Technical Memo

ECMWF
European Centre for Medium-Range Weather Forecasts

863

Learning earth system models from observations: machine learning or data assimilation?

Alan J. Geer (Research Department)

Preprint of work submitted to Phil. Trans. A

May 2020

Series: ECMWF Technical Memoranda

A full list of ECMWF Publications can be found on our website under:
http://www.ecmwf.int/en/publications

Contact: library@ecmwf.int

Abstract

Recent progress in machine learning (ML) inspires the idea of improving (or learning) earth system models directly from the observations. Earth sciences already use data assimilation (DA), which underpins decades of progress in weather forecasting. DA and ML have many similarities: they are both inverse methods that can be united under a Bayesian (probabilistic) framework. ML could benefit from approaches used in DA, which has evolved to deal with real observations – these are uncertain, sparsely sampled, and only indirectly sensitive to the processes of interest. DA could also become more like ML and start learning improved models of the earth system, using parameter estimation, or by directly incorporating machine-learnable models. DA follows the Bayesian approach more exactly in terms of uncertainty quantification, and retaining existing physical knowledge, which helps to better constrain the learnt aspects of models. This article makes equivalences between DA and ML in the unifying framework of Bayesian networks. These show, for example, that four-dimensional variational (4D-Var) data assimilation is equivalent to a Recurrent Neural Network (RNN). More broadly, Bayesian networks are graphical representations of the knowledge and processes embodied in earth system models. Even if their full Bayesian solution is not computationally feasible, they give a framework for organising modelling components and knowledge, whether coming from physical equations or learnt from observations. These networks can be solved using approximate Bayesian inverse methods (as in variational DA, or backpropagation in ML) and could be used to merge the best of DA and ML. Development of all these approaches could address the grand challenge of making better use of observations to improve physical models of earth system processes.

1 Introduction

Machine learning (ML) has made rapid progress in diverse areas including the classification of images (Krizhevsky et al., 2012; Le, 2013), translation between languages (Sutskever et al., 2014; Wu et al., 2016) and superseding human skill at the game of go (e.g. Silver et al., 2016, 2017). These applications can require neural networks with millions to billions of trainable parameters, large numbers of layers, and specialised architectures, such as convolutional networks. These tools are broadly referred to as 'deep learning' (LeCun et al., 2015) and, along with many other kinds of ML, are now available through easy-to-use open-source software such as SciKit Learn (Pedregosa et al., 2011), Keras (Chollet et al., 2015) and TensorFlow (Abadi et al., 2015). Along with developments in the broader fields of artificial intelligence (AI), computer science, and statistics, this has driven a re-evaluation of the possibilities of ML in the earth sciences (Dueben and Bauer, 2018; Boukabara et al., 2019, 2020). Many proposed applications start from two assumptions: that ML provides an all-purpose non-linear function-fitting capability, 'a universal approximator' (Hornik, 1991), and that ML fits (or 'emulators') will be faster than existing physical modelling approaches. Such a tool could be used almost anywhere in the earth sciences, and many applications have been explored (Chevallier et al., 2000; Krasnopolsky et al., 2005a; McGovern et al., 2017). Further, and a main focus of this article, ML is proposed as a way to better use the vast amounts of observational data available from satellites, in-situ scientific measurements, and in future, internet-of-things (IOT) devices. The goals include making better remote sensing products (Ball et al., 2017) and building models of earth-system processes directly from the data (Schneider et al., 2017; Reichstein et al., 2019).

The earth sciences already have a framework for using observations, which has underpinned three decades of improvements in weather forecasting (Bauer et al., 2015; Eyre et al., 2020) and is known as data assimilation (DA). As an example, in 2012 weather centres were able to give 5 days warning of the landfall of Hurricane Sandy in the vicinity of New York. This would not have happened without millions of observations from weather satellites and the DA framework that was used to ingest this information into the forecasting systems (McNally et al., 2014). Despite different origins and applications,

DA and ML have a lot in common, being able to learn about the world from data, and using 'inverse methods' to do so. There are strong mathematical similarities between the 'variational' form of DA and the way neural networks are trained (Hsieh and Tang, 1998; Abarbanel et al., 2018). In particular, these both use gradient descent techniques, and the adjoint method for calculating gradients in DA (Errico, 1997) is mathematically identical to the standard approach in ML, known as backpropagation. It is not possible to summarise the full complexity of DA or ML here, nor all the mathematical equivalences between them. However, from a broad enough viewpoint, DA and ML are just two flavours of inverse method that can be united under Bayesian statistics.

