Deep Orthogonal Decompositions for Convective Nowcasting

Daniel J. Tait
University of Warwick
Alan Turing Institute
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 prounced 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 to the now-casting problem. 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, as well as state of the art hybrid approaches, on complex real world datasets including sea surface temperature and precipitation.

1 Introduction

Nowcasting refers to the process 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 [23, 24, 16, 26, 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" [29].

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 [15]. This complexity, combined with the existence of the persistent, informative features that characterise convection has led to the development of deep learning (DL) approaches, notably the dynamics based approaches of [23, 24, 16, 26], or the images-to-images type

approaches adopted by [3, 12, 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 [19, 8] 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 [7, 28], or else regularised by a guiding mechanistic model [19, 25, 5, 18, 31, 33]

In this work we add to the argument that physical principles still have an important role to play in modelling by introducing a new physically informed method motivated by two of the guiding principles behind a successful mathematical analysis of a complex dynamical problem; (i) the projection of a high dimensional model onto a lower dimensional representation, and (ii) the existence of a transformation that brings the dynamics closer to linear. We use the feature extraction prowess of DL to perform our reduction, while using physical principles to guide the evolution of these features, constructing a flexible system that is well suited to modelling convection phenomena by ensuring that deep nonlinear components are strongly linked with the minimal physical dynamics.

In summary, 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

The classical approach to model based forecasting predicts the evolution of a variable using past records to calibrate a mechanistic model. These models are typically complex systems of partial differential equations (PDEs), which must be sufficiently rich to embed all relevant physical knowledge. These systems can be used to describe the change in intensity of some target variable, for example sea surface temperature or precipitation rate. Two of the most important physical processes describing these changes are diffusion – a smoothing transformation, and transport – volume preserving translations, and we now provide a brief review of the mathematical models encoding these effects.

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 [22, 10]. However, solving this system across multiple spatial

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

(a) Pseudo-linear transport

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

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

$$\nabla \cdot \mathbf{w} = 0 \qquad (2c)$$

(b) Rayleigh - Bénard convection

and temporal scales, even for known parameters, presents a formidable challenge.

Nevertheless, this motivating idea of forming a conditionally linear PDE has been successfully applied by [8]. 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

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

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

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

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 [20], 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 (10) 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 (10) we achieve a finite-dimensional projection of the dynamical problem as an ordinary differential equation (ODE)

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

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 [11, 13], 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 [14]. However, the optimality results concern POD as a linear re-constructor, and do not transfer to any optimality on the dynamic prediction problem [6]. 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.
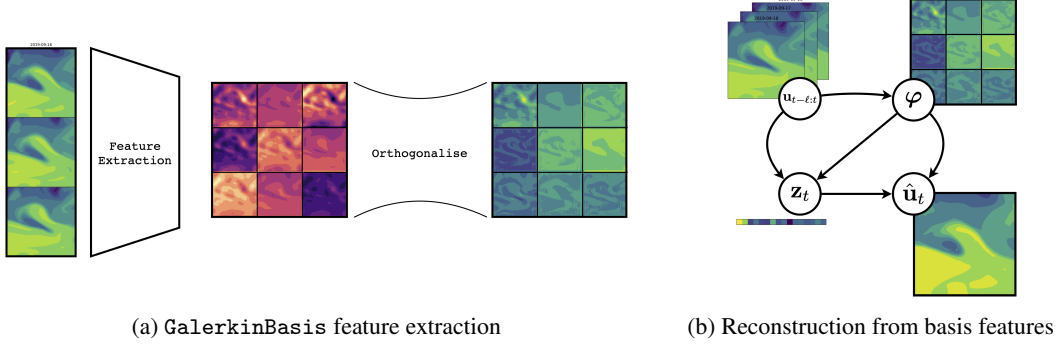
(a) `GalerkinBasis` feature extraction   (b) Reconstruction from basis features

Figure 2: (a) A local input sequence $\{\mathbf{u}_k\}_{t-\ell}^t$ is passed through the `GalerkinBasis` feature extraction, to produce a collection of M orthogonal images $\{\boldsymbol{\varphi}_m\}_{m=1}^M$. (b) An image can be projected onto the subspace $\mathrm{span}\{\boldsymbol{\varphi}_m\}$ and then reconstructed as the linear combination $\hat{\mathbf{u}}_t = \sum_{m=1}^M \langle \mathbf{u}_t, \boldsymbol{\varphi}_m \rangle \boldsymbol{\varphi}_m$.

