20

Deep Learning of Unresolved Turbulent Ocean Processes in Climate Models

Laure Zanna and Thomas Bolton

20.1 Introduction

Current climate models do not resolve many nonlinear turbulent processes, which occur on scales smaller than 100 km, and are key in setting the large-scale ocean circulation and the transport of heat, carbon and oxygen in the ocean. The spatial-resolution of the ocean component of climate models, in the most recent phases of the Coupled Model Intercomparison Project, CMIP5 and CMIP6, ranges from 0.1° to 1° (Taylor et al. 2012; Eyring et al. 2016b). For example, at such resolution, mesoscale eddies, which have characteristic horizontal scales of 10–100 km, are only partially resolved – or not resolved at all – in most regions of the ocean (Hallberg 2013). While numerical models contribute to our understanding of the future of our climate, they do not fully capture the physical effects of processes such as mesoscale eddies. The lack of a resolved mesoscale eddy field leads to biases in ocean currents (e.g., the Gulf Stream or the Kuroshio Extension), stratification, and ocean heat and carbon uptake (Griffies et al. 2015).

To resolve turbulent processes, we can increase the spatial resolution of climate models. However, we are limited by the computational costs of an increase in resolution (Fox-Kemper et al. 2014). We must instead approximate the effects of turbulent processes, which cannot be resolved in climate models. This problem is known as the parameterization (or closure) problem. For the past several decades, parameterizations have conventionally been derived from semi-empirical physical principles, and when implemented in coarse resolution climate models, they can lead to improvements in the mean state of the climate (Danabasoglu et al. 1994). However, these parameterizations remain imperfect and can lead to large biases in ocean currents, ocean heat and carbon uptake.

The amount – and availability – of data from observations and high-resolution simulations has been increasing. These data contain spatio-temporal information that can complement or surpass our theoretical understanding of the effects of unresolved (subgrid) processes on the large-scale, such as mesoscale eddies. Efficient and accurate deep learning algorithms can now be used to leverage information within this data, exploiting subtle patterns previously inaccessible to former data-driven techniques. The ability of deep learning to extract complex spatio-temporal patterns can be used to improve the parameterizations of subgrid scale processes, and ultimately improve coarse resolution climate models.

Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science, and Geosciences,
First Edition. Edited by Gustau Camps-Valls, Devis Tuia, Xiao Xiang Zhu, and Markus Reichstein.

20.2 The Parameterization Problem

The generic nonlinear equations for the evolution of a variable Y (e.g., velocity component, temperature, and salinity) in the ocean are given by

$$\frac{\partial Y}{\partial t} = -(\mathbf{u} \cdot \nabla)Y + F, \tag{20.1}$$

where $\mathbf{u} = (u, v, w)$ is the three-dimensional velocity (momentum), $\frac{\partial}{\partial t}$ the Eulerian acceleration, $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$ is the 3D gradient operator, and F is a set of forces or sources/sinks of the oceanic quantity Y. If, for example, the variable Y represents ocean momentum, then the force F includes all individual forces, such as those from the pressure, Coriolis effect, viscosity, and all external influences (e.g., wind forcing).

In ocean models, Equation 20.1 are numerically integrated at a finite spatial and temporal resolution. This leads to equations for the resolved fields, denoted by $\overline{()}$, as illustrated below:

$$\frac{\delta \overline{Y}}{\delta t} = -\left(\overline{u}\frac{\delta}{\delta x} + \overline{v}\frac{\delta}{\delta y} + \overline{w}\frac{\delta}{\delta z}\right)\overline{Y} + \overline{F}. \tag{20.2}$$

The discretization, and the truncation at a finite resolution, eliminates processes occurring at scales below the model resolution. Therefore, the ocean model described by Equation 20.2 is not an accurate representation of the true system described by Equation 20.1. To improve the fidelity of the climate model, the effects of unresolved (subgrid) scales on the large-scale flow (i.e., $\overline{Y}, \overline{\mathbf{u}}$) must be approximated – this is the parameterization problem.

