their predictive distributions. The points are observations versus the expected value. The bands correspond to 50%, 80% and 90% predictive intervals. Figure reproduced from [89].

Table 5. Predictive accuracy statistics computed pixel-wise and aggregated. Daily intensity index (SDII) and yearly precipitation events greater than 20 mm (R20) measure each model's ability to capture precipitation extremes. R20 Error and SDII Error measures the difference between observed indices and predicted indices (closer to 0 is better). Table reproduced from Vandal *et al*. [89].

	bias	RMSE	R20 error	SDII error
Gaussian	-0.11 ± 0.34	2.14 ± 1.31	-0.73 ± 1.94	-0.83 ± 0.93
DC-Gaussian	-0.11 ± 0.30	2.07 ± 1.28	-0.61 ± 1.67	-0.21 ± 0.78
DC-Lognormal	-0.02 ± 0.30	2.05 ± 1.27	-0.36 ± 1.63	-0.28 ± 0.81

sampling, as described in [84]. The first two moments are derived and used to estimate pixel-wise probabilistic estimates. The calibration metric used for UQ is the frequency of observations occurring within a varying predicted probability range.

Figure 8 shows uncertainties for increasingly intense precipitation days. At the highest rainfall days all models generally under-predict precipitation, but the Gaussian models often fail to capture these extremes. While the DC-lognormal model has wider uncertainty intervals, it is able to produce a well-calibrated distribution at the extremes. Furthermore, these wide intervals indicate that the model becomes less confident with rare events at higher intensities, suggesting that there exists a bias-variance trade-off between the Gaussian and lognormal distributions. The ability of the DC-lognormal model to produce well-understood uncertainties at the extremes suggests that Bayesian deep NNs can model non-normal distributions well when motivated by domain knowledge.

What are the key implications? This work shows a successful and careful characterization of uncertainties in an SR model using BDL. The UQ method presented here is versatile and can be used for many other PIML applications. Drawing on domain expertise, this work also provides data-driven approaches to model extreme events.

(iii) MeshFreeFlowNet: a physics-constrained deep continuous space–time super-resolution framework

Most SR work has focused on enhancing only the spatial resolution of coarse data. Recent work has made an initial step toward addressing temporal coherence between consecutive snapshots by performing spatial SR in a temporally coherent manner [57]. Although far more challenging, enhancing the spatial and temporal resolution simultaneously is powerful as it can provide the fine-grained evolution of complex systems at temporal scales of relevance. Space–time SR goes beyond simple coherence by inserting entirely new snapshots of data, at the also-enhanced spatial resolution, in between given time steps of data.

Here, we review MeshfreeFlowNet, a novel SR framework to generate continuous (grid-free) spatio-temporal solutions of complex systems from low-resolution inputs [104].

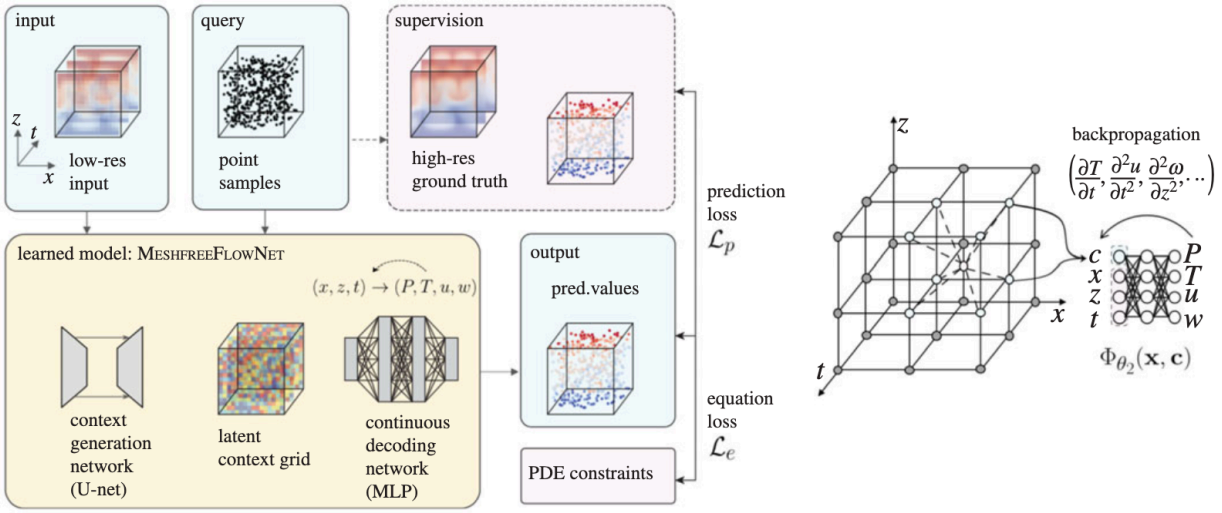

Figure 9. Left: Schematic for the training pipeline of MeshfreeFlowNet model for the continuous space-time super-resolution task. A input low-resolution grid is fed to the context generation network that creates a latent context grid. A random set of points in the corresponding space-time domain is sampled to query the latent context grid, and the physical output values at these query locations can be continuously decoded using a continuous decoding network, implemented as a Multilayer Perceptron. Right: Schematic for the continuous decoding module of MeshfreeFlowNet, a Multilayer Perceptron that inputs the spatio-temporal coordinates of a query point, along with a latent context vector, and is decoded into the required physical channels of interest. Since each query point falls into a cell bounded by eight neighbouring vertices, the query is performed eight times, each using a different latent context vector and a different relative spatio-temporal coordinates with respect to each vertex. The values are then interpolated using trilinear interpolation to get the final value at the query point. Figure reproduced from [104].

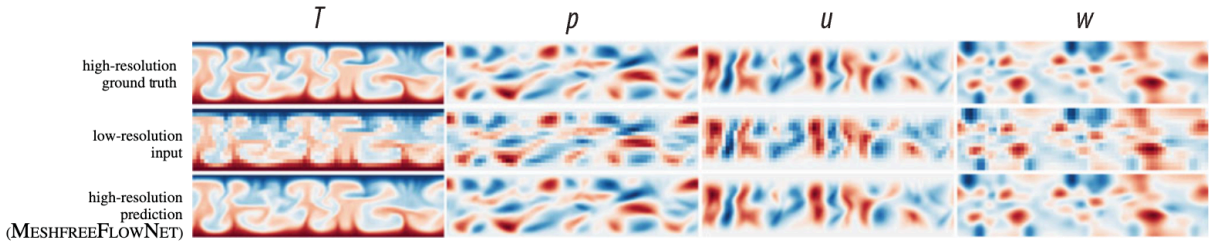

Figure 10. Sample tuples of low-resolution input data, the high-resolution super-resolved data by MeshfreeFlowNet, and the ground truth high-resolution data for the four physical parameters of the RBC system, i.e. T, p, u, w, respectively, as the temperature, pressure, and the x and z components of the velocity. A video that shows both spatial and temporal resolution enhancement is at https://tinyurl.com/y64papp9. Figure reproduced from [104]. (Online version in colour.)

MeshfreeFlowNet allows for: (i) the output to be sampled at all spatio-temporal resolutions, (ii) a set of PDE constraints to be imposed; and (iii) training on fixed-size inputs on arbitrarily sized spatio-temporal domains owing to its fully convolutional encoder. MeshfreeFlowNet learns the inherent statistical correlations between pairs of low-resolution and high-resolution solutions in a self-supervised manner to reconstruct high-resolution solutions from the low-resolution inputs. Although MeshfreeFlowNet can be queried at any spatio-temporal location and thus, in principle, produce outputs at any spatio-temporal resolution, the fidelity of the output is limited by the quality of the HR training data.

How is physics incorporated? MeshfreeFlowNet consists of two sub-networks: the context generation network and the continuous decoding network. The context generation network is a learned localized representation of the flow that encodes local correlations across space and time into a latent context grid, thus learning to preserve spatial and temporal coherence characteristic of the training data. A random set of points in the corresponding space-time domain is sampled to query the Latent Context Grid, and the physical output values at these query

locations can be continuously decoded using a continuous decoding network, implemented as a multilayer perceptron (MLP). Due to the differentiable nature of the MLP, any partial derivatives of the output physical quantities with respect to the input space–time coordinates can be effectively computed via backpropagation, enabling an easy way of enforcing PDE-based physical constraints as an *Equation Loss*. The whole framework is trained end-to-end with a weighted combination of two losses: (i) the L1 norm of the difference between the predicted physical outputs and the ground truth physical outputs, referred to as the *Prediction Loss*, and (ii) the L1 norm of the residuals of the governing PDEs, referred to as the *Equation Loss*. The residual of the PDE measures the imbalances in mass, momentum and energy conservation equations (Navier–Stokes equations) computed using the predicted output physical quantities. Gradients from the combined losses are backpropagated through the network for training. Figure 9 shows a schematic of the framework. NN architectures and mathematical details are in [104] (figure 10).