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 $\{\hat{\mathbf{u}}_k\}_{k=t+1}^{k=t+T} = f(\mathbf{u}_{t-\ell}, \dots, \mathbf{u}_t)$, 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 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 [21], 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

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

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 only on temporally local values, an overview of the composed process is represented in Fig. 2.

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).

(a) Single time step of [8]

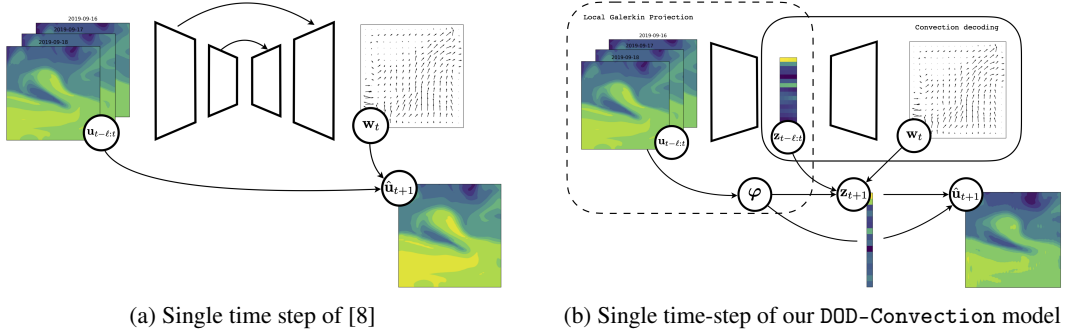(b) Single time-step of our `DOD-Convection` model

Figure 3: (a) The method of [8] 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 only information from the encoded variables to parameterise the `Convection` model, before finally reconstructing in the data-space

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 [8], 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. 3b.

Conversely, the approach taken in [8] requires one to first encode the surfaces, and then decode them again to estimate the motion field, see Fig. 3a. 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. 3b.

In summary, our model first uses the output of the `GalerkinBasis` to construct a set of orthogonal feature images and then uses the Ritz-Galerkin method to project the dynamics onto this space. The encoded variables are then passed to the `Convection` model to determine the dynamics, but notably while the convection model in Fig. 3a must account for all of the transformation applied to an input image, our model is able to share the dynamical description with the φ-features through the projection (5). This allows for a more parsimonious parameterisation of the the `Convection` model, and since only this smaller model is called repeatedly during the loop procedure (3) we achieve a more memory efficient implementation. Recognising the partition of our method into an orthogonal decomposition, and then a convection informed forward integration, we shall refer to it as a Deep Orthogonal Decomposition for Convection model, or simply `DOD-Convection`.

Reconstruction and supervised loss

Finally, having created a full set of latent states $\{\mathbf{z}^{(k)}\}_{k=t-\ell}^{k=T+t}$, we may use our Galerkin features to reconstruct the process. Note that we have both the historical states, $\{\mathbf{z}_k\}_{k=t-\ell}^{t}$, and the future states $\{\mathbf{z}_k\}_{k=t+1}^{T}$, we also further recall from our discussion of the POD method in Section 2.3 that when combined with the Galerkin features, the historical states should do a good job of reconstructing the inputs, but that the POD method produces the optimal such reconstruction. Together this partition into historical and future states suggests the following decomposition of our supervised loss function

$$\mathcal{L}_t = \sum_{k=1}^{T} \rho \left(u_{t+k} - \sum_m (\mathbf{z}_{t+k})_m \varphi_m(\mathbf{x}) \right) + \lambda_{\text{div}} \|\nabla \cdot \mathbf{w}\|^2 \tag{8a}$$

$$+ \lambda_{\text{recon}} \sum_{k=1}^{\ell} \rho \left(u_{t-k} - \sum_m (\mathbf{z}_{t-k})_m \varphi_m(\mathbf{x}) \right). \tag{8b}$$