Known problems in ML include the difficulty of incorporating existing physical knowledge (Von Rueden et al., 2019; Boukabara et al., 2020) and the brittleness of its results, such as image classifications that fail after small changes in object orientation (Alcorn et al., 2019) or the change of a single pixel in an image (Su et al., 2019). Improved handling of uncertainty is seen as a key development for using ML in earth system applications (Boukabara et al., 2020; Reichstein et al., 2019) but this is still under development and focuses on predictive uncertainty (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017). DA as currently applied has robust and well-established ways of handling uncertainty in all parts of the problem. DA can incorporate prior knowledge, which includes both physical laws and the accumulated knowledge built up from past observations. It also has tools for dealing with the complexity of real observations, which are usually sparsely and irregularly distributed, and measured using indirect techniques (Rodgers, 2000; Eyre et al., 1993, 2020). Looking from the shared Bayesian viewpoint, it might be straightforward for ML to adopt some of these tools during its adaptation to earth science applications.

DA could also take on some of the characteristics of ML. It has mainly been employed for state estimation, such as providing the initial conditions for weather forecasts, and generally a perfect forecast model has been assumed. Parameter estimation in DA relaxes this perfect model assumption and allows model parameters to be updated alongside the state (Aksoy et al., 2006; Norris and Da Silva, 2007). However, parameter estimation, and more broadly automated model discovery, has not been done widely in the earth sciences, and whether this is due to difficulty or lack of effort is not clear. In theoretical settings, DA has been used as an alternative ML framework to learn a geophysical model (Bocquet et al., 2019) and hybrid DA-ML approaches seek to incorporate a trainable model, such as a neural network, as a component of a physical model (Tang and Hsieh, 2001) or as a complete replacement for physical models within the main DA process for state estimation and forecasting (Brajard et al., 2020). However it is still to be seen whether these approaches will scale up to real geophysical applications. Finally, even among earth scientists, ML may be better known than DA. This might come from the apparently daunting mathematical framework of DA, the relative lack of training material available on the internet, or the reliance of most weather centres (and other users of DA) on bespoke and often private software systems. DA could learn from ML on this too.

This article will explore the crossover between ML and DA, with particular focus on how earth system models could be learnt directly from observations. Section 2 establishes a Bayesian framework for comparing DA and ML, focusing on uncertainty characterisation. Section 3 extends this to see how DA, and some forms of ML, can use observations to follow a chaotic dynamical system like the earth's atmosphere. A summary of typical physical modelling and observational issues in earth system DA is given in Sec. 4. Based on this, Sec. 5 considers how to learn better earth system models, Sec. 6 looks at the computational tools and Sec. 7 concludes.

2 Uniting ML and DA under a Bayesian framework

Both DA and ML solve an inverse problem, which we can understand by first defining the forward problem, where a function (or 'model') $h()$ maps from a state x to an observations y, and the function has some parameters w:

$$y = h(x, w), \tag{1}$$

Here, y, x and w can be vectors or scalars. The inverse problem (Tarantola, 2005) is to find the state x and/or parameters w from the observations. This becomes difficult when the function is either hard to invert, or there is no unique solution, for example when the same observations can be produced by different combinations of state and parameters. Through this article, we will have two different forward models in mind:

- In ML, inputs x are known as 'features', and outputs y are known as 'labels'. The forward model will typically be a deep neural network, and its parameters (or weights) w are learnt either from a large training dataset made by humans, such as a set of image classifications (Deng et al., 2009), or in an adversarial manner against another ML model (Goodfellow et al., 2014; Silver et al., 2017).

- To obtain initial conditions x for a global weather forecast through DA, observations y are typically combined from a time window around 6 h or 12 h long. Around 10 million observations are used per 6 h period. The state x is valid at the start of the window and a physical model is used to move the atmospheric state to the appropriate time of the observations (a "state model") and to simulate observations from the geophysical state (an "observation model", or "observation operator"). In $y = h(x, w)$ these two are combined, and typically the model is assumed perfect, meaning w is fixed.

