Model Order Reduction using a Deep Orthogonal Decomposition

Daniel J. Tait*
Departments of Computer Science and Statistics
University of Warwick
The Alan Turing Insitutute
dtait@turing.ac.uk

Abstract

Near-term prediction of the structured spatio-temporal processes driving our climate is of profound importance to the safety and well-being of millions, but the pronounced nonlinear convection of these processes make a complete mechanistic description even of the short-term dynamics challenging. However, convective transport provides not only a principled physical description of the problem, but is also indicative of the transport in time of informative features which has lead to the recent successful development of "physics free" approaches. In this work we demonstrate that their remains an important role to be played by physically informed models, which can successfully leverage deep learning (DL) to project the process onto a lower dimensional space on which a minimal dynamical description holds. Our approach synthesises the feature extraction capabilities of DL with physically motivated dynamics to outperform existing model free approaches on complex real world sea surface temperature prediction.

1 Introduction

We are interested in the problem of producing near-term predictions of the complex dynamical systems driving our natural world. Such systems have a significant impact on the daily lives and safety of our planets' population, with important examples including precipitation and sea surface temperature; variables crucial for flood warning and prevention, prediction of cyclogenesis, and maritime safety. A common theme is the dominance of short-term dynamics by convection – the transport of material along the stream lines of a vector field. Transport and the reduced role of diffusion at these time-scales leads to temporal feature preservation, and consequently these systems have received increasing interest from the machine learning (ML) community [21, 22, 14, 24, 2].

The classical approach to solving these problems in the natural sciences is to specify a model of the underling physical properties. A properly calibrated mechanistic model described by parameterised differential operators is able to provide future prediction and generalise to new scenarios, provided this change of physics can be encoded by some parameter. In a lingua franca of ML we would comment on the remarkable generalisation and transfer-ability of fundamental physical theories specified by mathematical models, a demonstration of "the unreasonable effectiveness of mathematics" [26].

Nevertheless, a successful physical theory often requires multiple simplifying assumptions so removing many of the complexities of real-world phenomena, done correctly this reduction allows the theory to focus on the most salient qualities of the system. However, this reduction can be insufficient for detailed prediction in the most complex open world systems we experience in nature, or else require expensive numerical models [13]. This complexity, combined with the existence of the persistent,

*Code for the experiments is available https://github.com/danieljtait/mordod

Workshop on machine learning for engineering modeling, simulation and design @ NeurIPS 2020

informative features that characterise convection has led to the development of deep learning (DL) approaches, notably the dynamics based approaches of [21, 22, 14, 24], or the images-to-images type approaches adopted by [3, 10, 2]. As DL methods continue to advance it becomes an open question as to whether one should still attempt to embed mechanistic structure into our models, or if one ought to go "physics free" [2]. We suggest this stance is information inefficient, ignoring the vast prior scientific knowledge acquired by human endeavour. This viewpoint has led to the emergence of *physics informed* ML [17, 7] which aims to combine the expressive power of modern ML methods, with the representative power of fundamental physical models. Typically these employ DL surrogates of the input-output map, then ensuring this map is either constructed to obey some physical principle [6, 25], or else regularised by a guiding mechanistic model [17, 23, 4, 16, 27, 28].

In this work we use the powerful feature extraction capabilities of modern ML to perform localised model order reduction by projection onto a subspace spanned by deeply-extracted orthogonal features. We then show how one can combine flexible methods with minimal physical representations to advance the dynamics in this reduced space, crucially we construct our method so as to enable a tighter link between the DL-based encoding and the dynamics than previous approaches. Finally we demonstrate that this synthesis of DL feature extraction and minimal physics outperforms both existing model free approaches and state of the art hybrid approaches on complex real world data-sets.