The first two terms of (8b) are analogous to those in (3), playing a similar role with regards to measuring the accuracy of the prediction through a metric ρ, while regularising the divergence of the output $\mathbf{w}$ of the `Convection` to ensure a valid divergence free vector field.

The final term is specific to our method, and addresses the optimality condition of the POD basis as discussed in Section 2.3, in particular we commented that when used to reconstruct an ensemble the POD basis gives the minimal reconstruction error for a particular dimension of the subspace. The coefficient λ_{recon} therefore controls the extent to which our local basis decomposition attempts to recover a version of this optimally condition. This trade-off helps ensure that the `GalerkinBasis` continues to have an identifiable role as an encoder/decoder of the surfaces given its additional role in assembling the discretised differential operator $\mathbf{L}_\varphi$. This dual role of the φ-features is one of the principal strengths of our method, allowing dynamical components lost by the reduction to the overly simplistic PDE (1), to be recaptured by the DL-features, at the expense of the optimality of these same features as re-constructors of the surface.

4 Experiments

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. For additional experiments and visualisations see the supplementary material.

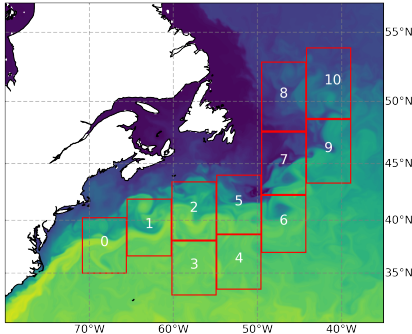### 4.1 Convective seasurface temperature

Figure 4: Regions used for the sea surface temperature experiment

As in [8] we demonstrate our model on the SST data obtained from the NEMO (Nucleus for European Modelling of the Ocean) engine [15]. We extract the 64 by 64 pixel sub-regions shown in Fig. 4, 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 and meets the colder water of the arctic regions. We assume that on the short-term horizons we are interested in, the information within a region is sufficient for prediction and do not incorporate information from adjacent regions. Data is normalised to remove trends on a yearly timescale by taking the mean and standard deviations over the day of the year.

We compare the method we have introduced to the physics informed model of [8], as well as a convolutional LSTM (Conv-LSTM) introduced by [23, 24] which enhances LSTM networks with a

6

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, 12, 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 [8] for which we use publicly available code. [1]

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 $M = 16$ φ-features

	AVERAGE SCORE (MSE)	NO. OF PARAMETERS	RUN-TIME [S]
CONV-LSTM[24]	0.2179	1,779,073	0.43
U-NET [12, 2]	0.1452	31,032,446	0.79
FLOW [8]	0.1344	22,197,736	0.60
DOD-CONVEC	**0.1242**	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. As discussed in Sec. 3 our approach leads to a more parsimonious parameterisation of the convection field than [8]. With regards to run-time the U-Net model is faster, and more memory efficient than the alternatives, including our own, which make use of a recurrent structure to advance the predictions. In spite of this our method still demonstrates competitive run time, with the increased accuracy a more than acceptable trade-off.

In Fig. 5 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 identifying these features, but only on a limit 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 [8] 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" [6] 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, and while we do pay a visible price in the smoothness of our reconstructions, overall we achieve better performance as presented in Tab. 1.

5 Related work

Spatio-temporal dynamical modelling Traditional dynamical modelling approaches have been based on either optical flows [9, 30] which estiamte the transport vector field by examining the differenced input sequence, or else numerical models [4] where the transporting field emerges as the output of a calibrated physical model. More principled physical models have the ability to display complex long-range spatio-temporal dependencies, but deriving and then calibrating such models is significantly harder than the DL and hybrid methods we consider.