In geophysical forecasting, DA is usually cycled through time, but this is covered in Sec. 3 and for the moment we consider a static problem. One major approximation will be made to simplify the discussion: learning of parameters will stand proxy not just for parameter-finding but for function-finding too. Given a broad enough function, such as a set of differential equations of different orders, parameter-finding can in any case also be function-finding (Bocquet et al., 2019).

Given that none of the variables in the inverse problem are fully known, they must be subject to uncertainty, and the most general way to represent uncertainty is through the mathematics of probability. Bayes' theorem solves the inverse problem (also known as 'inference') from a probabilistic point of view (Gelman et al., 2013) and data assimilation and other inverse problems can be derived from it (Lorenc, 1986; Wikle and Berliner, 2007; Stuart, 2010). Probabilistic problems can also be expressed in a graphical form known as Bayesian networks (Needham et al., 2007), which are themselves used for solving machine learning problems (Ghahramani, 2015). As will be seen in this article, these networks provide a convenient graphical way of describing complex procedures like atmospheric DA, but they also provide a general mathematical framework for defining complex inverse problems.

The Bayesian network in Fig. 1a encodes the same problem as Eq. 1 but from the probabilistic viewpoint, with circles (or 'nodes') representing uncertain variables, and arrows (or 'edges') representing causal relations between variables – in the current case, the direction of the forward function. The graph as a whole represents the joint probability distribution of variables, here $P(y, x, w)$. An assumption of independence between x and w will be made initially, but later relaxed in Sec. 3 and in the appendix. The diagram shows the exact symmetry between DA and ML: DA usually holds w constant to estimate x; ML holds x constant to estimate w.

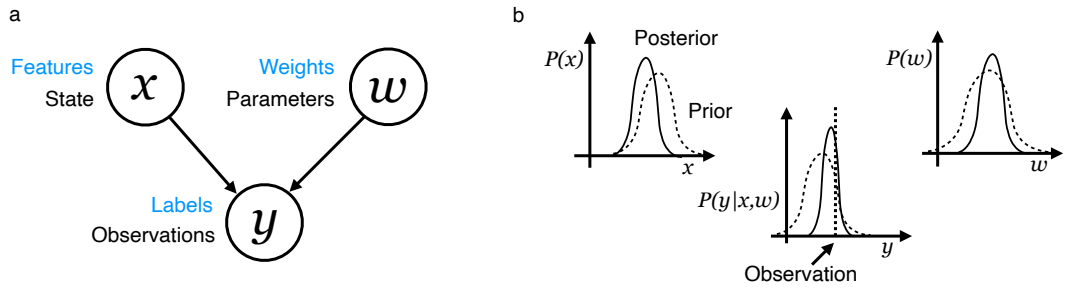

Figure 1: (a) A Bayesian network representing machine learning (legends in blue) or data assimilation (legends in black). Arrows represent dependence between variables (in other words, the direction of the forward model). (b) Probability distributions of the prior (dashed) and posterior (solid, updated from the observation indicated by a vertical dotted line) of these variables, for an illustrative scalar example of the Bayesian network on panel a.

From an ML point of view, it may seem odd that it is the features and weights (x and w) that are exactly equivalent, when typically it is the features and the labels (x and y) that are lumped together as "the data". But often a dependent relation between features and labels is clear: there may be infinitely many images that contain a cat, but an image classification system need only contain one label for "cat". Moreover, the symmetry between features and weights comes from acknowledging that all variables are uncertain: any updates to the weights need to be traded off against the possibility of errors in the features.

To make use of the Bayesian framework, we have to indicate what we already know, our 'prior' knowledge, as a probability distribution, here $P(x)$ and $P(w)$, the prior probabilities of particular features and weights (or states and parameters). Encoding existing knowledge in a probability distribution allows it to be statistically weighed against new knowledge coming from observations, which are themselves uncertain. In DA, we typically start with a detailed deterministic estimate for the the current state of the atmosphere x – for example we already have a weather forecast for today based on past data. There is also a statistical representation of the errors in that forecast, $P(x)$, typically based on a combination of an ensemble of forecasts and climatological statistics (Bonavita et al., 2016).