MeshfreeFlowNet's effectiveness is tested on the turbulent RBC (see [104] for details on the problem set-up, solvers and datasets). Figure 10 shows sample tuples of low-resolution input data, the high-resolution super-resolved data by MeshfreeFlowNet, and the ground truth high-resolution data for the four physical parameters of the RBC system, i.e. T, p, u, w, respectively, as the temperature, pressure, and the x and z components of the velocity. The super-resolved data is essentially indistinguishable from the true high-resolution data. A video that shows both spatial and temporal resolution enhancement is at https://tinyurl.com/y64papp9. A comprehensive set of turbulent flow metrics is used to rigorously test the physical validity and accuracy of the predictions: total kinetic energy (E_{tot}), Root-mean-squared velocity (u_{rms}), turbulence dissipation rate (ε), Taylor microscale (λ), Taylor-scale Reynolds number (Re_λ), Kolmogorov time (τ_η) and length scales (η), turbulent integral scale (L), and large eddy turnover time (T_L). A comparison between the performance of MeshfreeFlowNet framework against two baselines: a classic trilinear interpolation algorithm—Baseline (I)—and a DL-based 3D U-Net model—Baseline (II)—are shown in table 6. Normalized mean absolute error (NMAE) and R2-score of the flow-based evaluation metrics (R2-score is shown in brackets below NMAE) are evaluated for the predicted versus the ground truth high-resolution data. The R2-score is defined as 1 - ($SS_{\text{res}}/SS_{\text{tot}}$), where SS_{res} is the sum of squares of the differences between the predicted physical outputs and the ground truth physical outputs and SS_{res} is the total sum of squares. Because SS_{res} can be larger than SS_{tot}, the R2-score can be large and negative. γ is the weight of the *Equation loss*, hence $\gamma = 0$ is the case with only the *Prediction loss* and $\gamma = \gamma*$ is the optimal weight determined by hyper-parameter optimization. Baseline (I) fails to reconstruct the high-resolution data and resolve the fine-scale details, leading to large errors in the flow-based evaluation metrics. The DL Baseline (II) directly maps the low-resolution data to the high-resolution space, achieving better performance compared to Baseline (I). MeshfreeFlowNet outperforms the Baselines (I) and (II). In particular, it accurately recovers the fine-scale quantities of interest, Kolmogorov time (τ_η) and length scales (η), significantly better than any of the other methods.

Generalizability was tested by examining performance across the same set of nine metrics for Rayleigh numbers larger and smaller than the training dataset. Table 7 shows that the R2 scores for flow regimes that are up to two orders of magnitude above or below Rayleigh numbers it has been trained on decreases by less than 10%, suggesting that the model generalizes well. It also generalizes well to unseen physical initial conditions (not shown here).

Furthermore, a large-scale implementation of MeshfreeFlowNet shows that it efficiently scales across large clusters with hundreds of GPUs. Hence, it could be applied to large-scale realistic problems that require orders of magnitude more computational resources (see [104] for details on scaling).

What are the key implications? The MeshfreeFlowNet SR framework presented above has many powerful features: spatio-temporal coherence; PDE-constrained loss; improved performance on physically motivated metrics; the ability to super-resolve at arbitrary spatial and temporal locations (grid-free) on arbitrarily large domains; generalizability; and high scalability. Thus it is well-poised for applications in realistic three-dimensional turbulent flows in the atmosphere and ocean.

Table 6. Comparison between the performance of MeshfreeFlowNet framework for super-resolving the low-resolution data versus two baseline models. Normalized mean absolute error (NMAE) and R2-score (in brackets below NMAE) of the flow-based evaluation metrics evaluated for the predicted versus the ground truth high-resolution validation data. γ refers to the coefficient of the equation loss in the total loss function. Table reproduced from Jiang *et al*. [104].

	100 × NMAE (R2)									
model	E_{tot}	u_{rms}	ε	λ	Re_λ	τ_η	η	L	T_L	avg. R2
baseline (I)	69.6360	3470.132	76.338	78.164	75.729	77.410	122.55	137.376	109.398	−3541
	(−19.7894)	(−57.7717)	(−14)	(−11 096)	(−4934)	(−12 599)	(−3147)	(−0.5190)	(−3.0092)	
baseline (II)	6.489	8.769	6.144	3.903	2.489	5.584	6.019	2.902	5.076	0.9482
	(0.9557)	(0.8967)	(0.9593)	(0.9490)	(0.9711)	(0.9382)	(0.94019)	(0.9597)	(0.9644)	
MeshfreeFlowNet, $\gamma = 0$	0.667	0.768	0.666	0.545	0.444	0.753	0.752	0.837	0.548	0.9986
	(0.9991)	(0.9987)	(0.9991)	(0.9985)	(0.9989)	(0.9981)	(0.9984)	(0.9968)	(0.9994)	
MeshfreeFlowNet, $\gamma = \gamma^*$	0.621	0.603	0.617	0.431	0.429	0.461	0.483	0.857	0.515	0.9991
	(0.9993)	(0.9992)	(0.9993)	(0.9992)	(0.9989)	(0.9994)	(0.9994)	(0.9972)	(0.9992)	

Table 7. For a MeshfreeFlowNet model that has been trained on 10 datasets each having a different boundary condition (Rayleigh number) as $Ra \in [2, 90] \times 10^5$ with $Pr = 1$, the super-resolution performance evaluation is reported for: a Rayleigh number within the range of boundary conditions of the training sets (i.e. $Ra = 5 \times 10^6$), Rayleigh numbers slightly below and above the range of boundary conditions of the training sets (i.e. $Ra = 1 \times 10^5$ and $Ra = 1 \times 10^7$, respectively), and Rayleigh numbers far below and above the range of boundary conditions of the training sets (i.e. $Ra = 1 \times 10^4$ and $Ra = 1 \times 10^8$, respectively). Normalized mean absolute error (NMAE) and R2-score (in brackets below NMAE) of the flow-based evaluation metrics evaluated for the predicted versus the ground truth high-resolution validation data. Table reproduced from Jiang *et al*. [104].

	100 × NMAE (R2)									
Ra	E_{tot}	u_{rms}	ε	λ	Re_λ	τ_η	η	L	T_L	avg. R2
1×10^4	1.480	2.444	0.881	31.319	0.422	1.947	1.564	32.070	0.388	0.8328
	(0.9962)	(0.9872)	(0.9988)	(0.3788)	(0.9998)	(0.8005)	(0.9490)	(0.3881)	(0.997)	
1×10^5	1.299	1.226	1.247	0.339	0.325	1.022	1.040	0.946	0.735	0.9983
	(0.9968)	(0.9981)	(0.9972)	(0.9996)	(0.9996)	(0.9978)	(0.9981)	(0.9988)	(0.9988)	
5×10^6	1.242	1.703	1.270	1.070	0.931	1.661	1.669	0.794	0.446	0.9952
	(0.9961)	(0.9923)	(0.9955)	(0.99517)	(0.9967)	(0.9927)	(0.9927)	(0.9970)	(0.9987)	
1×10^7	1.336	2.659	1.352	1.742	1.504	2.205	2.275	4.218	0.831	0.9878
	(0.9965)	(0.9839)	(0.9965)	(0.9873)	(0.9926)	(0.9890)	(0.9885)	(0.9578)	(0.9985)	
1×10^8	2.115	3.11	2.084	5.873	7.483	4.607	4.281	7.381	1.077	0.9543
	(0.9931)	(0.9832)	(0.9933)	(0.9209)	(0.8742)	(0.9478)	(0.9594)	(0.9192)	(0.9978)	

(c) Spatio-temporal forecasting

Predicting the spatio-temporal evolution of weather and climate phenomena by learning their highly nonlinear dynamics from large-scale simulations or observational data is extremely challenging. Given the computational efficiency of ML techniques and the limitations of today's numerical weather prediction and climate models, in particular, with respect to resolution and complexity, ML offers attractive alternatives for weather and climate forecasting [120]. ML models trained on samples from very high resolution simulations or observations can be evolved forward in time as has been shown in [98,120–123]. However, it remains to be seen if purely data-driven models will be capable of forecasting large ensembles at high resolutions. In this section, we survey PIML approaches for predicting the spatio-temporal evolution of turbulent flows in the atmosphere and ocean.