Physics free deep learning As noted by [2] "Physics free" deep learning approaches can be boradly divided into attempts to model the time evolution as a temporal sequence, or else those which treat the problem as learning an unknown "image(s)-to-images" map. The former include the convolutional LSTM model of [23, 24], as well as video generation methods [16, 26] For the latter option U-Net based approaches have been popular, [3, 12] and [2] who argue that the overly simplistic assumptions in optical flow or poorly specified numerical models are often unviable, so that free-form DL methods display superior performance methods tied to a misspecified physical model. While our work does use an overly simplistic model of the dynamics, by projecting on the latent space we are able to ensure this simplification does not overly constrain the resulting predictions.

Physics informed deep learning Aside from the work of [8] which we have discussed in Section 2, we also mention the more complicated U-Net architecture inspired by multiscale approximations to

---

[1]A PyTorch [17] implementation for the model of [8] is available at `https://github.com/emited/flow`
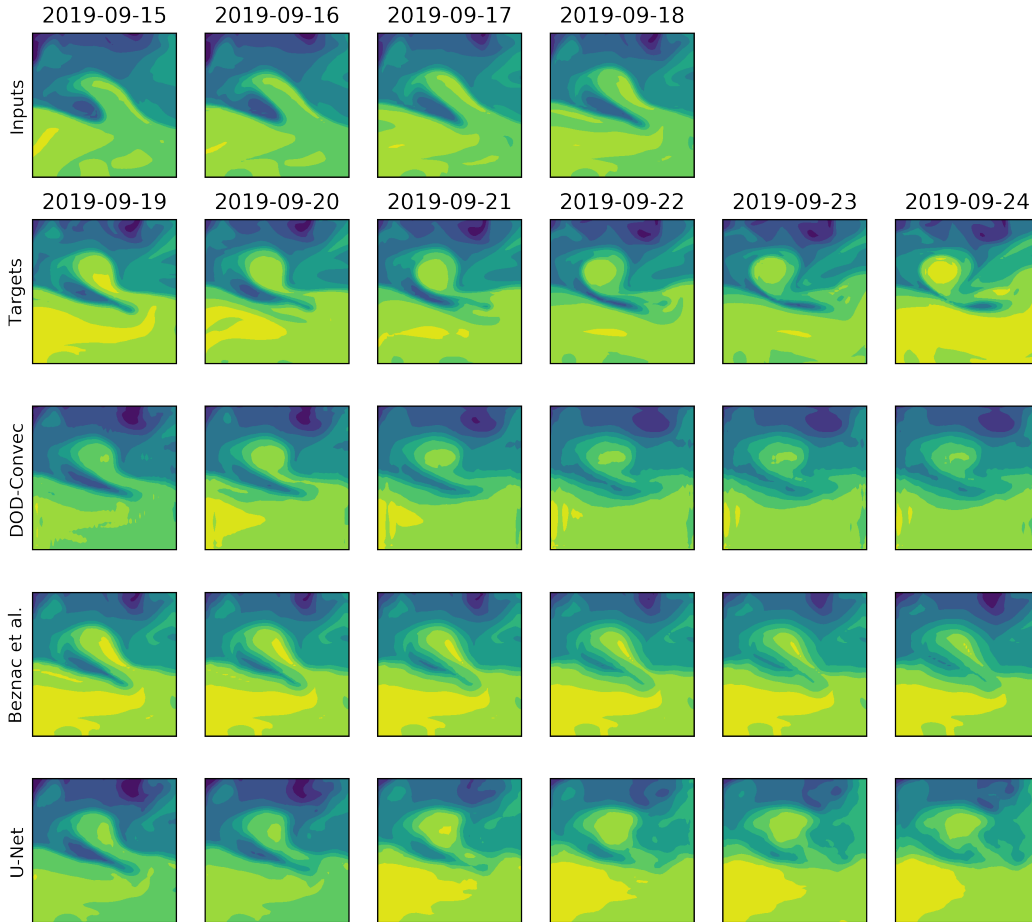
Figure 5: Predicted surfaces from an input sequence of length four, top row, compared to a target output sequence of length six, second row. from a test set sequence in Region 2 of Fig. 4