The final component in a Bayesian framework is the probabilistic version of the forward model, in this case $P(y|x,w)$, the conditional probability of observing y given x and w. This is the probabilistic equivalent of the typical feedforward neural network in ML or the models for the atmosphere and the observation in DA. Given the three probabilistic models, $P(x)$, $P(w)$ and $P(y|x,w)$, we can incorporate observations to find the updated probability of the state and the parameters, known as the posterior probability, $P(x,w|y)$. The chain rule of probability can be used, as in the derivation of Bayes theorem, to find a generalised solution to the DA or ML problem in Fig. 1 (see appendix):

$$P(x,w|y) = \frac{P(y|x,w)P(x)P(w)}{P(y)}. \tag{2}$$

The denominator $P(y)$ is a normalising factor that makes sure the equation produces a valid PDF that integrates to 1, but its calculation can often be avoided (see below, and appendix). Fig. 1b illustrates this with a scalar example: an observation, indicated by the vertical dotted line, helps reduce the posterior uncertainty of the state and parameters. As with the standard version of the Bayes theorem, this 'solves' the inverse problem: we have a way to improve our knowledge of unknown or partly known variables (x and w) by comparing observations y to predictions from a model.

The Bayesian framework can seem abstract, partly because its practical application is harder than it may

appear. A brute force solution would be to explore all combinations of possible x and w, making point evaluations the forward model $P(y|x,w)$ – essentially a parameter search, which as the size of the vectors w and x increases, rapidly becomes impossible through 'combinatorial explosion', also known as the 'curse of dimensionality'. Bayesian techniques took off with methods to more economically sample the search space, such as the Markov Chain Monte Carlo (MCMC) approach (Gelman et al., 2013). However, these approaches are still not feasible for typical DA problems like weather forecasting, due to the cost of running the forward model (if ML emulators could make that faster, then MCMC techniques could be more viable (Cleary et al., 2020)). Another way to make Bayesian techniques more efficient is to assume all uncertainties are described by Gaussian distributions. This approach will get us back to a more recognisable DA or ML formulation.

If we can combine a Gaussian distribution of errors with the forward function $h()$, then we have a way to mathematically evaluate the conditional probability of observing a particular value y, (Rodgers, 2000):

$$P(y|x,w) = \frac{1}{c_1} exp\left(-\frac{1}{2}\frac{(y-h(x,w))^2}{(\sigma^y)^2}\right), \tag{3}$$

The normalising constants of the Gaussian are folded into $c_1 = \sigma^y\sqrt{2\pi}$. Here σ^y is the observation error. It is also possible to include forward modelling error (i.e. error in $h()$) but for simplicity in this article we have assumed that all model errors are explained by errors in the parameters w. For convenience, this and the next two equations have been written with scalar variables, but one of the benefits of this approach is that it can be generalised, in a computationally feasible way, to handle millions of observations and high-dimensional states such as gridded representations of the atmosphere. The full equations can be found in many of the aforementioned citations. To specify Gaussian prior probability distributions of the state and parameters in DA (or features and weights in ML) then we need central starting estimates, x^b and w^b. In DA, x^b is known as the background (for example yesterday's forecast of today's weather). In ML, w^b would be the initial settings for the weights. With estimates of the size of the Gaussian error in each, σ_x and σ_w, and some more normalising constants, we have:

$$P(x) = \frac{1}{c_2} exp\left(-\frac{1}{2}\frac{(x^b-x)^2}{(\sigma^x)^2}\right); P(w) = \frac{1}{c_3} exp\left(-\frac{1}{2}\frac{(w^b-w)^2}{(\sigma^w)^2}\right), \tag{4}$$

Putting these into the Eq. 2 version of Bayes' rule, taking the logarithm of both sides, hiding the normalising constants in c, and multiplying by -1, we get a quadratic cost function that is conventionally denoted $J()$:

$$J(x,w) = -ln(P(x,w|y)) + c = \underbrace{\frac{(y-h(x,w))^2}{(\sigma^y)^2}}_{J^y} + \underbrace{\frac{(x^b-x)^2}{(\sigma^x)^2}}_{J^x} + \underbrace{\frac{(w^b-w)^2}{(\sigma^w)^2}}_{J^w}. \tag{5}$$

The minimum of $J(x,w)$ is the location of the most probable x and w, given the observation y, and it can be found economically using gradient descent methods, just as in the variational form of DA and in forms of ML that use backpropagation, particularly neural networks.