(i) TFNet: a hybrid physics-ML spatio-temporal forecasting model

Simulating the fluid dynamics of the atmosphere and oceans using first principles requires significant computational resources and domain expertise. Hybrid approaches that combine physics and data-driven methods show great promise for this grand challenge [62,63,70,74].

A widely used computational fluid dynamics technique solves the Reynolds-averaged Navier–Stokes (RANS) equations with a model for the closure term such as an eddy viscosity model. As described in §2(viii), LES resolves large-scale motions and models the effect of SGS turbulence. The hybrid RANS-LES coupling approach combines the computational efficiency of RANS with the more accurate resolving power of LES to provide a technique that is less expensive and more tractable than pure LES [124]. Here, we review TurbulentFlowNet (TFNet), proposed by Wang *et al.* [105], which applies scale separation and builds upon the structure of existing turbulence models.

How is physics incorporated? TFNet uses the physics-based structure of the RANS-LES coupling approach and replaces *a priori* spectral filters with trainable convolutional layers. The turbulent flow is decomposed into three components: mean flow, resolved fluctuations, and unresolved (subgrid) fluctuations, each of which is approximated by a specialized U-Net to preserve the multi-scale properties of the flow. The motivation for this design is to explicitly guide the ML model to learn the nonlinear dynamics of large-scale and SGS motions as relevant to the task of spatio-temporal prediction. Furthermore, a custom-designed loss that penalizes the absolute divergence is used as a regularizer. Figure 11 shows the overall architecture of TFNet. More details on the architecture are in [105].

TFNet is tested on RBC, as in §3ai(i). The unconstrained and constrained TFNet (Con TFNet in figure 12) are compared against four purely data-driven spatio-temporal ML models: ResNet, ConvLSTM, U-Net, and GAN; and against two PIML models: SST [59] and DHPM [15]. Besides RMSE, physically relevant metrics (turbulence kinetic energy, divergence and energy spectrum) are used to evaluate the performance of these models, as shown in figure 12. One large eddy turnover time is approximately 10 prediction steps.

TFNet is capable of generating both accurate and physically meaningful predictions that preserve critical quantities of relevance for forecasting the velocity field up to 60 steps ahead given the history. It consistently outperforms all baselines. The divergence constraint in Con TFNet further improves its performance. TFNet also generalizes well to a new Rayleigh number outside the regime it was trained on (not shown here). Furthermore, TFNet has a significantly smaller number of parameters than most baselines, and hence is a compact and efficient PIML model.

What are the key implications? This work shows an efficient and effective way to combine physics-based turbulence models with DL that can more accurately forecast complex spatio-temporal dynamics while capturing key physical properties such as the divergence-free condition. This further motivates the integration of physics-based models with ML.

A crucial requirement, however, is domain-specific expertise to identify suitable physical model structures that can be built upon, components of these models that can be augmented

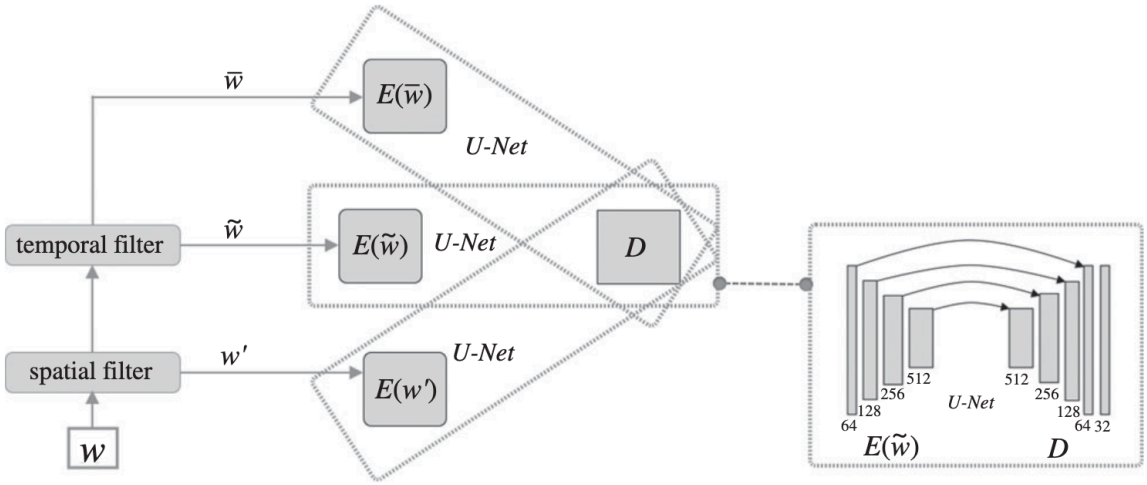

Figure 11. Turbulent flow net (TFNet) consists of three identical encoders that learn the transformations of the three components of the flow at different scales, and one shared decoder that learns the interactions among these three components to predict the velocity field at the next instant. Each encoder–decoder pair can be viewed as a U-Net. Figure reproduced from [105]. (Online version in colour.)

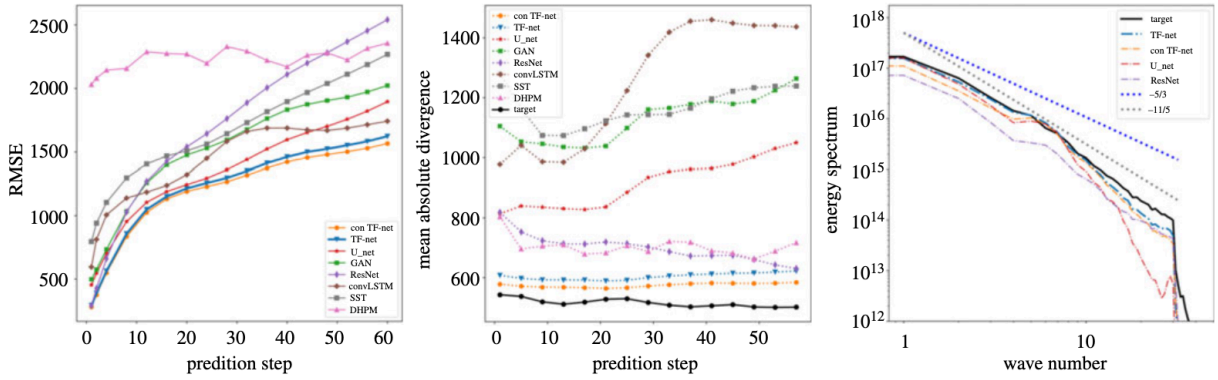

Figure 12. Comparison between performance of models: RMSE, mean absolute divergence and energy spectra. One large eddy turnover time is approximately 10 prediction steps. Figure reproduced from [105]. (Online version in colour.)

or replaced by ML, and training that is consistent with the modelling assumptions of the physics-based model.

(ii) Symmetric and equivariant deep dynamics models

As discussed in §2biii, designing a model that is equivariant to transformations of its input guarantees that the model generalizes across these transformations, making it more robust to distributional shifts and out-of-sample scenarios. These models are also more compact and data efficient because of the embedded symmetries and equivariances. Here, we review recent work by Wang *et al.* [30] in building equivariant deep dynamics models for predicting the spatio-temporal evolution of RBC and ocean currents.

How is physics incorporated? Wang *et al.* [30] consider symmetries of translation, rotation, uniform motion, and scaling. They develop the mathematical framework and tailor practical methods for incorporating each symmetry into deep NNs. Their key to building equivariant networks is that the composition of equivariant functions is equivariant. They show that if the maps between layers of an NN are equivariant, then the whole network will be equivariant. Details on how these are implemented in ML models are in [30].