the Navier-Stokes equations in [27], since this model uses physical information to design a specific U-Net architecture there some overlap with the already mentioned images-to-images methods. Finally we note that we have advanced the solution to the problem by solving something which closely approximates the solution of a PDE. Alternative approaches [19, 5, 33] have avoided this explicit solve step and instead use flexible DL surrogates of the output map, with the PDE operators re-appearing as regularisers. This is a much more dramatic regularisation than the λ_{div} loss we have considered, and we are unaware of this complete operator regularisation scheme having been applied to convective non-linear systems at the scale considered in the experiments.

6 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.

Broader Impact

Climate change manifests not only as long-term trends, but also as the increased short-term volatility of these processes, and as such the development of near-term prediction methods which can be rapidly evaluated and re-calibrated is only increasing in importance. Accurate near term prediction of rapidly changing events is vital for public safety in situations including extreme flooding, cyclone formation, and forest fires – all of these systems possessing the informative transport features which would allow our method to be successfully applied. These applications will continue to add to the evidence we have already provided that physical principles still have an important role to play in designing models, and argue against the "physics free" philosophy. We demonstrate that there is much to be gained from strongly connecting learned latent representations with global model dynamics, achieving not only interpret-able latent spaces, but also interpretable dynamics on these spaces. The resulting models demonstrate better performance, and have a clearer physical picture, and we envision our work continuing to invite methodological study of the generalisation benefits to be obtained by this stronger entwining between encoded representations and the original data-space dynamics.

Despite the progress we have made, predicting these systems remains a hard problem, and it is unlikely that any one method will work best in all scenarios and so appropriate aggregation strategies should be adopted. However, our discussion of the sea surface predictions in Sec. 4 demonstrates that our model does a better job of predicting certain nonlinear patterns in convective flows, and so will be an effective complement to those existing methods which perform well on more linear flows. Finally while we have presented an effective supervised learning method, it will be important to extend the work we have done to a generative probabilistic framework to provide further uncertainty quantification.

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] Dominique Béréziat and Isabelle Herlin. Coupling dynamic equations and satellite images for modelling ocean surface circulation. *Communications in Computer and Information Science*, 550:191–205, 2015. doi: 10.1007/978-3-319-25117-2_12. URL `https://hal.inria.fr/hal-01245369`.

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

[6] 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`.

[7] 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`.

[8] 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`.

[9] David J. Fleet and Y. Weiss. Optical flow estimation. In Nikos Paragios, Yunmei Chen, and Olivier D. Faugeras, editors, *Handbook of Mathematical Models in Computer Vision*. Springer, 2006.

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

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

[12] 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`.

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

[14] 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`.

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

[16] 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`.

[17] 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`.

[18] 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`.

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

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

[21] 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.

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

[23] 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.

[24] 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`.

[25] 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`.

[26] 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`.

[27] Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction, 2020. URL https://openreview.net/forum?id=Hkg5lAEtvS.

[28] 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.

[29] 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.

[30] W.-C. Woo and W.-K. Wong. Operational application of optical flow techniques to radar-based rainfall nowcasting. *Atmosphere*, 8(5):48 – 68, 2017.

[31] 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.

[32] Jian Zhang, Kenneth Howard, Carrie Langston, Brian Kaney, Youcun Qi, Lin Tang, Heather Grams, Yadong Wang, Stephen Cocks, Steven Martinaitis, Ami Arthur, Karen Cooper, Jeff Brogden, and David Kitzmiller. Multi-radar multi-sensor (mrms) quantitative precipitation estimation: Initial operating capabilities. *Bulletin of the American Meteorological Society*, 97(4):621–638, 2016. doi: 10.1175/BAMS-D-14-00174.1. URL https://doi.org/10.1175/BAMS-D-14-00174.1.