The different terms of the cost function, denoted J^y, J^x and J^w, can be related back to the familiar forms of DA and ML. The observation and state terms J^y and J^x are always present in DA. Here we already have good knowledge of the state of the atmosphere x, based on a short-range ('background') weather forecast. It is important to represent its uncertainty, σ_x, relative to that of the observations, σ_y. Bayes theorem, and hence Eq. 5, gives the tools for making just the right nudge in the direction of the observations to improve on the background forecast and find a posterior estimate of $P(x)$ that is closer to the truth than either the background state or the observations. The relative size of the errors determines how much

weight is given to observations versus prior knowledge, so error diagnosis and modelling is the key to successful DA (Bormann and Bauer, 2010; Bannister, 2008; Bonavita et al., 2016).

In DA the perfect model assumption should allow the final J^w term to be ignored. However, real models and observations are not perfect, and in practice systematic errors are estimated using various flavours of J_w term. One type, known as variational bias correction, estimates a bias model as part of the observation model (Dee, 2004; Eyre, 2016). It is also possible to estimate errors in the state model, an approach that is known as 'weak constraint' in variational DA (Trémolet, 2006; Laloyaux et al., 2020). The closest that routine DA gets to learning models is parameter estimation, where typically a small subset of parameters from the state model are allowed to be estimated alongside the state (Aksoy et al., 2006; Norris and Da Silva, 2007).

Turning to ML, the first term, J^y, is recognisable as the squared loss function that is often used to measure the misfit between the ML-generated label $h(x,w)$ and the 'true' label y. A squared loss term is equivalent to assuming Gaussian errors in the labels and setting all those errors to 1. The second term J^x represents errors in the features, and is not present in ML algorithms to this author's knowledge. Omitting this term is equivalent to assuming a perfect knowledge of the features x. The final term J^w is the weights regularisation term that is often used in ML. The typical squared norm regularisation, w^2, (also known as Tikhonov regularisation, or ridge regression) is therefore equivalent to assuming that all weights have a prior best estimate of 0 ($w^b = 0$) with Gaussian errors of 1. However, typical ML approaches do not use these explicit descriptions of uncertainty in the features, labels or parameters.

Uncertainty representation in ML tends to focus on prediction errors (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Gagne et al., 2019; Sønderby et al., 2020). The problem of erroneous labels is recognised, albeit through ad-hoc regularisation tools such as dropout (Srivastava et al., 2014) or adding network layers to learn label noise (Jindal et al., 2016). The process of data normalisation (putting the features and weights onto a common scale, such as 0 – 1, prior to training) must also implicitly set up a balance of errors in Eq. 5, although whether this is the right balance is not guaranteed. Some error-related balancing must also be achieved through the hyperparameter tuning that is needed to get good results in ML. With the Bayesian perspective, the errors could be specified quantitatively in the loss function (Eq. 5) and more explicitly weighed against errors in the other parts of the problem. An example would be uncertainty in the features, such as the pixel errors that can make image classification fail (Su et al., 2019). Adding a J^x term to the loss function might help address this, at least during the training phase. If the errors were predictable (for example, some images might contain glinting from direct sunlight, some sensors might be subject to higher pixel errors) then different features could be given different weights. This would mirror the way that observation error models in DA are becoming situation-dependent (Geer and Bauer, 2011). These could be more quantitative ways to prevent over-fitting in ML.

Assigning meaningful parameter uncertainty is difficult in ML. In DA we typically have highly informative prior knowledge, and its uncertainty characterisation is critical. In ML there is typically no prior knowledge of the parameters. However, ML parameter errors are sometimes assumed implicitly. In a typical transfer learning task, ML weights are pre-trained in one domain, and then re-trained on a typically more limited set of data in a different domain (Cireşan et al., 2012). If it were possible to specify prior errors for the weights (σ^w), then old (prior) and new information could be objectively weighted in this process. Otherwise, the weighting would depend on ad-hoc decisions such as the relative amount of new training data and how many epochs to use in the training. However, it has been the work of decades to find good ways of diagnosing and representing errors for DA. One technique proposed for ML is to represent the prior error of a neural network as a Gaussian Process (GP) (Neal, 1995) and it is possible to make an exact equivalence between GP and deep neural networks (Lee et al., 2017) but it is hard to find evidence of this being used in practice. It would be attractive to learn these errors as part of the