2 Background

2.1 Models for transport and diffusion

Our goal is to introduce a flexible family of models for the evolution of a field variable $u(x,t)$ indexed by spatial coordinates $\mathbf{x}$ in a region $\Omega \subset \mathbb{R}^D$, and temporal coordinate $t \in [t_0, T]$. A starting point for describing diffusion and transport is the PDE (1), which is parameterised by a diffusion coefficient $\alpha(\mathbf{x}, u)$, governing smoothing over time, and a transport vector field, $\mathbf{w}(\mathbf{x}, u)$, describing the direction of bulk motion, this a (pseudo-)linear PDE and so in principle easy to solve. However, this only defers modelling complexity because one must still motivate an appropriate transport vector field. Mathematically the most well studied models jointly describing convective transport are the system 2 of Rayleigh-Bénard equations [20, 8]. However, solving this system across multiple spatial and temporal scales, even for known parameters, presents a formidable challenge.

$$\frac{\partial u}{\partial t} = -\nabla \cdot (\alpha \nabla u) + \mathbf{w} \cdot \nabla \mathbf{u} \tag{1}$$

(a) Pseudo-linear transport

$$\frac{\partial u}{\partial t} + \mathbf{w} \cdot \nabla u = \nabla \cdot (\alpha \nabla u) \tag{2a}$$

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbf{w} \cdot \nabla \mathbf{w} = \nu \Delta \mathbf{w} + f \tag{2b}$$

$$\nabla \cdot \mathbf{w} = 0 \tag{2c}$$

(b) Rayleigh - Bénard convection

Nevertheless, this motivating idea of forming a conditionally linear PDE has been successfully applied by [7]. Their approach uses data to inform the vector field $\mathbf{w}$, and then use the Gaussian integral kernel obtained by solving (1) to motivate an integral-convolution update of the current image of the field variable. They propose to discard the non-linear components (2b) – (2c) of the system (2) when advancing the solution, then reincorporate them in the supervised per-time step loss function

$$\mathcal{L}_t = \sum_{x \in \Omega} \rho(\hat{u}_{t+1}(x) - u_{t+1}(x)) + \lambda_{\text{div}} \|\nabla \cdot \mathbf{w}(x)\|^2 + \lambda_{\text{magn}} \|\hat{\mathbf{w}}(x)\|^2 + \lambda_{\text{grad}} \|\nabla \mathbf{w}(x)\|^2 \tag{3}$$

where ρ is a loss function on the space of pixel-images, $\hat{u}_{t+1}$ is the predicted surface and u_{t+1} is the actual surface. Ultimately, this assumes the linear PDE (2a) is sufficient to capture the dynamics in the original space, instead we shall attempt to project the dynamics onto a space where this assumption becomes more plausible, but first we briefly review how to numerically solve a PDE like (1) or (2).

2.2 Ritz-Galerkin discretisation of PDEs

It is typically necessary to solve a PDE problem, like (1), numerically via discretisation, one powerful method of doing so is to first multiply the problem by a test function $v \in \hat{V}$, and then integrate to

form a variational problem. It is typical to choose tests functions which vanish on the boundary, and so one arrives at the classical *variational form* of the PDE problem: find $u \in V$ such that

$$\int_\Omega \frac{\partial u(t,\mathbf{x})}{\partial t} v(\mathbf{x}) d\mathbf{x} = \int_\Omega \alpha(\mathbf{x}, u) \nabla u(t,\mathbf{x}) \cdot \nabla v(\mathbf{x}) d\mathbf{x} + \int_\Omega \mathbf{w}(\mathbf{x}) \cdot \nabla u(t,\mathbf{x}) v(\mathbf{x}) d\mathbf{x} \qquad (4)$$