The crux of parameterizing subgrid physical processes within models is done by introducing using some function P which only depends on $\overline{Y}$ on the right hand side of Equation 20.2. The challenge lies in constructing $P(\overline{Y})$, which: (i) accurately captures the physical processes being parameterized; (ii) respects physical principles such as conservation laws; (iii)) is numerically stable when implemented into a climate model; and (iv) generalizes to new dynamical regimes. There are two main ways of constructing parameterizations: physics-driven and data-driven.

The physics-driven approach has been the prevailing and conventional approach for the past few decades. This approach starts with some physical principle, mechanism, or theory, related to the process to be parameterized. From this physical knowledge, a bulk formula is derived to approximate the effect of that process on the resolved flow. The main advantage of physics-driven parameterizations is their interpretability. However, physics-driven parameterizations only include the bulk effect of a process and parameters are often not observable and/or not constrained by observations. As a result of these caveats, ocean models, which mainly implement physics-driven parameterizations, still exhibit large biases in both the mean and variance of the climate state. Hence, alternative and complementary approaches are necessary to address these shortcomings. Data-driven approaches can help overcome these issues.

The data-driven approach makes no physical assumptions about the process to be parameterized. Here, data – from high-resolution models, observations, or a combination of both – guides the construction of the parameterization. An algorithm (e.g., simple linear regression, or machine learning techniques) is employed to directly learn the parameterization from the data. In general, the functional form and parameters are empirically estimated from the data. However, the choice of algorithm will implicitly make assumptions regarding the parameterization and its functional form. Some early attempts of data-driven ocean eddy (subgrid) parameterizations showed interesting results in idealized model setups (e.g., Berloff 2005; Mana and Zanna 2014; Zanna et al. 2017). More generally, data-driven modeling of turbulence has advanced in recent years (Duraisamy et al. 2019). The advent of new tools from machine learning can improve the computational efficiency and generalization of data-driven parameterizations. Combined with the increasing wealth of data from observations and high-resolution simulations, it is now possible to begin investigating more thoroughly how data-driven parameterizations can improve the representation of unresolved processes and reduce model biases in long-range climate simulations.

20.3 Deep Learning Parameterizations of Subgrid Ocean Processes

20.3.1 Why DL for Subgrid Parameterizations?

There are a plethora of machine learning algorithms which are well suited for the supervised regression problem needed for parameterization (linear regressions, random forests, support vector machines, etc.). However, one particular form of algorithm has stood out in recent years: neural networks (NNs).

Computational resources have now reached the levels required to train large NNs containing thousands or millions of parameters, across a range of architectures such as deep fully-connected networks, deep belief networks, recurrent neural networks, and convolutional neural networks (CNNs) (Goodfellow et al. 2016). For example, the power and success of CNNs comes from the fact that the convolution layers – which typically extract the most vital information from 2D spatial fields – are learned from data. A disadvantage of convolution layers is their computational cost when forming a prediction during implementation, compared to simpler physics-driven parameterizations - however, parameterizing with CNNs is still computationally cheaper than running high-resolution models.

20.3.2 Recent Advances in DL for Subgrid Parameterizations

This chapter is concerned with ocean subgrid parameterizations, but there have been many developments for the parameterization of atmospheric processes which are described in Chapter 20.

Data-driven parameterizations of turbulence, that use deep learning, initially appeared in direct numerical simulation studies (Tracey et al. 2015; Ling et al. 2016a, b), which concern spatial scales that are orders of magnitude smaller than the spatial scales of climate models. For example, Ling et al. (2016b) used a deep fully-connected NN to learn an eddy momentum parameterization of the anisotropic stress tensor. They introduced a physical constraint into their NN using the Galilean-invariant tensor basis of Pope (1975), $\mathbf{T}^{(n)}$, which ensures the data-driven parameterization has particular symmetries. The predicted momentum parameterization ($\hat{\mathbf{S}}_{\mathbf{u}}$) of Ling et al. (2016b) takes the form

$$\hat{\mathbf{S}}_{\mathbf{u}} = \begin{pmatrix} \hat{S}_x \\ \hat{S}_y \end{pmatrix} = \overline{\nabla} \cdot \left(\sum_n g_n \mathbf{T}^{(n)} \right), \tag{20.3}$$

where the coefficients g_n are predicted by the NN using only Galilean-invariant inputs. The deep NN outperform linear regression models, only after applying these physical constraints. Integrating physical principles into data-driven algorithms is important for fidelity, but can also boost the predictive skill of the resulting parameterization.