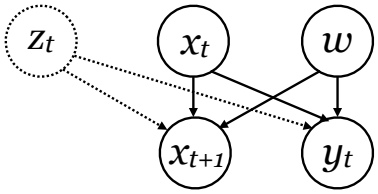

Figure 2: The solid circles and lines show a Bayesian network representing one cycle of a cycled data assimilation system, such as 4D-Var or the Kalman Filter, but with the added facility of parameter or model estimation. The posterior $P(x_{t+1}, w|y_t)$ provides the prior information on the state and parameters, $P(x_t, w)$, for the next cycle (see appendix). Adding the dotted parts gives a Bayesian description of a recurrent neural network (RNN).

process, rather than being forced to specify them – this is implicit in most ML, and an explicit goal in DA research (Satterfield et al., 2018).

3 Cycling through time

Bayes' rule is just one link in a chain of probabilistic information gathering. Figure 1 and Eq. 2 do not stand alone, but instead the posterior probability from one application of the Bayes rule gives the prior probability for the next, allowing new knowledge to be incorporated when it comes available. Joint probability distributions can be factorised, via the chain rule of probability, into a series of conditional distributions. When some variables are conditionally independent of others, it allows the problem to be simplified and broken into a chain of smaller calculations. Bayesian networks, explained in more detail in (Gelman et al., 2013; Needham et al., 2007; Ghahramani, 2015), are a graphical representation and consequence of these rules: probabilistic calculations for one node need only involve the variables on which that node has a direct dependence. In the context of DA, the state model carries information through time, so Bayesian calculations can be cycled forward, updating a modelled representation of the world (a 'digital twin') with new information as it arrives in real time. This approach can be used for controlling machines and robots (linking to developments in control theory and especially the Kalman filter) and it is also how we can track and forecast the state of physical, biological, and earth processes in real time. These techniques are particularly applicable for chaotic dynamical systems, such as the atmosphere, where errors between the forward model and the true state are continually growing, but can be reduced again by a DA cycle using new observations.

Figure 2 incorporates this key additional property of the forward model, its forecast of the future (probabilistic) state of the atmosphere. This is shown by including two time levels for the atmospheric state, x_t and x_{t+1}, with y_t representing all the observations in the time window between them (for example, the 6 h or 12 h window mentioned earlier). We can solve the network in Fig. 2 in two steps, with the appendix giving full details. The first step would update a prior estimate of state and parameter uncertainty $P(x_t, w)$ given the observations y_t, to find $P(x_t, w|y_t)$. This follows the Bayes solution to the network we started with in Fig. 1, but allowing dependence between x and w. From this updated starting point, we can run the same probabilistic model forward in time to perform the 12 h integration to forecast $P(x_{t+1}, w|y_t)$. This then provides the background joint probability of the state and parameters, $P(x_t, w)$ for the next cycle of data assimilation (dropping the known conditional variable y_t from the notation). In the second step it is not strictly necessary to simulate the observations, but for diagnostic reasons it is usually done, so in practice the first and second step use the same model, one that we could write