for any test function $v \in \hat{V}$, this is the *weak form* of the classical problem (1). In order to numerically implement this idea one must also replace the trial and test spaces by finite dimensional sub-spaces and we take $V = \text{span}\left(\{\varphi_m\}_{m=1}^M\right) = \hat{V}$. While there are multiple choices of basis functions $\{\varphi_m\}_{m=1}^M$, for example one can use the nodal basis functions of the FEM [18], in this work we shall consider sets of orthonormal basis functions. That is we specify a collection of M basis functions such that $\langle \varphi_i, \varphi_j \rangle_{L^2(\Omega)} = \delta_{ij}$, where $\langle \cdot, \cdot \rangle_{L^2(\Omega)}$ is the L^2-inner product.We then search for solutions to (4) with representation $u(t,\mathbf{x}) = \sum_{m=1}^M (\mathbf{z}_t)_m \varphi_m(\mathbf{x})$, where $\mathbf{z}_t \in \mathbb{R}^M$ is a vector of unknown coefficients to be determined. Inserting this representation into (4) we achieve a finite-dimensional projection of the dynamical problem as an ordinary differential equation (ODE)

$$\frac{d}{dt}\mathbf{z} = \mathbf{L}_\varphi \mathbf{z}, \qquad (\mathbf{L}_\varphi)_{ij} = \int_\Omega \alpha(\mathbf{x}) \nabla \varphi_j(\mathbf{x}) \cdot \nabla \varphi_i(\mathbf{x}) dx + \int_\Omega \mathbf{w}(\mathbf{x}) \cdot \nabla \varphi_j(\mathbf{x}) \varphi_i(\mathbf{x}) dx \qquad (5)$$

This is the Ritz-Galerkin projection of the dynamics onto the subspace V. We use the notation $\mathbf{L}_\varphi$ to make it clear that where as the classical operator in the RHS of (1) depended only up on the coefficient functions parameterising it, the projected problem further depends on the ability of the basis functions to capture information about the problem, entwining the encoding and the dynamics. The less informative the basis functions the higher the dimension of M will be needed to faithfully reproduce the dynamics in the function space. In what follows we refer to the process of forming the state dynamic matrix $\mathbf{L}$ in (5) as the `Assembly` operation, which involves performing quadratures to compress the parameter fields $\alpha, \mathbf{w}$ and basis functions into an $M \times M$ matrix.

2.3 Proper orthogonal decompositions

Given an ensemble $\{u_t\}_{t=1}^N$ of field variables over a domain Ω, the *proper orthogonal decomposition* (POD) is a technique for extracting an informative modal basis, that is a low-dimensional basis which captures most of the information or "energy" of the ensemble. The decomposition in POD is more familiar to the machine learning community under the name principal components analysis (PCA), or alternatively the Karhunen-Loeve decomposition [9, 11], and involves reconstructing elements of the ensemble as $\hat{u} = \sum_m \mathbf{z}_k \varphi_m(\mathbf{x})$ where $\{\varphi_m(\mathbf{x})\}_{m=1}^M$ are the first M-eigenfunctions of the empirical covariance matrix of the ensemble ordered by largest eigenvalue.

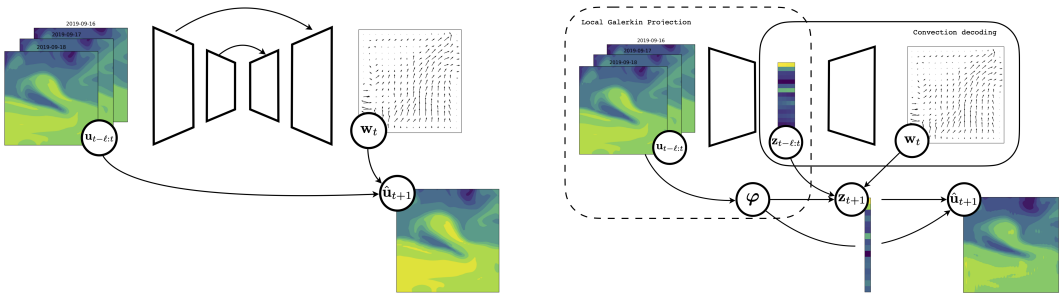
The idea of using the POD eigenfunctions, as an "empircal basis" onto which to perform the above Ritz-Galerkin projection for modelling turbulent flows was introduced in [12]. However, the optimality results concern POD as a linear re-constructor, and do not transfer to any optimality on the dynamic prediction problem [5]. Therefore, in this work we shall instead use deep networks to extract our orthogonal subspace, however motivated by the POD idea we shall attempt to still include some version of this optimal reconstruction to motivate the regulariser in our supervised loss.