The ML models used are ResNet and U-Net, and their equivariant counterparts. Spatio-temporal prediction is done autoregressively. Standard RMSE and an RMSE computed on the PSD of energy spectra are used to measure performance. The models are tested on RBC and

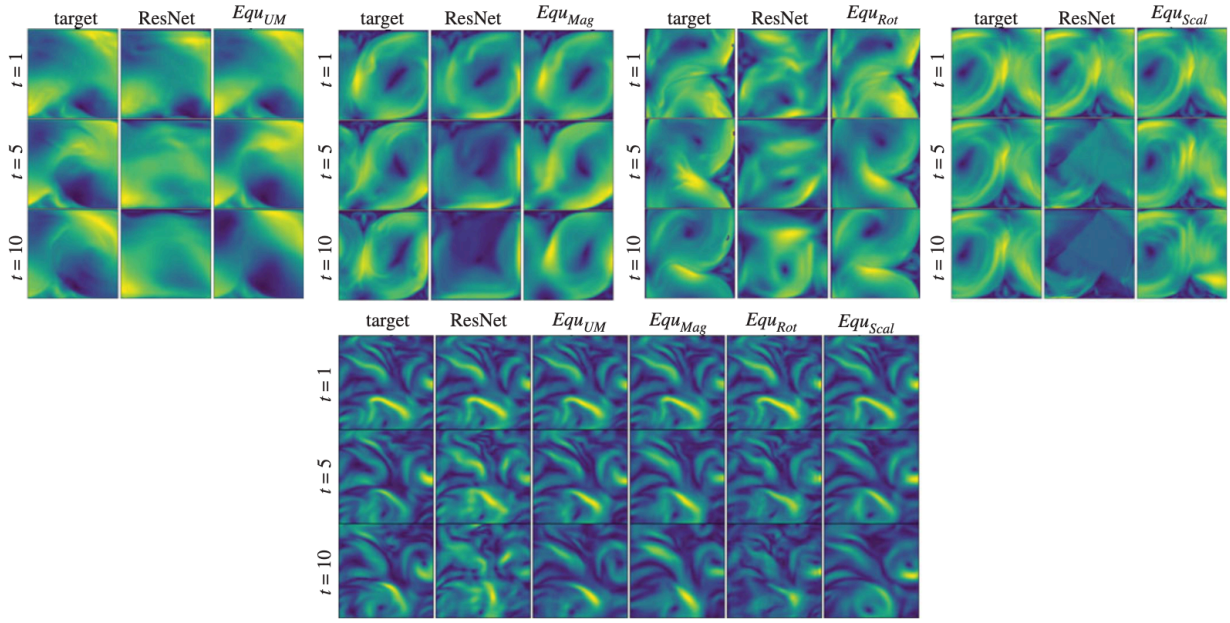

Figure 13. Comparison between performance of equivariant (Equ) and non-equivariant ResNet models for RBC velocity fields. From left to right are equivariant models under uniform motion, magnitude, rotation, and scale equivariance transformations. Tests are on future times, $t = 1, 5$ and 10. Bottom: Comparison between performance of equivariant (*Equ*) and non-equivariant ResNet models for ocean currents. *Equ* columns are equivariant models under uniform motion, magnitude, rotation, and scale equivariance transformations. No single equivariant model captures the target accurately; however, all equivariant models perform better than the non-equivariant baseline. Figure reproduced from [30]. (Online version in colour.)

reanalysis ocean current velocity data generated by the NEMO ocean engine (ORAS5) at three different locations in the Atlantic, Pacific, and Indian oceans. For RBC, the test sets have random transformations from the relevant symmetry groups applied to each sample. This mimics real-world data in which each sample has an unknown reference frame. For ocean data, tests are also performed on different time ranges and different domains from the training set, representing distributional shifts. In all cases, the equivariant models' predictions are more accurate than the baselines' (see [30] for quantitative comparisons). Figure 13 shows that the equivariant models perform significantly better than their non-equivariant counterparts, preserving spatial and temporal coherence as well as fine scale structures. In [30], the authors show quantitatively that the equivariant models are robust under data transformations and distributional shifts.

What are the key implications? This work incorporates various symmetries into NNs to develop novel equivariant models for forecasting atmospheric and oceanic flows that generalize well. Combining different symmetries into a single equivariant model and extending to three-dimensional flows can have important implications for developing compact and tractable ML models for realistic geophysical flows.

(iii) Deep spatial transformers for autoregressive forecasting

Here, we review work by Chattopadhyay *et al.* [31] that shows that the equivariance preserving properties of modern spatial transformers incorporated within a convolutional encoder–decoder module can predict the spatio-temporal evolution of geophysical turbulence successfully [125]. Furthermore, preserving the equivariance and using custom-designed losses enables stable predictions for multiple years, providing promise for the development of a stable and physical data-driven PIML model for weather and climate forecasting.

Chattopadhyay *et al.* [31] consider a fully turbulent flow represented by the two-layered quasi-geostrophic equations (QG) with a baroclinically unstable jet. The complexity of this QG system based on the instantaneous attractor dimension of the upper layer's stream function (Ψ_1) is about 20.9 and comparable to the instantaneous attractor dimension of geopotential height at 500 hPa

(Z500) in the observed atmosphere [98]. A PIML model is developed that can accurately predict the short-term dynamics of the upper layer's stream function without any information about the lower layer's stream function, even during training, thus having important implications for data-driven forecasting from partial observations.

How is physics incorporated? Physics is incorporated into the model by implementing a spatial transformer module embedded within the encoding block of an encoder–decoder style DL model. The transformer module preserves, within feature maps, equivariance of translation, scale, rotation, and generic warping of local features of turbulent vortices. Incorporating these equivariance properties are shown to increase accuracy in the short term, and provide a physical meaningful stable climate in long-term predictions. Additionally, custom loss functions are used that address instabilities and long-term drift. Furthermore, spatial transformers are memory efficient and easier to implement as a fully differentiable layer inside any DL architecture, thus providing attractive computational advantages over other DL architectures. Details on how these are implemented, the training datasets, and the training procedure are in [31].

Performance is measured using the correlation coefficient between predictions and ground truth starting from a random initial condition on an unseen dataset. Figure 14*a* shows that the model with custom losses outperforms persistence and the baseline encoder–decoder without a spatial transformer even at the sub-seasonal scale (10 days). At longer time scales (20–90 days), the model remains physical and the predicted jet-stream does not drift and maintains a stable physical climate at all times. By contrast, models without the custom loss and transformer module do not simulate a physical climate and show a drifting jet where low pressure anomalies dominate. Persistence performs well at sub-seasonal to seasonal timescales because in a chaotic system small errors quickly accumulate and the predicted trajectory diverges away from the true trajectory of the dynamical system. This corroborates findings by Weyn *et al.* [121].

Figure 14*b* shows long-term averaged quantities of the dynamics. The zonally averaged Ψ_1 has been obtained by, first, averaging over 1000 days of predictions to obtain the time average and then zonally averaged. A similar analysis has been done for meridional averaging. The long-term mean of the meridionally averaged Ψ_1 resembles the true long-term mean of the system more closely than persistence while in the case of zonally averaged Ψ_1, both persistence and data-driven models resemble the truth quite well.

What are the key implications? By incorporating equivariance-preserving properties of a spatial transformer and custom loss functions, this work develops a novel PIML approach to reliable forecasting of a realistic geophysical flow. The ability to predict a stable, physically meaningful climate in the long term and achieve reliable forecasts from partial observations is promising. Demonstrating these capabilities on realistic datasets such as a reanalyses will have important implications for forecasting of geophysical phenomena using PIML.

How can uncertainty be quantified in spatio-temporal forecasting? Although UQ has as yet not been employed in the PIML models for spatio-temporal forecasting discussed above, ensemble ML techniques provide opportunities for characterizing and estimating uncertainties [126]. For purely data-driven ML-based forecasting, Bihlo [127] shows how GANs can be used with Monte Carlo dropout to develop an ensemble weather prediction model and Fanfarillo *et al.* [128] develop a deep generative model for probabilistic forecasts. Adapting these and other techniques [84] for PIML models could provide new opportunities for systematic UQ in forecasting.

4. Synthesis and outlook

In this article, we review progress in PIML towards addressing some critical challenges in weather and climate modelling, namely: (i) building better emulators for complex multi-scale physical processes; (ii) downscaling (super-resolving) coarse data to produce high-fidelity high-resolution data; and (iii) forecasting the spatio-temporal dynamics of the atmosphere and ocean.

Using the 10 approaches described in §2b, the case studies characterized in table 1 illustrate the field's significant progress in advancing PIML.