Additional studies have used NNs to parameterize eddy momentum fluxes in models of freely-decaying 2D turbulence (Maulik and San 2017; San and Maulik 2018; Maulik et al. 2019; Cruz et al. 2019) or in large-eddy simulations (Zhou et al. 2019). For example, Maulik et al. (2019) used NNs to parameterize eddy vorticity fluxes, which were then implemented back into the same model. They used the conventional Smagorinsky and Leith eddy viscosity functions (Smagorinsky 1963; Leith 1968) as input features to one of the NNs: this did not improve the predictive skill of the NN but did improve the numerical stability of the turbulence model once the NN is implemented. By doing so, they removed upgradient momentum fluxes to stabilize the numerical simulations but therefore altered the physics of turbulence processes. Nonetheless, incorporating physical and mathematical properties into DL algorithms is an important step for making parameterizations physically-consistent and potentially improving their performance when implemented into a climate model.

For parameterizations of ocean turbulence, a handful of studies using CNNs have emerged. Bolton and Zanna (2019) and Zanna and Bolton (2020) used CNNs to parameterize ocean mesoscale eddies in idealized models. They showed that CNNs can be extremely skillful for eddy momentum parameterizations, with predictions that generalize very well to different dynamical regimes (e.g., different ocean conditions and turbulence regimes). Another idealized study by Salehipour and Peltier (2018) showed the potential of CNNs to parameterize ocean vertical mixing rates. The DL algorithm could predict the mixing efficiency well beyond the range of the training data, producing a more universal parameterization compared to previous studies.

20.4 Physics-aware Deep Learning

CNNs, and NNs in general, are good candidates to capture the spatio-temporal variability of the subgrid eddy momentum forcing and potentially other subgrid ocean processes which are not resolved in current climate models. However, one of the main criticism of NNs and DL approaches is that they do not include physical constraints. Physics-based parameterizations, on the other hand, are often developed using physical constraints, together with conservation and symmetry laws. We have presented several examples highlighting the improvements of DL parameterizations when physical constraints are included. The constraints ensure that the parameterizations remain faithful to the physics of the underlying process which we are trying to capture, as well as improving the numerical stability of the global climate model when the new parameterizations are implemented. There are several routes to embed physics-constraints in machine learning-based parameterizations. For example, pre-processing the input data or the post-process the output (Bolton and Zanna 2019) could be used, but will likely introduce some biases. However, there are other avenues that might show more promise.

The first avenue is the most conventional and entails using DL to optimally learn the unknown coefficients used in physics-based parameterizations (Schneider et al. 2017a).

However, the underlying assumptions behind this approach is that the structural form of the parameterization is a correct representation of a given process, and no other parameterizations than the ones already in use are needed. Neither of these assumptions are valid in ocean models (Zanna et al. 2018), since not all parameterizations included in ocean models are correct or encompass all the missing processes.

A second avenue is a change in the loss function used during optimizations. The loss function can be adjusted to include additional constraints, such as global conservation of mass, momentum, salt, or energy. This simple approach helps ensure that the system tends toward such conservation principles (Beucler et al. 2019). However, the conservation laws may not be strictly enforced, only approximately, unless hard constraints are used.