3 Methodology

Our objective will be to estimate a future length T sequence of realisations of the process starting from time t, using only information coming from the length ℓ history process $\{u_k(\mathbf{x})\}_{k=t-\ell}^t$. Infact, we shall also discretise the domain as an $n_x \times n_y$ pixelated grid, and instead aim to estimate the vectorised field variable $\mathbf{u} \in \mathbb{R}^{n_x \times n_y}$. That is we aim to learn an images-to-images map $\{\hat{\mathbf{u}}_k\}_{k=t+1}^{k=t+T} = f(\mathbf{u}_{t-\ell}, \ldots, \mathbf{u}_t)$, which also embodies some minimal dynamical representation. To do so we introduce a DL approach to building a localised version of the POD-basis discussed above.

Deep Galerkin Features

In order to project the inputs onto a reduced space it is required to construct a set of functions, $\{\varphi_j\}_{j=1}^M$, such that $\langle \varphi_i, \varphi_j \rangle_{L^2(\Omega)} = \delta_{ij}$, again we shall work with the vectorised images of these functions and replace the $L^2(\Omega)$ orthogonality condition with a quadrature approximation. Furthermore, and



(a) Single time step of [7] (b) Single time-step of our `DOD-Convection` model

Figure 2: (a) The method of [7] uses DL to estimate a motion field and then transport the solution in the original data-space, no further use is made of the latent representations. (b) Our method uses the `GalerkinBasis` network to encode the inputs to a latent space, and then advances the solution in the latent space using information from the encoded variables to parameterise the `Convection` model, before reconstructing in the data-space

without loss of generality, we can re-scale the domain to have unit volume so that we seek a collection of vectors such that $\langle \boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j \rangle = \delta_{ij}, \forall i, j = 1, \dots, M$ under the Euclidean inner product on $\mathbb{R}^{n_x \times n_y}$.

Any given input $\mathbf{u}_t$ can then projected onto an M dimensional space by the Galerkin projection $\Pi_{\text{Galerkin}} : \mathbb{R}^{n_x \times n_y} \to \mathbb{R}^M$ where $\mathbf{z}_t = \Pi_{\text{Galerkin}} \mathbf{u}_t$ is the M-vector with coefficients $(\mathbf{z})_j = \langle \mathbf{u}, \boldsymbol{\varphi}_j \rangle$, for $j = 1, \dots, M$. While the POD method discussed in the previous section constructs this basis vector using the *complete* set of inputs, we wish to perform our projection using only the most recent ℓ inputs, relying on the power of DL approaches to extract sufficient information from this reduced set. Our chosen feature extractor will be the ubiquitous U-Net [19], owing to its demonstrated successes in performing feature extraction from image data. Since we seek M basis functions we shall consider architectures which takes as input an image sequence of shape (n_x, n_y, ℓ), and outputs an image sequence of shape (n_x, n_y, M). Our `GalerkinBasis` model is then a composition of this map, with an orthogonalisation of the outputs and we write

$$\{\boldsymbol{\varphi}_j^{(t)}\}_{j=1}^M = \texttt{GalerkinBasis}(\mathbf{u}_{t-\ell}, \dots, \mathbf{u}_t) = \texttt{Orthogonalise} \circ \texttt{Unet}(\mathbf{u}_{t-\ell}, \dots, \mathbf{u}_t). \tag{6}$$

Functionally, we have constructed the basis of vectors $\boldsymbol{\varphi}_j^{(t)} = \boldsymbol{\varphi}_j^{(t)}(\mathbf{u}_{u_{t-\ell}}, \dots, \mathbf{u}_t)$, which depend on temporally local values. In practice we use a QR decomposition to perform this orthogonalisation.