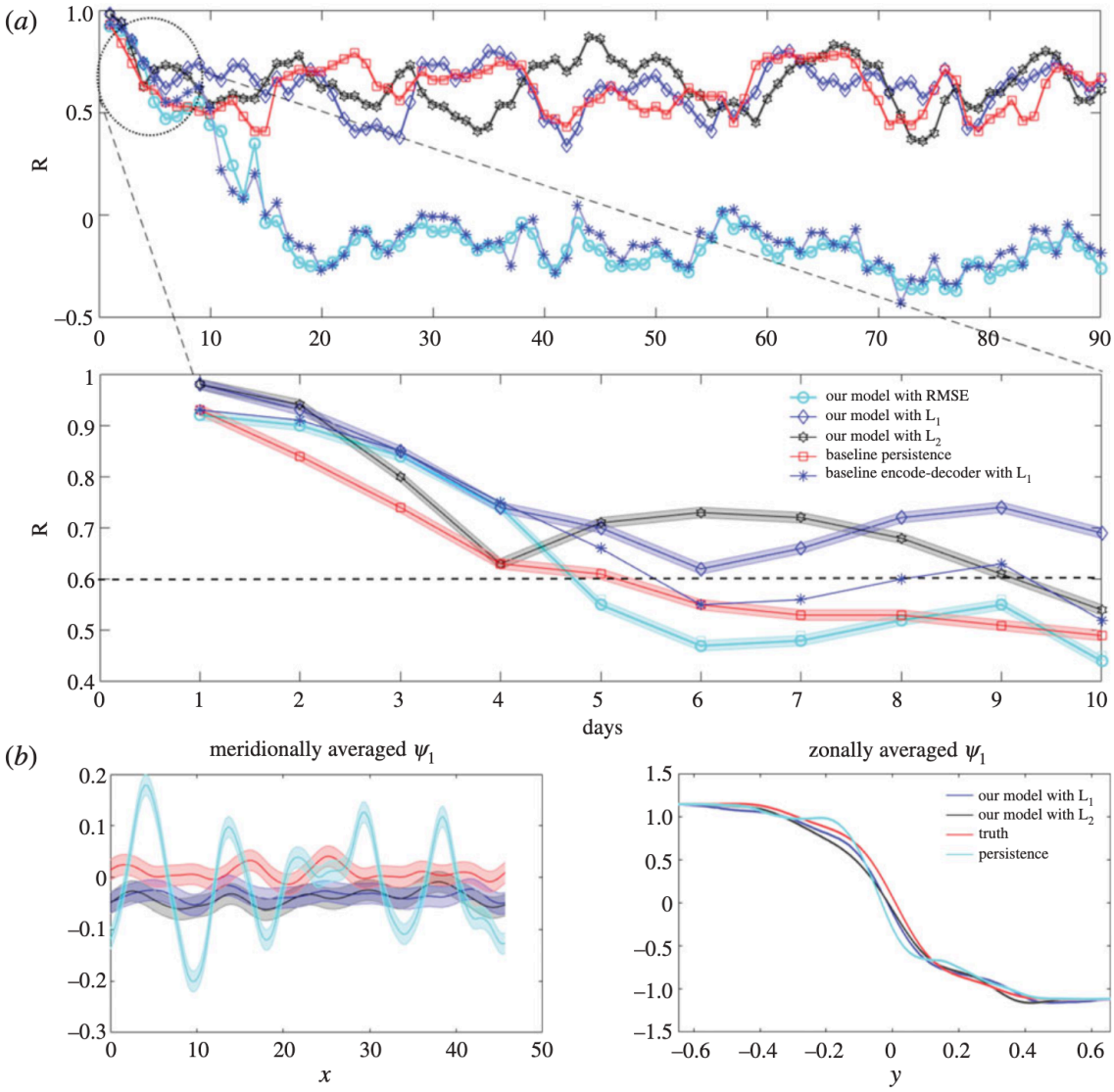

Figure 14. (*a*) R is the correlation coefficient for short-term and sub-seasonal to seasonal prediction with three different network architectures. L1 and L2 are custom losses. All models outperform persistence for short-term prediction up to 4 days. Beyond 4 days the architectures with L1 and L2 losses outperform persistence for up to 10 days (sub-seasonal scale) and remain comparable with persistence for up to 90 days, with a stable physical climate. All of the analyses were repeated for 10 different initial conditions chosen from the test set and separated by at least 1000 days. The mean (symbols) and standard deviation (shading) are reported in the figure. The top plot shows R up to 90 days; the bottom plot zooms into the first 10 days. (*b*) Long-term averaged dynamical quantities are predicted by models and compared against persistence and truth. Figure reproduced from [31]. (Online version in colour.)

(a) Key achievements

In aggregate, the 10 case studies discussed in this article demonstrate that PIML can achieve increased physical consistency, higher accuracy, faster training, better convergence, data efficiency, improved generalization, greater interpretability, and increased scalability to more complex physical systems and larger computational platforms. The cumulative accomplishments reveal the leading edge of PIML's contribution to weather and climate modelling.

(b) Lessons learned

The case studies presented here, complemented by other studies in the larger scientific community, offer lessons that can help guide current and future research. We believe that the following set of guidelines will enable the development of robust and reliable PIML models:

— Enforce standards for testing accuracy and physical consistency applicable to state-of-the-art physics-based models of the relevant domain.
— Characterize and quantify all sources of uncertainty during model development and in predictions.
— Set realistic development objectives by identifying errors that cannot be reduced and discrepancies that cannot be addressed, including limitations of model structure and training data.
— Train models with data characteristics, such as noise, sparsity, and incompleteness, that are representative of the downstream application.
— Promote model-consistent training for PIML models that will be embedded into weather or climate models.
— Quantify generalizability in terms of how performance degrades with degree of extrapolation to unseen initial conditions, boundary conditions, and scenarios.
— Derive or estimate stability and convergence properties.
— Evaluate model fidelity on rare events, extremes, and tails of distributions.
— Build interpretable models, explain predictions, perform ablation studies, detect biases, and identify limitations.
— Encourage reproducible research.

(c) Where do we go from here?

Although PIML has progressed significantly in the past few years, at least four types of grand challenges are yet to be addressed: scientific, diagnostic, computational, and resource. We pose these challenges in the form of the following questions:

Scientifically, how can we incorporate the lessons learned into future model development?

Diagnostically, how do we develop systematic tests and standardize evaluation for these models across benchmark datasets and problems? Model inter-comparison projects are routine for weather and climate scientists; however, systematic diagnostics are largely lacking for ML applications in weather and climate. A first step towards this in medium-range weather forecasting is WeatherBench [129].

Computationally, how do we scale the training, testing and deployment of complex PIML models on large datasets efficiently, so that they perform well in a rapidly changing computational landscape [130]?

Resource-wise, how can we effectively collaborate across many diverse communities: physicists, mathematicians, computer scientists, statisticians, and domain scientists from many different domains? This collaboration is fundamental for the rapid growth and success of PIML.

Through addressing these challenges, we anticipate the development of truly robust and reliable PIML models, ultimately making them invaluable for scientific discovery and indispensable to weather and climate modelling.

Data accessibility. Data, code and supporting materials are publicly available via the following links: https://github.com/jinlong83/statistical-constrained-GANS; https://github.com/maxjiang93/space_time_pde; https://github.com/Rose-STL-Lab/Turbulent-Flow-Net; https://github.com/Rui1521/Equivariant-Neural-Nets; https://github.com/ashesh6810/Deep-Spatial-Transformers; https://resources.marine.copernicus.eu/?option=com_csw&view=details&product_id=GLOBAL_ANALYSIS_FORECAST_PHY_001_024; https://portal.edirepository.org/nis/mapbrowse?packageid=edi.200.6; https://doi.org/10.6073/pasta/8f19c5d19d816857e55077ba20570265; https://prism.oregonstate.edu/; https://github.com/arkadaw9/PGA_LSTM; https://lter.limnology.wisc.edu/data; https://gitlab.com/mspritch/spcam3.0-neural-net; https://doi.org/10.5281/zenodo.2559313.

Authors' contributions. K.K. conceived the idea and designed the structure of the manuscript, wrote the manuscript, and responded to reviewer comments. K.K., M.M., and A.A. led the majority of the research reviewed as case studies in this article. The rest of the authors contributed to the research reviewed as case studies or provided feedback on sections of the manuscript. K.K. dedicates this work to A.A., a colleague and dear friend, who unfortunately was killed in a hit-and-run road accident while he was biking, during the course of preparation of this manuscript.

Competing interests. We declare we have no competing interests.
Funding. No funding has been received for this article.