The third avenue is to modify the architecture of the NNs (Ling et al. 2016a; Zanna and Bolton 2020). For example, Zanna and Bolton (2020) used maps of resolved velocity components as input to the CNN in order to predict both components of the subgrid eddy momentum forcing $\hat{S}_x$ and $\hat{S}_y$. To physically-constrain the architecture, they used a specifically-constructed final convolutional layer with fixed parameters (Figure 20.1). The activation maps of the second-to-last convolution layer represent the elements of an eddy stress tensor **T**. The final convolution layer then takes the spatial derivatives of the activation maps of the second-to-last convolutional layer (i.e., the eddy tensor elements) using fixed filters, representing central-difference stencils to form the two outputs $\hat{S}_x$, and $\hat{S}_y$ for eddy momentum forcing. This ensures that the final prediction originates from taking the divergence of a symmetric eddy stress tensor, achieving global momentum and

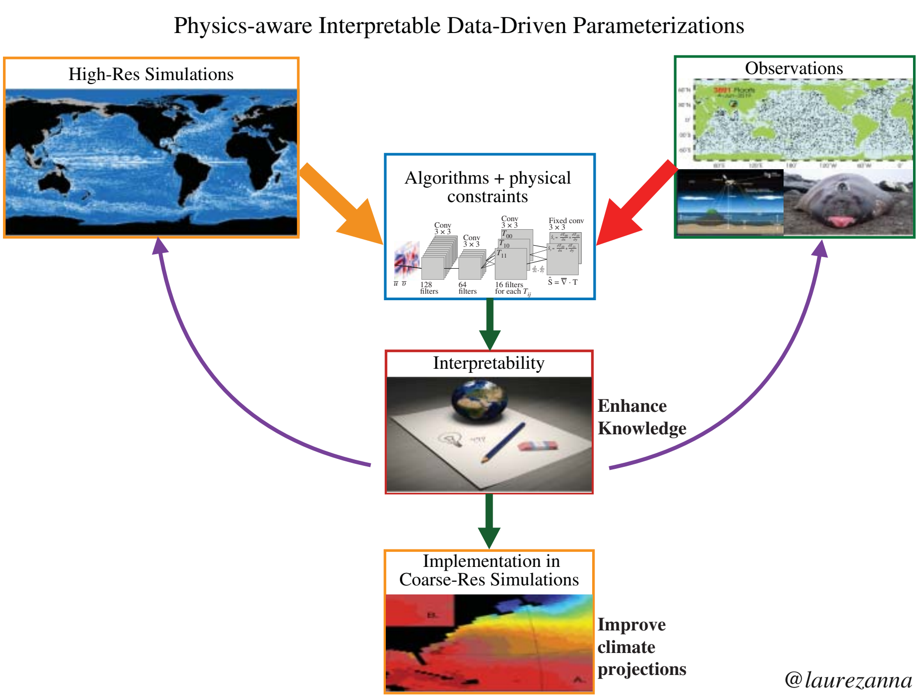

Figure 20.1 Schematic of physics-aware deep learning parameterizations for implementation in coarse-resolution models.