[33] 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.

Appendices

A Ritz-Galerkin method for PDEs

Since the forward map is unavailable in closed form it becomes necessary to solve the PDE numerically. To accompany the necessarily brief review in the main paper we provide an extended presentation of how to discretise a PDE on the model Poisson problem, though see also [20] for more information

$$-\Delta u(\mathbf{x}) = f(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega \tag{9a}$$
$$u(\mathbf{x}) = g(\mathbf{x}) \quad \text{on } \partial\Omega. \tag{9b}$$

To discretise one first multiplies (9a) by some test function $v \in \hat{V}$ and then performs an integration by parts, it is standard to further assume that the test functions vanish on the boundary, so one arrives at the classical form of the *variational problem*: find $u \in V$ such that

$$\int_\Omega \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x}) d\mathbf{x} = \int_\Omega f(\mathbf{x}) v(\mathbf{x}) d\mathbf{x} - \int_{\partial\Omega} g(\mathbf{s}) v(\mathbf{s}) ds \tag{10}$$

for any test function $v \in \hat{V}$, this is the *weak form* of the classical problem (9).

To numerically implement this idea one replaces the trial and test spaces by finite dimensional sub-spaces V_h and $\hat{V}_h$ respectively, with corresponding basis functions denoted by ϕ_i, and $\hat{\phi}_i$.

Once we have specified this basis we can then search for solutions with a representation of the form $u(\mathbf{x}) = \sum_{j=1}^M (\boldsymbol{\xi})_j^u \phi_j(\mathbf{x})$, where $\boldsymbol{\xi}^u \in \mathbb{R}^M$ is the vector of coefficients parameterising the finite-dimensional representation of u. Inserting this representation into (10) we arrive at the finite-dimensional version on the weak-form given by the system of equations

$$\sum_{j=1}^N (\boldsymbol{\xi}^u)_j \int_\Omega \nabla \phi_j(\mathbf{x}) \cdot \nabla \hat{\phi}_i(\mathbf{x}) d\mathbf{x} = \int_\Omega f(\mathbf{x}) \hat{\phi}_i(\mathbf{x}) d\mathbf{x} - \int_{\partial\Omega} g(\mathbf{x}) \hat{\phi}_i(\mathbf{s}) d\mathbf{s},$$

which must hold for each basis vector $\hat{\phi}_i, i = 1, \ldots, M$. We may represent this in equivalent matrix-vector notation as $\mathbf{A}\xi^u = \mathbf{f}$, where the *stiffness matrix* and *load vector* are given by

$$(\mathbf{A})_{ij} = \int_\Omega \nabla\phi_j(\mathbf{x}) \cdot \nabla\hat{\phi}_i(\mathbf{x})dx \tag{11a}$$

$$(\mathbf{f})_i = \int_\Omega f(\mathbf{x})\hat{\phi}_i(\mathbf{x})dx - \int_{\partial\Omega} g(\mathbf{s})\hat{\phi}_i(\mathbf{s})d\mathbf{s}. \tag{11b}$$

Following the same procedure we can arrive at a weak-form discretisation of the transport equation (1). In this case the discrete representation analogous to (11a) is given by $\mathbf{L}[\mathbf{z}]$ where

$$(\mathbf{L}[\mathbf{z}])_{ij} = \int_\Omega a(\mathbf{x})\nabla\phi_j(\mathbf{x}) \cdot \nabla\hat{\phi}_i(\mathbf{x})dx + \int_\Omega \mathbf{w}(\mathbf{x}) \cdot \nabla\phi_j(\mathbf{x})\hat{\phi}_i(\mathbf{x})dx. \tag{12}$$