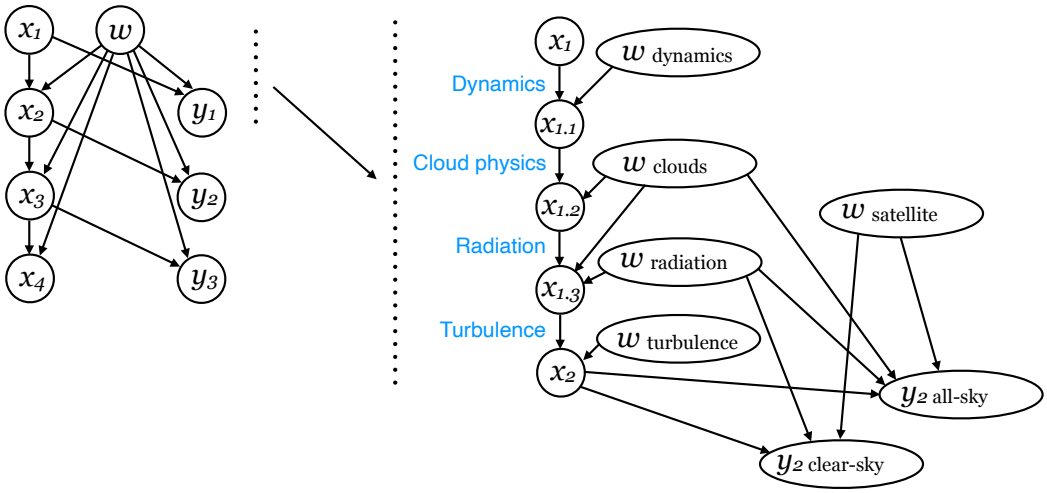

Figure 3: Bayesian networks representing the internals of a hypothetical forward model for atmospheric data assimilation. Left: timesteps; right: inside one timestep.

deterministically as $x_{t+1}, y_t = f(x_t, w)$. This is a reason why this article does not make more distinction between the "state model" and "observation model". The network diagram represents all the main data assimilation approaches including ensemble Kalman filters (Evensen, 2009) and four-dimensional variational data assimilation (4D-Var), which is the basis of the most successful DA algorithms used in weather forecasting (Lorenc, 1986; Rabier et al., 2000; Lorenc and Jardak, 2018).

The time dimension in the DA process can be equated with the layers in a feedforward neural network (Abarbanel et al., 2018) but a clearer parallel is between DA and Recurrent Neural Networks (RNNs). The dotted parts of Fig. 2 make the equivalence. An RNN takes a sequence of inputs, here labelled z_t, and provides a sequence of outputs y_t. The network has learnable parameters w that are the same for every iteration. As well as providing the outputs y_t, the network also provides input to itself for the next iteration, which gives it a memory to store the evolving state x_t. These correspondences make RNNs an obvious candidate for monitoring and forecasting a chaotic dynamical system like the atmosphere, and in providing a complete replacement for DA using ML (Park and Zhu, 1994; Pathak et al., 2018; Vlachas et al., 2020; Sønderby et al., 2020). Making the link back to atmospheric data assimilation, the additional time-dependent inputs z_t actually do exist - this term can describe external forcings or boundary conditions, such as the sea-surface temperature (if the ocean is not coupled into the atmospheric model) or the changing solar flux or carbon dioxide in the atmosphere.

So far our forward model has hidden a lot of detail, such as the layer configurations in a deep neural network or the internals of the physical models in DA. But Bayesian networks can also represent these internal structures and hierarchies as smaller probabilistic problems. For example, geophysical state models will usually propagate the state forward in time through a series of timesteps x_1, x_2 and so on, and Fig. 3 shows a Bayesian network representation of this. In most modern DA algorithms including 4D-Var, the atmospheric state at each timestep is passed through the observation operator to simulate observations relevant to that timestep, y_1, y_2 and so on. To solve this network by Bayesian factorisation, we could start at the end and work back in time, finding $P(x_3, w|y_3)$ in the same way we solved the original network in Fig. 1, using the Bayes rule. To take our information further back in time to get $P(x_2, w|y_2, y_3)$, we solve a network like that in Fig. 2, but for the initial state x_t rather than the final state x_{t+1}. We can continue back in time until we get $P(x_1, w|y_3, y_2, y_1)$. This full Bayesian approach

is hypothetical; in 4D-Var this role is performed by the adjoint technique, equivalently backpropagation in ML. However, both in these networks and their real-world simplified implementations, we can take observational information both forward and backward in time.

4D-Var DA uses backward-in-time propagation of information only within the 12 h observation window, and the overall cycling scheme (Fig. 2) propagates information forward from one cycle to the next. Past observations are an important part of the current knowledge of the atmospheric state, and a typical global forecast is based on information extracted from observations over the last 10 days (Fisher et al., 2005). For ML approaches attempting to replace DA altogether, keeping a representation of the atmospheric state is a way to retain information from past observations, and this suggests RNN-type approaches. To benefit from this information in a non-recursive network without any explicit representation of the state, the features would have to include the last 10 days of observations.