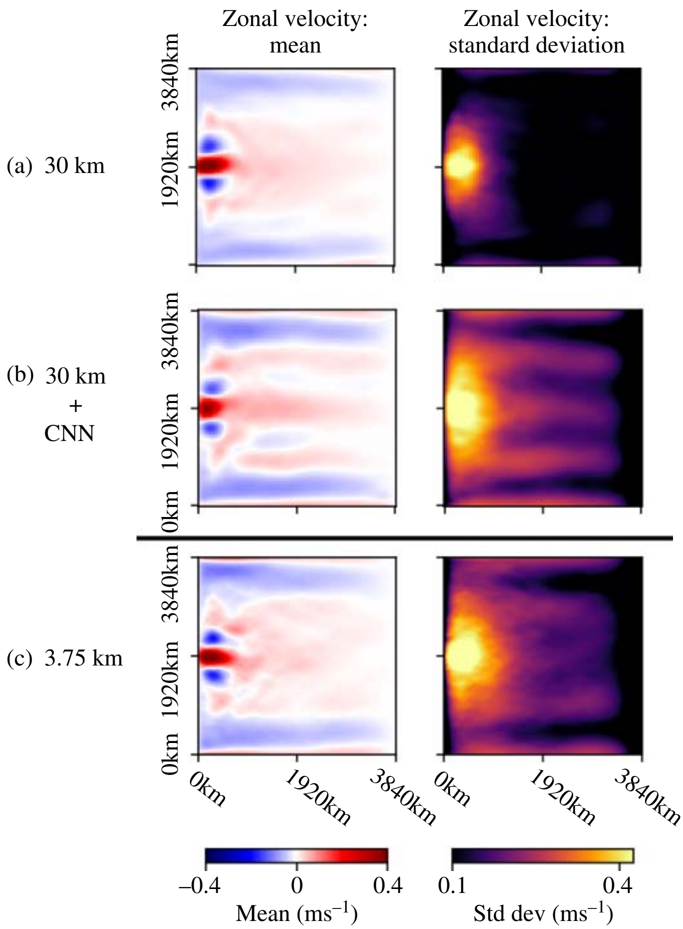

Figure 20.2 Evaluation in an idealized model, zonal velocity (time-mean, left and standard deviation, right): a) Coarse-resolution (30 km), b) Coarse-resolution (30 km) with physics-aware CNN parameterization implemented, c) High-resolution (3.75 km).

vorticity conservation (via the divergence theorem) by prescribing appropriate boundary conditions (the latter might be difficult in climate simulations with complex geometry). This approach, directly integrating physical-principles with data-driven algorithms, leads to more physically-robust ML parameterizations and vastly superior results compared to purely physics-driven parameterizations (Figure 20.2) (Zanna and Bolton 2020). However, it requires the largest knowledge of the practitioner: expertise in both deep learning and physics is necessary.

20.5 Further Challenges ahead for Deep Learning Parameterizations

Parameterizations of unresolved ocean processes will be in demand in climate models for many decades to come. The traditional approach of physics-driven parameterizations, while showing some success, remain sub-optimal as many processes remain poorly represented or are missing from models. Deep learning can help bridge the gap and improve the representation of missing processes using the wealth of new data from high-resolutions simulations and observations, together with physical constraints (as described in section 20.4 and other chapters of this book). There are, however, several challenges ahead in developing physics-aware ML parameterizations, which relate to: how and what to learn from data; how to improve the generalization of ML parameterizations; the interpretability of the resulting algorithm.

Learning from data. Dealing with substantial amounts of data to train ML algorithms remains an obstacle in deriving subgrid parameterizations, but coordinated efforts are well underway to break this barrier, such as the Pangeo project (e.g. Eynard-Bontemps et al. 2019).

However, defining "subgrid" (or unresolved) scales, via an averaging procedure, from either model or observational data is a non-trivial but crucial component of any data-driven parameterization which is often overlooked. The choice of subgrid definition directly impacts what physical processes will be captured by the data-driven parameterization. Gentine et al. (2018) and Rasp et al. (2018) were able to by-pass this problem by using data directly extracted from a 2D high-resolution model embedded into a coarse-resolution climate model; therefore, the "subgrid" scales were available without additional processing. However, this case is an exception. Most other groups tackling ML parameterizations have so far used spatial coarse-graining, which produces a local definition of eddy forcing, on uniform grid (Bolton and Zanna 2019; Zanna and Bolton 2020) or non-spherical geometry (Brenowitz and Bretherton 2018; Yuval and O'Gorman 2020). The choice of averaging procedure has a significant impact on the nature of the resulting subgrid forcing and the separation of scales (as illustrated in Figure 20.3 for the subgrid momentum forcing). The choice of how to separate resolved and unresolved scales can lead to artifacts in the evaluation of nonlinear subgrid forcing (e.g., panel d in which a simple coarse-graining procedure is used), or can produce different patterns and magnitudes of subgrid forcings (e.g., panels d–f, which show the effects of using course-graining, a low-pass filter, or a combination of a low pass filter with coarse-graining). If using a (low-pass) filter, the spatial scale of the filter should also be carefully considered when dealing with spherical coordinates as the subgrid forcing will change in spatial scale as well; e.g., the Rossby deformation scale at which mesoscale eddies are resolved varies with latitudes. Whether these definitions (panels d–f) are truly representative of the missing forcing in a coarse-resolution model remains to be determined.