References

1. The Royal Society 2019 The AI revolution in scientific research. London, UK: The Royal Society. (http://www.The-Royal-Society-The-AI-revolution-in-scientific-research.com)
2. Hey T, Trefethen A. 2020 The fourth paradigm 10 years on. *Informatik Spektrum* **42**, 441–447. (doi:10.1007/s00287-019-01215-9)
3. LeCun Y, Bengio Y, Hinton G. 2015 Deep learning. *Nature* **521**, 436–444. (doi:10.1038/nature14539)
4. Moore GE. 1965 Cramming more components onto integrated circuits. *Electronics* **38**, 114–117.
5. Shalf J. 2020 The future of computing beyond Moore's law. *Phil. Trans. R. Soc. A* **378**, 20190061. (doi:10.1098/rsta.2019.0061)
6. Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N, Prabhat . 2019 Deep learning and process understanding for data-driven Earth system science. *Nature* **566**, 195–204. (doi:10.1038/s41586-019-0912-1)
7. Kasim MF *et al.* 2020 Building high accuracy emulators for scientific simulations with deep neural architecture search. (http://arxiv.org/abs/2001.08055).
8. Buchanan M. 2020 Living up to the promise. *Nat. Phys.* **16**, 706–706. (doi:10.1038/s41567-020-0962-1)
9. Hutson M. 2020 From models of galaxies to atoms, simple AI shortcuts speed up simulations by billions of times. *Science* **367**, 728–728. (doi:10.1126/science.367.6479.728)
10. Hestness J, Narang S, Ardalani N, Diamos G, Jun H, Kianinejad H, Patwary MMA, Yang Y, Zhou Y. 2017 Deep learning scaling is predictable, empirically. (http://arxiv.org/abs/1712.00409).
11. Karpatne A, Atluri G, Faghmous JH, Steinbach M, Banerjee A, Ganguly A, Shekhar S, Samatova N, Kumar V. 2017 Theory-guided data science: a new paradigm for scientific discovery from data. *IEEE Trans. Knowl. Data Eng.* **29**, 2318–2331. (doi:10.1109/TKDE.2017.2720168)
12. Willard J, Jia X, Xu S, Steinbach M, Kumar V. 2020 Integrating physics-based modeling with machine learning: a survey. (http://arxiv.org/abs/2003.04919).
13. Karpatne A, Watkins W, Read J, Kumar V. 2018 Physics-guided neural networks (PGNN): an application in lake temperature modeling. In *Proc. of the 2018 SIAM Int. Conf. on Data Mining*.
14. Beucler T, Rasp S, Pritchard M, Gentine P. 2019 Achieving conservation of energy in neural network emulators for climate modeling. (http://arxiv.org/abs/1906.06622).
15. Raissi M, Perdikaris P, Karniadakis GE. 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707. (doi:10.1016/j.jcp.2018.10.045)
16. Zhu Y, Zabaras N, Koutsourelakis PS, Perdikaris P. 2019 Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *J. Comput. Phys.* **394**, 56–81. (doi:10.1016/j.jcp.2019.05.024)
17. Beucler T, Pritchard M, Rasp S, Ott J, Baldi P, Gentine P. 2019 Enforcing analytic constraints in neural-networks emulating physical systems. (http://arxiv.org/abs/1909.00912).
18. Wu JL, Kashinath K, Albert A, Chirila D, Xiao H. 2020 Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. *J. Comput. Phys.* **406**, 109209. (doi:10.1016/j.jcp.2019.109209)
19. Márquez-Neila P, Salzmann M, Fua P. 2017 Imposing hard constraints on deep networks: promises and limitations. (http://arxiv.org/abs/1706.02025).
20. Mohan AT, Lubbers N, Livescu D, Chertkov M. 2020 Embedding hard physical constraints in neural network coarse-graining of 3D turbulence. (http://arxiv.org/abs/2002.00021).
21. Jiang CM, Kashinath K. 2020 Enforcing physical constraints in CNNs through differentiable PDE layer. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*.
22. Daw A, Thomas RQ, Carey CC, Read JS, Appling AS, Karpatne A. 2020 Physics-guided architecture (PGA) of neural networks for quantifying uncertainty in lake temperature modeling. In *Proc. of the 2020 SIAM Int. Conf. on Data Mining*, pp. 532–540.

23. Cohen TS, Weiler M, Kicanaoglu B, Welling M. 2019 Gauge equivariant convolutional networks and the icosahedral CNN. In *Proc. of the 36th Int. Conf. on Machine Learning*.
24. Maron H, Fetaya E, Segol N, Lipman Y. 2019 On the universality of invariant networks. In *Proc. of the 36th Int. Conf. on Machine Learning*.
25. Maron H, Ben-Hamu H, Shamir N, Lipman Y. 2019 Invariant and equivariant graph networks. In *Int. Conf. on Learning Representations*.
26. Thomas N, Smidt T, Kearnes S, Yang L, Li L, Kohlhoff K, Riley P. 2018 Tensor field networks: rotation-and translation-equivariant neural networks for 3D point clouds. (http://arxiv.org/abs/1802.08219).
27. Ling J, Kurzawski A, Templeton J. 2016 Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166. (doi:10.1017/jfm.2016.615)
28. Cohen TS, Geiger M, K¨ohler J, Welling M. 2018 Spherical CNNs. In *Int. Conf. on Learning Representations*.
29. Jiang C, Huang J, Kashinath K, Marcus P, Nießner M. 2019 Spherical CNNs on unstructured grids. In *Int. Conf. on Learning Representations*.
30. Wang R, Walters R, Yu R. 2020 Incorporating symmetry into deep dynamics models for improved generalization. (http://arxiv.org/abs/2002.03061).
31. Chattopadhyay A, Mustafa M, Hassanzadeh P, Kashinath K. 2020 Deep spatial transformers for autoregressive data-driven forecasting of geophysical turbulence. *In Proc of the 10th Workshop on Climate Informatics, Oxford, UK, 2020*.
32. Beucler T, Pritchard M, Gentine P, Rasp S. 2020 Towards physically-consistent, data-driven models of convection. (http://arxiv.org/abs/2002.08525).
33. Palmer TN. 2019 Living up to the promise. *Nat. Rev. Phys.* **1**, 463–471. (doi:10.1038/s42254-019-0062-2)
34. Ghahramani Z. 2015 Probabilistic machine learning and artificial intelligence. *Nature* **521**, 452–459. (doi:10.1038/nature14541)
35. Krasnopolsky VM, Fox-Rabinovitz MS, Belochitski AA. 2013 Using ensemble of neural networks to learn stochastic convection parameterizations for climate and numerical weather prediction models from data simulated by a cloud resolving model. *Ad. Artif. Neural Syst.* **2013**, 485–913.
36. Gagne DJ, Christensen HM, Subramanian AC, Monahan AH. 2020 Machine learning for stochastic parameterization: generative adversarial networks in the lorenz '96 model. *J. Adv. Model. Earth Syst.* **12**, e2019MS001896. (doi:10.1029/2019MS001896)
37. Groenke B, Madaus L, Monteleoni C. 2020 ClimAlign: unsupervised statistical downscaling of climate variables via normalizing flows. *Proc. of the 10th Workshop on Climate Informatics*, Oxford, UK, 2020.
38. Tadmor E. 2012 A review of numerical methods for nonlinear partial differential equations. *Bull. Am. Math. Soc.* **49**, 507–554. (doi:10.1090/S0273-0979-2012-01379-4)
39. Miller J, Hardt M. 2019 Stable recurrent models. In *Int. Conf. on Learning Representations*.
40. Erichson NB, Muehlebach M, Mahoney MW. 2019 Physics-informed Autoencoders for Lyapunov-stable Fluid Flow Prediction. (http://arxiv.org/abs/1905.10866).
41. Lusch B, Kutz JN, Brunton SL. 2018 Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.* **9**, 4950. (doi:10.1038/s41467-018-07210-0)
42. Mamakoukas G, Abraham I, Murphey TD. 2020 Learning data-driven stable Koopman operators. (http://arxiv.org/abs/2005.04291).
43. Brenowitz ND, Beucler T, Pritchard M, Bretherton CS. 2020 Interpreting and stabilizing machine-learning parametrizations of convection. (http://arxiv.org/abs/2003.06549).
44. Ott J, Pritchard M, Best N, Linstead E, Curcic M, Baldi P. 2020 A Fortran-Keras deep learning bridge for scientific computing. *arXiv:2004.10652* (https://arxiv.org/abs/2004.10652)
45. Rasp S. 2020 Coupled online learning as a way to tackle instabilities and biases in neural network parameterizations: general algorithms and Lorenz 96 case study (v1.0). *Geosci. Model Dev.* **13**, 2185–2020. (doi:10.5194/gmd-13-2185-2020)
46. Yuval J, O'Gorman P. 2020 Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nat. Commun.* **11**, 3295. (doi:10.1038/s41467-020-17142-3)