Linear latent convection model

The `GalerkinBasis` provides our desired nonlinear transformation of the input sequence onto a linear space parameterised by $\mathbf{z} \in \mathbb{R}^M$, using only information up to time t. For the remainder of the prediction horizon we wish to forward solve the problem using the physically informed convection dynamics on a linear space determined by the extracted features. We shall assume that in this space the dynamics are reasonably well described by the linear PDE (1), and use our learned features to carry out a projection of the dynamics onto a local processes obtained via (5).

It remains to parameterise the dynamics with a diffusion coefficient, and a transport vector field. We shall refer to the model component performing this parameterisation as the `Convection` model, and we shall also allow it to be informed by the previous ℓ observations. Once this component has been specified we shall advance the latent state according to an updating scheme of the form

$$\alpha^{(k)}, \mathbf{w}^{(k)} = \texttt{Convection}(\mathbf{z}_{k-\ell}, \dots, \mathbf{z}_k), \tag{7a}$$

$$\mathbf{L}_\varphi^{(k)} = \texttt{Assembly}(\alpha^{(k)}, \mathbf{w}^{(k)}, \{\varphi_m(\mathbf{u}_{t-\ell}, \dots, \mathbf{u}_t)\}_{m=1}^M) \tag{7b}$$

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \int_{t_k}^{t_{k+1}} \mathbf{L}_\varphi \mathbf{z}_\tau d\tau \tag{7c}$$

for $k = 0, \dots, T-1$. First the previous values of the latent process are used to build new parameters of the transport model (7a), these are then combined with the features to assemble the state dynamic matrix (7b), and finally the linear ODE (5) is solved to give the new state, (7c), and this process is repeated until a complete set of T latent states are obtained. Our general approach to parameterisation

of the transporting physical model is therefore similar to that in [7], but crucially our approach propagates the dynamics in the *latent space* by way of the Galerkin projection, rather than applied to the complete surface. This leads to an important difference since any surface now has a finite dimensional representation by $\mathbf{z}_t$ our convection step (7a) may be taken as a function of the low dimensional latent states, that is we can consider a decoder which takes the $\ell \times M$ lagged latent states and outputs the diffusion and transport field over Ω this is the top-right block of Fig. 2b.

Conversely, the approach taken in [7] requires one to first encode the surfaces, and then decode them again to estimate the motion field, see Fig. 2a. While this step necessarily creates an implicit latent representation as a byproduct of this encode/decode step, this representation has no further role to play in the model dynamics. In contrast, since we have already performed encoding through the Galerkin projection, we require no further compression allowing the latent representations produced by our approach to feature more directly in the dynamical description, see Fig. 2b.

4 Experiments

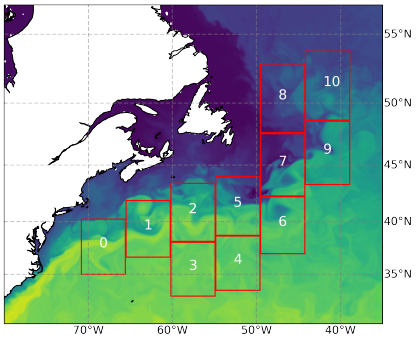


Figure 3: Regions used for the sea surface temperature experiment

We now apply our method to the large-scale convective systems encountered in the climate sciences in order to assess the accuracy and computational efficiency of our method compared to alternatives. All experiments were conducted on a single Nvidia GeForce RTX 2060 SUPER GPu with 8GB of memory. As in [7] we demonstrate our model on the SST data obtained from the NEMO (Nucleus for European Modelling of the Ocean) engine [13] which uses assimilation to accurately reflect real world temperature history. We extract the 64 by 64 pixel sub-regions shown in Fig. 3, and use data from 2016-2018 to train the model, a give a total of 11935 acquisitions. We then test on data from the years 2019 using the same regions giving 3894 test instances. The regions were selected to capture the complicated nonlinear mixing that occurs as warm-water is transported along the gulf stream to meet colder water.