Generalization of ML parameterizations. Another obstacle to accurate ML parameterizations is their ability to generalize to different regimes or conditions (i.e., to extrapolate outside the range on which they were trained). While CNNs for ocean eddy momentum parameterizations have shown great success in generalizing to different turbulent regimes (Bolton and Zanna 2019), when implemented into ocean models, even if physical constraints imposed, they can lead to unphysical behaviors without ad-hoc tuning (Zanna and Bolton 2020). There are several ways to improve generalizations of ML-parameterizations, which include: (i) learning from a range of high-resolution simulation under different regimes (O'Gorman and Dwyer 2018) and optimally combining the resulting DL parameterizations as suggested by Bolton and Zanna (2019) while imposing physical constraints; (ii) the use of causal inference to target physical relationships in the training data to be used

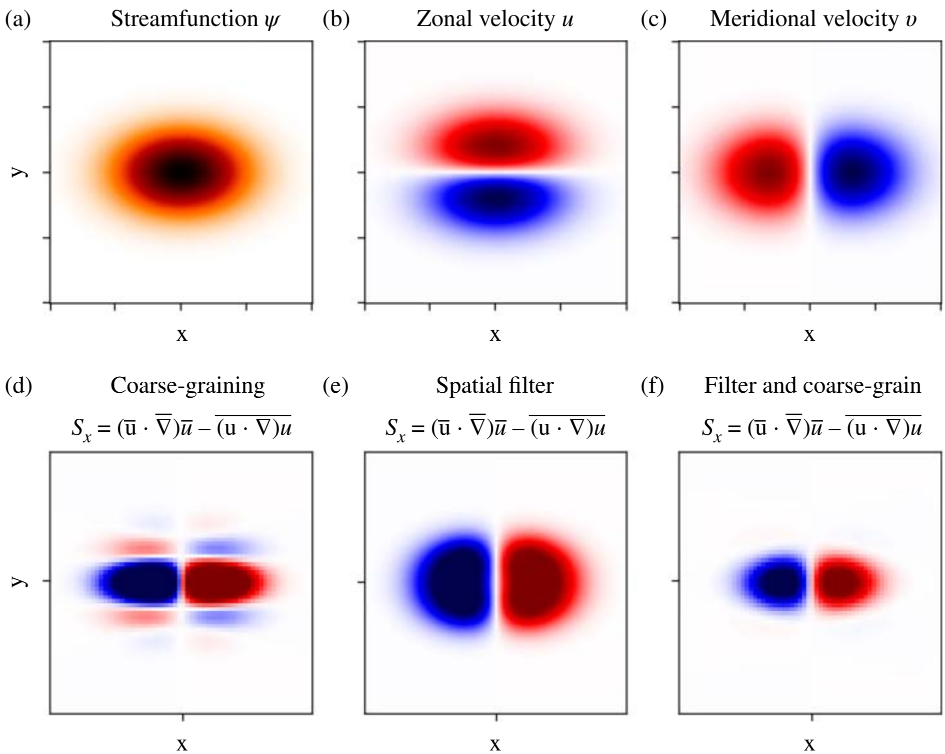

Figure 20.3 Illustrative example considering the effects of averaging procedure on the corresponding zonal eddy momentum forcing (S_x): Assume a Gaussian ellipse streamfunction $\psi \propto e^{-(ax^2+by^2)}$, which emulates a coherent vortex (panel a), and associated velocity components $u = -\frac{\partial \psi}{\partial y}$ (panel b) and $v = \frac{\partial \psi}{\partial x}$ (panel c). Panel (d) shows coarse-graining, Panel (e) a Gaussian spatial filter, and Panel (f) a Gaussian spatial filtering followed by coarse-graining.

as input in DL algorithm; this has the potential to select variables which co-vary according to physical laws and therefore constrain the algorithm to reproduce that relationship, even in unseen conditions. The training data, whether from high-resolution numerical models or observations, possess some biases which may limit the performance or accuracy of the ML parameterizations. A potential way forward is to use transfer learning: one trains ML algorithms with abundant model data and re-tune the ML parameterizations with observations Chattopadhyay et al. (2020), which have less biases. Transfer learning could also potentially improve the generalization of these deep learning models.