47. Yuval J, O'Gorman PA, Hill CN. 2020 Use of neural networks for stable, accurate and physically consistent parameterization of subgrid atmospheric processes with good performance at reduced precision. (http://arxiv.org/abs/2010.00947).
48. Slingo J, Bates K, Nikiforakis N, Piggott M, Roberts M, Shaffrey L, Stevens I, Vidale PL, Weller H. 2008 Developing the next-generation climate system models: challenges and achievements. *Phil. Trans. R. Soc. A* **367**, 815–831. (doi:10.1098/rsta.2008.0207)
49. Mohan A, Livescu D, Chertkov M. 2020 Wavelet-powered neural networks for turbulence. In *ICLR 2020 Workshop on Climate Change AI*.
50. Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, Anandkumar A. 2020 Fourier neural operator for parametric partial differential equations. (http://arxiv.org/abs/2010.08895).
51. Tancik M, Srinivasan PP, Mildenhall B, Fridovich-Keil S, Raghavan N, Singhal U, Ramamoorthi R, Barron JT, Ng R. 2020 Fourier features let networks learn high-frequency functions in low-dimensional domains. *NeurIPS* **33**.
52. Pratt H, Williams B, Coenen F, Zheng Y. 2017 FCNN: fourier convolutional neural networks. *ECML PKDD 17*, pp. 786–798.
53. Guan B, Zhang J, Sethares WA, Kijowski R, Liu F. 2019 SpecNet: spectral domain convolutional neural network. (http://arxiv.org/abs/1905.10915).
54. Williams PD *et al.* 2017 A census of atmospheric variability from seconds to decades. *Geophys. Res. Lett.* **44**, 11 201–11 211.
55. Haller G. 2015 Lagrangian coherent structures. *Annu. Rev. Fluid Mech.* **47**, 137–162. (doi:10.1146/annurev-fluid-010313-141322)
56. Rupe A, Kumar N, Epifanov V, Kashinath K, Pavlyk O, Schlimbach F, Patwary M, Maidanov S, Lee V, Prabhat M, Crutchfield JP. 2019 DisCo: Physics-based unsupervised discovery of coherent structures in spatiotemporal systems. In *Proceedings of the 2019 IEEE/ACM Workshop on Machine Learning in High Performance Computing Environments* (MLHPC), Denver, CO, pp. 75–87.
57. Xie Y, Franz E, Chu M, Thuerey N. 2018 tempoGAN: a temporally coherent, volumetric GAN for super-resolution fluid flow. *ACM Trans. Graph.* **37**, 1–15.
58. Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, Anandkumar A. 2020 Neural operator: graph kernel network for partial differential equations. (http://arxiv.org/abs/2003.03485).
59. de Bezenac E, Pajot A, Gallinari P. 2018 Deep learning for physical processes: incorporating prior scientific knowledge. In *Int. Conf. on Learning Representations*.
60. Xu K, Wen L, Li G, Bo L, Huang Q. 2019 Spatiotemporal CNN for video object segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1379–1388.
61. Smagorinsky J. 1963 General circulation experiments with the primitive equations: I. The basic experiment. *Mon. Weather Rev.* **91**, 99–164. (doi:10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2)
62. Cheng Y *et al.* 2019 Deep learning for subgrid-scale turbulence modeling in large-eddy simulations of the atmospheric boundary layer. (http://arxiv.org/abs/1910.12125).
63. Pal A. 2020 Deep learning emulation of subgrid–scale processes in turbulent shear flows. *Geophys. Res. Lett.* **47**, e2020GL087005. (doi:10.1029/2020GL087005)
64. Bar-Sinai Y, Hoyer S, Hickey J, Brenner MP. 2019 Learning data-driven discretizations for partial differential equations. *Proc. Natl Acad. Sci. USA* **116**, 15 344–15 349. (doi:10.1073/pnas.1814058116)
65. Pathak J, Wikner A, Fussell R, Chandra S, Hunt BR, Girvan M, Ott E. 2018 Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model. *Chaos* **28**, 041101. (doi:10.1063/1.5028373)
66. Watson PAG. 2019 Applying machine learning to improve simulations of a chaotic dynamical system using empirical error correction. *J. Adv. Model. Earth Syst.* **11**, 1402–1417. (doi:10.1029/2018MS001597)
67. Bonavita M, Laloyaux P. 2020 Machine learning for model error inference and correction. *J. Adv. Model. Earth Syst.* **12**, e2020MS002232. (doi:10.1029/2020MS002232)
68. Farchi A, Laloyaux P, Bonavita M, Bocquet M. 2020 Using machine learning to correct model error in data assimilation and forecast applications. *arXiv:2010.12605*. (https://arxiv.org/abs/2010.12605)

69. Mauritsen T *et al.* 2012 Tuning the climate of a global model. *J. Adv. Model. Earth Syst.* **4**, M00A01. (doi:10.1029/2012MS000154)
70. Duraisamy K, Iaccarino G, Xiao H. 2019 Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51**, 357–377. (doi:10.1146/annurev-fluid-010518-040547)
71. Ollinaho P, Bechtold P, Leutbecher M, Laine M, Solonen A, Haario H, Järvinen H. 2019 Parameter variations in prediction skill optimization at ecmwf. *Nonlin. Processes Geophys.* **71**, 1001–1010.
72. Tett SFB, Yamazaki K, Mineter MJ, Cartis C, Eizenberg N. 2017 Calibrating climate models using inverse methods: case studies with hadam3, hadam3p and hadcm3. *Geosci. Model Dev.* **10**, 3567–3589.
73. Bellprat O, Kotlarski S, Lüthi D, Elía RD, Frigon A, Laprise R, Schär C. 2016 Objective calibration of regional climate models: Application over europe and north america. *Journal of Climate* **29**, 819–838.
74. Duraisamy K. 2020 Machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence. (http://arxiv.org/abs/2009.10675).
75. Samek W, Montavon G, Vedaldi A, Hansen LK, Müller KR. 2019 *Explainable AI: interpreting, explaining and visualizing deep learning*. New York, NY: Springer Nature.
76. Molnar C. 2019 *Interpretable machine learning*. https://christophm.github.io/interpretable-ml-book/.
77. Lapuschkin S, Wäldchen S, Binder A, Montavon G, Samek W, Müller KR. 2019 Unmasking Clever Hans predictors and assessing what machines really learn. *Nat. Commun.* **10**, 1096. (doi:10.1038/s41467-019-08987-4)
78. Rudin C. 2019 Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **1**, 206–215. (doi:10.1038/s42256-019-0048-x)
79. McGovern A, Lagerquist R, Gagne DJ, Jergensen GE, Elmore KL, Homeyer CR, Smith T. 2019 Making the black box more transparent: understanding the physical implications of machine learning. *Bull. Am. Meteorol. Soc.* **100**, 2175–2199. (doi:10.1175/BAMS-D-18-0195.1)
80. Ebert-Uphoff I, Hilburn K. 2020 Evaluation, tuning and interpretation of neural networks for working with images in meteorological applications. *Bull. Am. Meteorol. Soc.* **101**, E2149–E2170.
81. Gagne DJ, Haupt SE, Nychka DW, Thompson G. 2019 Interpretable deep learning for spatial analysis of severe hailstorms. *Mon. Weather Rev.* **147**, 2827–2845. (doi:10.1175/MWR-D-18-0316.1)
82. Toms BA, Barnes EA, Ebert-Uphoff I. 2020 Physically interpretable neural networks for the geosciences: applications to earth system variability. *J. Adv. Model. Earth Syst.* **12**, e2019MS002002.
83. Toms BA, Kashinath K. 2020 Testing the reliability of interpretable neural networks in geoscience using the Madden-Julian oscillation. *Geosci. Model Dev.* 1–22. (https://gmd.copernicus.org/preprints/gmd-2020-152/)
84. Caldeira J, Nord B. 2020 Deeply uncertain: comparing methods of uncertainty quantification in deep learning algorithms. In *ICLR 2020 Workshop on Fundamental Science in the era of AI*.
85. Kendall A, Gal Y. 2017 What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, vol. 30, pp. 5574–5584.
86. Gal Y, Ghahramani Z. 2016 Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In *Proc. of the 33rd Int. Conf. on Machine Learning (ICML-16)*.
87. Lakshminarayanan B, Pritzel A, Blundell C. 2017 Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, vol. 30, pp. 6402–6413.
88. Yang Y, Perdikaris P. 2019 Adversarial uncertainty quantification in physics-informed neural networks. *J. Comput. Phys.* **394**, 136–152. (doi:10.1016/j.jcp.2019.05.027)
89. Vandal T, Kodra E, Dy J, Ganguly S, Nemani R, Ganguly AR. 2018 Quantifying uncertainty in discrete-continuous and skewed data with Bayesian deep learning. In *Proc. of the 24th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*.
90. Schneider T, Lan S, Stuart A, Teixeira J. 2017 Earth system modeling 2.0: a blueprint for models that learn from observations and targeted high-resolution simulations. *Geophys. Res. Lett.* **44**, 12 396–12 417.