We compare the method we have introduced to the physics informed model of [7], as well as a convolutional LSTM (Conv-LSTM) introduced by [21, 22] which enhances LSTM networks with a spatial convolution to better model spatio-temporal process, and finally to compare with "phyiscs-free" approaches we use a U-Net as proposed in [3, 10, 2] which treats prediction as an images-to-images problem, and makes no attempt to embed recurrent dynamical or physical structure. All models were implemented in Tensorflow [1], apart from [7] for which we use publicly available code. [2]

Table 1: Comparison of methods on the SST data. Average score is the mean squared error (MSE), No. of parameters is the total number of trainable parameters in each model, and run-time is the mean time per-batch with the maximum batch size that can fit in memory. We fit our model with $M = 16$ $\boldsymbol{\varphi}$-features

	Average Score (MSE)	No. of Parameters	Run-Time [s]
Conv-LSTM[22]	0.2073	1,779,073	0.43
U-Net [10, 2]	0.1473	31,032,446	0.79
Flow [7]	0.1304	22,197,736	0.60
DOD-Convec	**0.1132**	10,106,339	0.48

Results are displayed in Table 1, examining the test error for each method we see that our method demonstrates superior performance with a lower test error than the purely data-driven approaches and the more dynamic alternatives. In Fig. 4 we plot a representative sequence from the test set, in which we observe that our method, row three, seems to be better at identifying the "whorl" pattern formations that characterise turbulence, but that the U-Net feature extraction also does a job job of

[2] A PyTorch [15] implementation for the model of [7] is available at `https://github.com/emited/flow`