Interpretability. Finally, for deep learning models in general, predictive skill is valued above other factors such as interpretability. In general, it is difficult to understand how deep learning methods transform an input into the target variable. The final prediction of a CNN is a culmination of the information extracted from the previous convolutional layers of the network. We can talk broadly about how convolution layers automate feature extraction, and then attempt to dissect the feature maps of the intermediate layers, but identifying exactly what features are being extracted by the many learnt filters can be cumbersome or sometimes completely unfeasible. For example, in Figure 20.4, the first layers extract

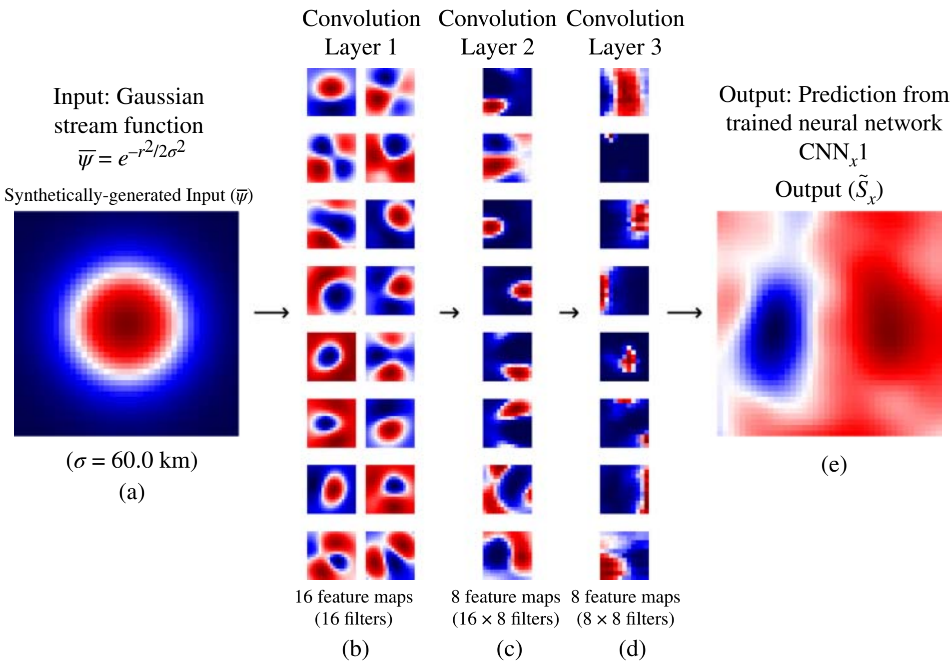

Figure 20.4 Interpretability: Activation maps are the result of the convolution acting on the previous layers output, and then passing it through the activation function. Here, a radially-symmetric Gaussian function to generate an eddy is fed into the already-trained NN for an ocean subgrid parameterization by Bolton and Zanna (2019). The activation maps for each convolutional layers are shown. The activation maps for the first convolution layer are collection of first- and second-order derivatives. Therefore, without a-priori knowledge, the neural network learns to take derivatives of the input streamfunction, which corresponds to velocities and velocity shears. This is a robust feature across all of the NNs trained to predict the eddy momentum forcing.

derivatives but subsequent features are harder to interpret. Interpretability is particularly hard to extract in large networks with 10^5–10^6 parameters. For simpler NNs architecture, different techniques are currently being developed (Toms et al. 2020), and it would be interesting though challenging to applying them to more complex NN architectures. Finally, other machine learning methods, such as data-driven equation-discovery, can lead to interpretable parameterizations of ocean eddy forcing (Zanna and Bolton 2020); this approach aims to construct a closed-form equation from data, harnessing the power of data-driven algorithms while retaining interpretability. Data-driven equation discovery could be used in conjecture with deep learning methods, to extract information from the complex NNs to go beyond the traditional parameterizations already in use.

We are at the very beginning of what deep learning can bring to development of ocean parameterizations, with many challenges ahead, but with the exciting potential to discover new physics, further our understanding and representation of ocean processes, and improve the fidelity and reliability of climate models.