B   Network architectures

U-Net architectures

We use a U-Net [21] for both our basis feature extractor, and in its own right as a benchmark DL method. These take the standard U-Ne form being divided into an encoder and a decoder, the encoder is a sequence of computational units called the *Downsample Blocks* and the encoder a sequence of *Upsample Blocks*, long range skips connections are then provided to connect each downsample block with the corresponding upsampling block in the decoding phase. The basis form of these blocks is given by

- Downsample Block: MaxPooling $\rightarrow$ Conv2D $\rightarrow$ ReLU $\rightarrow$ Conv2D $\rightarrow$ ReLU
- Upsample Block: Upsample $\rightarrow$ Conv2D $\rightarrow$ Relu $\rightarrow$ Conv2D $\rightarrow$ Relu $\rightarrow$ Conv2D

where the upsampling operation doubles the pixel resolution in both directions using nearest neighbour interpolation.

The U-Net used in the experiments has four down-sample and up-sample blocks, the smaller U-Net used as our feature extractor has three.

Convolution decoder

As discussed in Sec. 3 our method estimates the transport vector field and diffusion coefficient from the latent variables $\mathbf{z} \in \mathbb{R}^{\ell \times M}$, this is done in two phases, first using a sequence of dense networks, and then convolutions. For the 64×64 image patches used in the sea surface temperature experiment this network takes the form

- Dense phase: Flatten $\rightarrow$ Dense(64, 'relu') $\rightarrow$ Dense(256, 'relu') $\rightarrow$ Dense(1024, 'relu') $\rightarrow$ Dense(1024, 'relu') $\rightarrow$ Dense(1024, 'relu')
- Convolution phase: Reshape$((4, 4, 64))$ $\rightarrow$ Upsample $\rightarrow$ Conv2D(32, 3, 'relu') $\rightarrow$ Upsample $\rightarrow$ Conv2D(16, 3, 'relu') $\rightarrow$ Upsample $\rightarrow$ Conv2D(8, 3, 'relu') $\rightarrow$ Upsample $\rightarrow$ Conv2D(3, 3)

where Dense$(n, $'relu'$)$ is a densely connected layer with n features and ReLU activation function. Similarly Conv2D$(n, k, $'relu'$)$ is a 2D convolution layer with n features and a $k \times k$-kernel operation. The Upsample operation is the same as for the U-Net architecture just described. The Flatten operation takes the sequence of latent states and flattens them into a single $\ell \times M$ vector.

The output of this as $(64, 64, 3)$ image, final we split on the last dimension to produce the $(64, 64, 1)$ diffusion field, and the $(64, 64, 2)$ transport vector field, and the diffusion field is passed through an additional softplus layer to ensure positivity.

C   Further experimental details

A   NEMO Sea Surface Temperature data

The NEMO (Nucleus for European Modelling of the Ocean) engine [15] is a state-of-the-art framework for modelling the convective processes that drive the worlds oceans, including temperature, salinity, sea ice area and thickness etc., and is provided by the Copernicus Marine Service portal with product identifier `GLOBAL_ANALYSIS_FORECAST_PHY_001_024`. While it is a simulation engine, it accumulates historical data
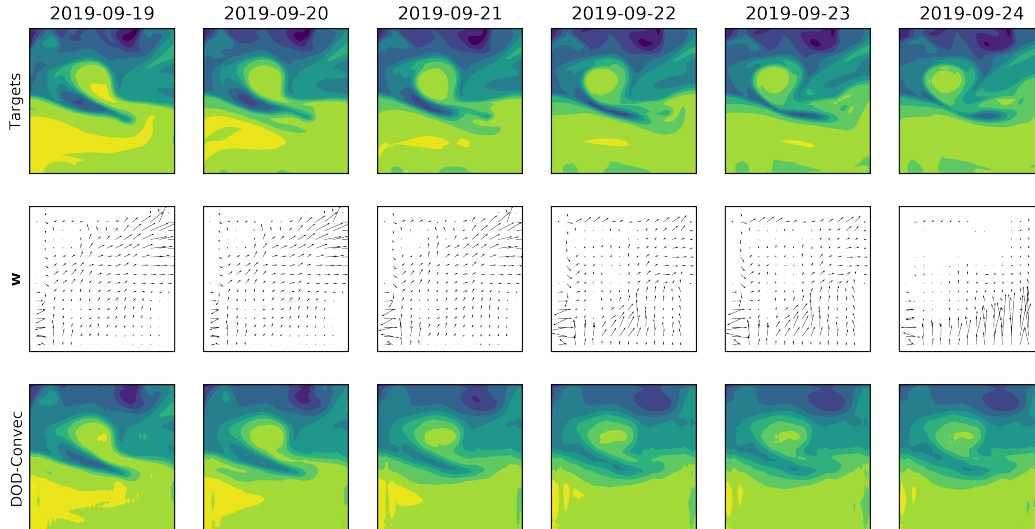
Figure 6: Estimated convection field for our model.