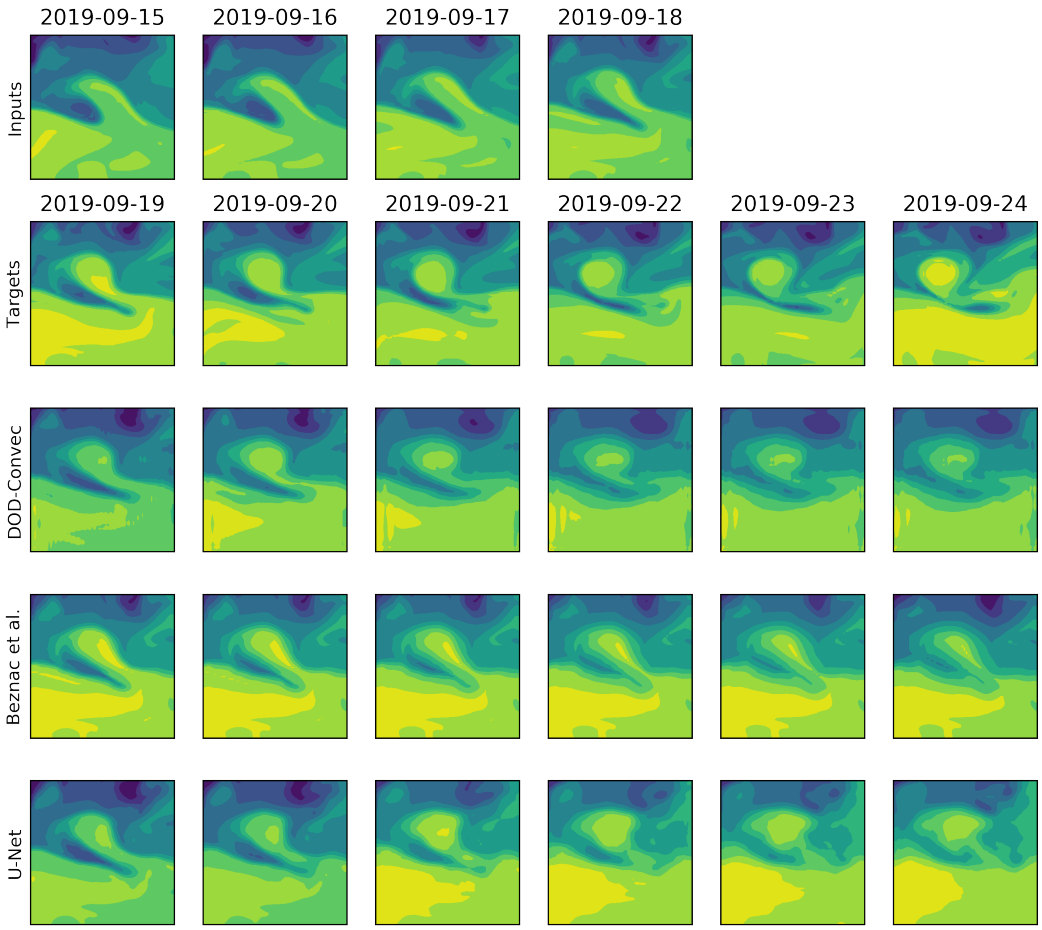


Figure 4: Predicted surfaces from an input sequence of length four, top row, compared to a target output sequence of length six, second row. from a test set sequence in Region 2 of Fig. 3

identifying these features, but only on a limited time-horizon with the structure degrading as the sequence continues, this is most noticeable in the loss of structure of the in-flowing cold flux in the top-right of the images in row four. On the other-hand the method of [7] does a better job capturing and preserving linear features, this is likely because this method is ultimately solving a linear PDE (2a), and identifying a convection model from data alone that will do the "violence" [5] of a nonlinear model from data alone is hard. By projecting to a latent space, and allowing this linear space to be adaptivly determined by nonlinear transformations of the local inputs we are better able to recover nonlinear features with a simpler convection model.

5 Discussion

In this work we have combined the powerful feature extraction capabilities of DL, with minimal physical representations of the dynamics to introduce a physically informed model demonstrating superior predictive performance on the convective physical processes that dominate the atmospheric and fluid transport problems pervading our natural world. To the extent that this model is "physically informed" we have a priori specified only a minimal representation of the dynamics, we justify this by our desire to avoid overly strong, and so impossible to justify, assumptions on the complex generating mechanism and therefore maintain as much as possible the ability of the model to learn the underlying dynamics from data. Crucial to this has been our efforts to ensure that the latent representations formed by DL-encoding are more strongly entwined with the dynamics of the process than previous approaches, leading to a model that demonstrates superior predictive performance, and improved recovery of temporally persistent nonlinear features.

Acknowledgments and Disclosure of Funding

This work is supported by HVM Catapult project number 8205 and Lloyd's Register Foundation together with the Alan Turing Institute. Thank you to Dr. Theodoros Damoulas for helpful comments, feedback and guidance.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[2] Shreya Agrawal, Luke Barrington, Carla Bromberg, John Burge, Cenk Gazen, and Jason Hickey. Machine learning for precipitation nowcasting from radar images, 2019.

[3] G. Ayzel, M. Heistermann, A. Sorokin, O. Nikitin, and O. Lukyanova. All convolutional neural networks for radar-based precipitation nowcasting. *Procedia Computer Science*, 150:186 – 192, 2019. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2019.02.036. URL `http://www.sciencedirect.com/science/article/pii/S1877050919303801`. Proceedings of the 13th International Symposium "Intelligent Systems 2018" (INTELS'18), 22-24 October, 2018, St. Petersburg, Russia.

[4] Jens Berg and Kaj Nyström. A unified deep artificial neural network approach to partial differential equations in complex geometries, 2017.