91. Rasp S, Pritchard MS, Gentine P. 2018 Deep learning to represent subgrid processes in climate models. *Proc. Natl Acad. Sci. USA* **115**, 9684–9689. (doi:10.1073/pnas.1810286115)
92. Gentine P, Pritchard MS, Rasp S, Reinaudi G, Yacalis G. 2018 Could machine learning break the convection parameterization deadlock? *Geophys. Res. Lett.* **45**, 5742–5751. (doi:10.1029/2018GL078202)
93. O'Gorman PA, Dwyer JG. 2018 Using machine learning to parameterize moist convection: potential for modeling of climate, climate change, and extreme events. *J. Adv. Model. Earth Syst.* **10**, 2548–2563. (doi:10.1029/2018MS001351)
94. Chattopadhyay A, Subel A, Hassanzadeh P. 2020 Data-driven super-parameterization using deep learning: experimentation with multi-scale Lorenz 96 systems and transfer-learning. *J. Adv. Model. Earth Syst.* e2020MS002084.
95. Bolton T, Zanna L. 2019 Applications of deep learning to ocean data inference and subgrid parameterization. *J. Adv. Model. Earth Syst.* **11**, 376–399. (doi:10.1029/2018MS001472)
96. Baño-Medina J, Manzanas R, Gutiérrez JM. 2020 Configuration and intercomparison of deep learning neural network models for statistical downscaling. *Geosci. Model Dev.* **13**, 2109–2124. (doi:10.5194/gmd-13-2109-2020)
97. Stengel K, Glaws A, Hettinger D, King RN. 2020 Adversarial super-resolution of climatological wind and solar data. *Proc. Natl Acad. Sci. USA* **117**, 16 805–16 815. (doi:10.1073/pnas.1918964117)
98. Scher S, Messori G. 2019 Weather and climate forecasting with neural networks: using general circulation models (gcms) with different complexity as a study ground. *Geosci. Model Dev.* **12**, 2797–2809. (doi:10.5194/gmd-12-2797-2019)
99. Chattopadhyay A, Nabizadeh E, Hassanzadeh P. 2020 Analog forecasting of extreme-causing weather patterns using deep learning. *J. Adv. Model. Earth Syst.* **12**, e2019MS001958.
100. Pathak J, Mustafa M, Kashinath K, Motheau E, Kurth T, Day M. 2020 Using Machine Learning to Augment Coarse-Grid Computational Fluid Dynamics Simulations. (http://arxiv.org/abs/2010.00072).
101. Champion K, Lusch B, Kutz JN, Brunton SL. 2019 Data-driven discovery of coordinates and governing equations. *Proc. Natl Acad. Sci. USA* **116**, 22 445–22 451. (doi:10.1073/pnas.1906995116)
102. Manepalli A, Albert A, Rhoades A, Feldman D. 2019 Emulating numeric hydroclimate models with physics-informed cGANs. In *Climate Change AI Workshop at the 33rd Conf. on Neural Information Processing Systems*.
103. Singh A, Albert A, White B. 2019 Downscaling numerical weather models with GANs. In *Climate Change AI workshop at the 33rd Conf. on Neural Information Processing Systems*.
104. Jiang CM, Esmaeilzadeh S, Azizzadenesheli K, Kashinath K, Mustafa M, Tchelepi HA, Marcus P. 2020 MeshfreeFlowNet: a physics-constrained deep continuous space-time super-resolution framework. In *SC '20: Proc. of the 2020 ACM Conf. on Supercomputing (to appear)*.
105. Wang R, Kashinath K, Mustafa M, Albert A, Yu R. 2020 Towards physics-informed deep learning for turbulent flow prediction. In *Proc. of the 26th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'20)*.
106. Brenowitz ND, Bretherton CS. 2018 Prognostic validation of a neural network unified physics parameterization. *Geophys. Res. Lett.* **45**, 6289–6298. (doi:10.1029/2018GL078510)
107. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. 2014 Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680.
108. King R, Hennigh O, Mohan A, Chertkov M. 2018 From deep to physics-informed learning of turbulence: diagnostics. *Workshop on Modeling and Decision-Making in the Spatiotemporal Domain, NIPS 2018*.
109. Mescheder L, Geiger A, Nowozin S. 2018 Which Training Methods for GANs do actually Converge? (http://arxiv.org/abs/1801.04406).
110. Arjovsky M, Bottou L. 2017 Towards principled methods for training generative adversarial networks. (http://arxiv.org/abs/1701.04862).

111. Yang L, Zhang D, Karniadakis GE. 2020 Physics-informed generative adversarial networks for stochastic differential equations. *SIAM J. Sci. Comput.* **42**, A292–A317. (doi:10.1137/18M1225409)
112. Stinis P, Hagge T, Tartakovsky AM, Yeung E. 2019 Enforcing constraints for interpolation and extrapolation in generative adversarial networks. *J. Comput. Phys.* **397**, 108844. (doi:10.1016/j.jcp.2019.07.042)
113. Isola P, Zhu J, Zhou T, Efros AA. 2017 Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976.
114. Rhoades AM, Jones AD, Ullrich PA. 2018 Assessing mountains as natural reservoirs with a multimetric framework. *Earth's Future* **6**, 1221–1241. (doi:10.1002/2017EF000789)
115. Maberly SC *et al.* 2020 Global lake thermal regions shift under climate change. *Nat. Commun.* **11**, 1232. (doi:10.1038/s41467-020-15108-z)
116. Yang C, Ma C, Yang M. 2014 Single-image super-resolution: a benchmark. In *European Conf. on Computer Vision*, pp. 372–386.
117. Ledig C *et al.* 2017 Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 4681–4690.
118. Vandal T, Kodra E, Ganguly S, Micahelis A, Nemani R, Ganguly AR. 2017 Quantifying uncertainty in discrete-continuous and skewed data with Bayesian deep learning. In *Proc. of the 23rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*.
119. Wang X, Yu K, Wu S, Gu J, Liu Y, Dong C, Qiao Y, Change Loy C. 2018 ESRGAN: enhanced super-resolution generative adversarial networks. In *Proc. of the European Conf. on Computer Vision (ECCV) Workshops*.
120. Dueben PD, Bauer P. 2018 Challenges and design choices for global weather and climate models based on machine learning. *Geosci. Model Dev.* **11**, 3999–4009. (doi:10.5194/gmd-11-3999-2018)
121. Weyn JA, Durran DR, Caruana R. 2019 Can machines learn to predict weather? Using deep learning to predict gridded 500-hpa geopotential height from historical weather data. *J. Adv. Model. Earth Syst.* **11**, 2680–2693. (doi:10.1029/2019MS001705)
122. Chattopadhyay A, Hassanzadeh P, Subramanian D, Palem K. 2020 Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine learning methods: reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Process. Geophys.* **27**, 373–389. (doi:10.5194/npg-27-373-2020)
123. Bocquet M, Brajard J, Carrassi A, Bertino L. 2020 Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization. *Foundations of Data Science* **2**, 55–80.
124. Labourasse E, Sagaut P. 2002 Reconstruction of turbulent fluctuations using a hybrid RANS-LES approach. *J. Comput. Phys.* **182**, 301–336. (doi:10.1006/jcph.2002.7169)
125. Jaderberg M, Simonyan K, Zisserman A. 2015 Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pp. 2017–2025.
126. Cao Y, Geddes TA, Yang JYH, Yang P. 2020 Ensemble deep learning in bioinformatics. *Nat. Mach. Intell.* **2**, 500–508. (doi:10.1038/s42256-020-0217-y)
127. Bihlo A. 2019 A generative adversarial network approach to (ensemble) weather prediction. (http://arxiv.org/abs/2007.07718).
128. Fanfarillo A, Roozitalab B, Hu W, Cervone G. 2020 Probabilistic forecasting using deep generative models. (http://arxiv.org/abs/1909.11865).
129. Rasp S, Dueben PD, Scher S, Weyn JA, Mouatadid S, Thuerey N. 2020 WeatherBench: a benchmark dataset for data-driven weather forecasting. *J. Adv. Model. Earth Syst.* **12**, e2020MS002203.
130. Jouppi NP, Yoon DH, Kurian G, Li S, Patil N, Laudon J, Young C, Patterson D. 2020 A domain-specific supercomputer for training deep neural networks. *Commun. ACM* **63**, 67–78. (doi:10.1145/3360307)