to produce a synthesised estimate of the state of the system using data assimilation methods, as such the data does follow the true temperature, making this an ideal dataset to apply our method to.

Our dataset is formed by first decomposing a larger study region into 64 by 64 pixel image patches, these are displayed in Figure 4. Our regions differ from those used in [8] in that we have selected our grouping to more closely follow the movement of the gulf stream up the Eastern seaboard of North America and then accross the Atlantic, mixing with the colder water from the Arctic regions. This choice allows us to model the most interesting regions, with the complex nonlinear mixing dynamics we are most interested in.

Due to a re-calibration of the GLOBAL_ANALYSIS_FORECAST_PHY_001_024 product we only have access to data from January 1st, 2016 onwards and so cannot use the same temporal range as considered in [8]. Instead we use data from 2016–2018 as training data, and then the data from 2019 to test on. We also partitioned our data into length 4 input sequences, and then predicted on the next 8 sequences. Leading to 11935 training examples and 3984 test examples, we also test our data on the same set of regions as used for training.

Additional figures In Fig. 6 we display the predictions from our model first presented in Fig 5, but now including also the convection field used to advance our model. This vector field displays both long range stream lines, as well as localised regions around which a particle rotates around a center, furthermore the richness of all of these fields was obtained by decoding a 4×16 dimensional vector, and not as an image-to-image transform. This parsimonious deconvolution from the latent space, combined with the ability of our method to learn complex vector fields is the principle component of its demonstrated success on convective fields such as the sea surface temperature data.

B Precipitation nowcasting

For this experiment we consider the problem of providing short term precipitation forecasts from Doppler radar images as considered in [2]. The data comes from the multi-radar multi-sensor (MRMS) system developed by NOAA National Severe Storms Laboratory, and provides precipitation rate updates at 2 minute intervals over a spatial resolution of 1km $\times$ 1km. We use the MRMS dataset [32] re-sampled to 10 minute intervals. The dataset covers the whole of the United States, but we restrict out attention to sixteen 128×128 regions in the Pacific Northwest, and then further restrict our attention to the two wettest regions in the training set. We train on data from January 2018 and then test on the first week of November 2017 giving 8904 training instances and 2568 test instances. Following [2] we frame the problem as a classification problem and discretise the precipitation data using three thresholds in units of millimetres of rain per hour, these ranges are given by $[0, 0.1), [0.1, 1.0), [1.0, 2.5)$ and $[2.5, \infty)$, and the model is then trained using a cross-entropy loss target these pixel values.

The resulting accuracy on the classification problem is displayed in Table 2, which demonstrate that there is little to choose between the two approaches in this instance. It would appear that once the process has been coarsened in this way there is no significant difference between the physics free method and our physics informed method on the precipitation data, the implications of this are the subject of further study, however it is ultimately

reassuring that our method proves no worse on this problem, and indeed significant better on the "full-image" temperature data reported in Sec. 4

Table 2: One hour ahead precipitation nowcasting accuracy.

	Training Accuracy	Test Accuracy
U-Net [12, 2]	0.9353	0.8047
DOD-Convec	0.8445	0.8052