[5] G Berkooz, P Holmes, and J L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25(1):539–575, 1993. doi: 10.1146/annurev.fl.25.010193.002543. URL `https://doi.org/10.1146/annurev.fl.25.010193.002543`.

[6] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1321–1330, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL `http://proceedings.mlr.press/v97/cohen19d.html`.

[7] Emmanuel de Bezenac, Arthur Pajot, and Patrick Gallinari. Deep learning for physical processes: Incorporating prior scientific knowledge. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=By4HsfWAZ`.

[8] A. V. Getling and Edward A. Spiegel. *Rayleigh-Bénard Convection: Structures and Dynamics*. Columbia University, New York, 1998.

[9] K. Karhunen. Zur spektraltheorie stochastischer prozesse. *Annales Academiæ Scientiarum Fennicæ, Series A*, 1946.

[10] Vadim Lebedev, Vladimir Ivashkin, Irina Rudenko, Alexander Ganshin, Alexander Molchanov, Sergey Ovcharenko, Ruslan Grokhovetskiy, Ivan Bushmarinov, and Dmitry Solomentsev. Precipitation nowcasting with satellite imagery. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2680–2688, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330762. URL `https://doi.org/10.1145/3292500.3330762`.

[11] M. Loeve. Fonctions aleatoires de second ordre. *Comptes Rendus De L'Académie Des Sciences*, 220, 1945.

[12] J.L. Lumley. Coherent structures in turbulence. In RICHARD E. MEYER, editor, *Transition and Turbulence*, pages 215 – 242. Academic Press, 1981. ISBN 978-0-12-493240-1. doi: https://doi.org/10.1016/B978-0-12-493240-1.50017-X. URL `http://www.sciencedirect.com/science/article/pii/B978012493240150017X`.

[13] G. Madec. NEMO ocean engine. Technical report, Institut Pierre-Simon Laplace IPSL, 2008.

[14] Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL `http://arxiv.org/abs/1511.05440`.

[15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf,

Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8026–8037. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[16] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686 – 707, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL http://www.sciencedirect.com/science/article/pii/S0021999118307125.

[17] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, 2017.

[18] J Reddy. *An Introduction to the Finite Element Method*. McGraw-Hill Education, New York, 2005.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

[20] Barry Saltzman. *Selected Papers on the THEORY OF THERMAL CONVECTION with special application to the earth's planetary atmosphere*. Dover, 1962.

[21] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *CoRR*, 2015.

[22] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Deep learning for precipitation nowcasting: A benchmark and a new model. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5617–5627. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7145-deep-learning-for-precipitation-nowcasting-a-benchmark-and-a-new-model.pdf.

[23] Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339 – 1364, 2018. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.08.029. URL http://www.sciencedirect.com/science/article/pii/S0021999118305527.

[24] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 613–621. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6194-generating-videos-with-scene-dynamics.pdf.

[25] Maurice Weiler and Gabriele Cesa. General e(2)-equivariant steerable cnns. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 14334–14345. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9580-general-e2-equivariant-steerable-cnns.pdf.

[26] Eugene P. Wigner. The unreasonable effectiveness of mathematics in the natural sciences. richard courant lecture in mathematical sciences delivered at new york university, may 11, 1959. *Communications on Pure and Applied Mathematics*, 13(1):1–14, 1960. doi: 10.1002/cpa.3160130102. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160130102.

[27] Alireza Yazdani, Maziar Raissi, and George Em Karniadakis. Systems biology informed deep learning for inferring parameters and hidden dynamics. *bioRxiv*, 2019. doi: 10.1101/865063. URL https://www.biorxiv.org/content/early/2019/12/04/865063.

[28] Yinhao Zhu, Nicholas Zabaras, P. Koutsourelakis, and Paris Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 394, 05 2019. doi: 10.1016/j.jcp.2019.05.024.