4 Physical processes and observations

Within each timestep, an atmospheric model has components for the dynamics and thermodynamics of the atmosphere, as well as non-resolved processes such as cloud and precipitation (moist physics), radiation, turbulence, and others. To provide a simplified representation, Fig. 3 further breaks down the model timesteps and shows these operators acting sequentially on the state, though real models can have more complex arrangements. As mentioned before, and to make the parallels with ML, in this article the physical knowledge is represented in a simplified way by the parameters w, even where in reality the knowledge may be encoded in equations. The knowledge can be broken down into that needed by different schemes. Some knowledge is used in multiple areas. For example, the microphysical details of cloud and precipitation particles, such as their sizes and shapes, affect how fast the particles fall, the rate at which they evaporate, and many other processes in the cloud physics. Simultaneously, the size and shape of these particles is important in the radiative transfer of the atmosphere. Hence the cloud parameters affect both the cloud physics and radiation steps.

Observations, particularly those made by satellites, also have a complex dependence on the physical state of the atmosphere. Satellite sensors typically measure the intensity of upwelling earth radiation (the radiance) at a particular frequency, relying on the DA process to infer the physical state of the atmosphere or surface (Eyre et al., 2020). Modelling of clear-sky satellite observations ($y_{2\,\text{clear-sky}}$) relies on the same spectroscopic physical parameters (and many of the same equations) used in the model radiation scheme, hence the dependence on $w_{\text{radiation}}$. There are also observation-specific parameters, represented by $w_{\text{satellite}}$. Increasingly satellite radiances are being assimilated in 'all-sky' conditions, i.e. including clear-sky, cloudy and precipitating scenes (Bauer et al., 2010; Geer et al., 2018). These observations are shown by $y_{2\,\text{all-sky}}$, and they are modelled including the radiative effect of cloud and precipitation particles. Hence this relies on physical knowledge shared with the cloud and radiation schemes, w_{clouds} and $w_{\text{radiation}}$. Here, the Bayesian network representation helps clarify issues that are deeply hidden in real atmospheric DA systems; for example the cloud physics, the radiation scheme, and the observation operator may in practice all make different physical assumptions about unrepresented cloud microphysics (Geer et al., 2017).

Figure 3 helps explain another key property of DA (and more generally, Bayesian inverse methods): the ability to infer indirect information from observations. The observations in the first timestep of a DA cycle, y_1, are a special case, being directly dependent on the background state x_1 via only an observation operator. But those in the second timestep, y_2, are dependent on all the dynamical and physical processes in step 1, and we can improve our knowledge of these processes based on later observations. In atmo-

spheric 4D-Var DA, this property is known as the tracer effect because its most obvious initial example was inferring winds from humidity or ozone features in the atmosphere (Andersson et al., 1994; Peubey and McNally, 2009) ('tracer' meaning an atmospheric constituent whose evolution is mainly driven by advection, which is applicable to humidity and ozone under some circumstances). However its impact is far broader, and it is how satellite observations are used to provide information on the positions of atmospheric fronts, to infer winds and other dynamical information, and even to improve the hidden internal details of tropical cyclones (Bauer et al., 2010; Geer et al., 2018).

5 Learning new earth system physics

The cloud physics step in Fig. 3 is a main target for ML or DA approaches seeking to learn new physical models from observations (Schneider et al., 2017; Gentine et al., 2018). Global models for weather and climate work with horizontal grid scales from around 10 km upwards, but the scales of typical cloud features, such as deep convection, can be 1 km or less. The microphysical part of cloud processes – the formation and growth of individual cloud and precipitation particles – involves scales down possibly to the molecular level. Hence, cloud parametrisations must model the average impact of these much smaller scale-processes at the model grid scale. Uncertainty in cloud parametrisations leads to major uncertainty in climate change projections (Zelinka et al., 2020) and must also reduce our ability to predict weather on shorter timescales. Further, as earth system modelling moves further into representing surface and biological processes, where we have even less ability to express our understanding in physically-based forward models, ML is again an attractive approach to improve scientific knowledge (and hence models) using observations (Reichstein et al., 2019).

However, ML approaches will have difficulty finding the necessary observations. Initial ML training of cloud physics or radiation schemes used the inputs and outputs from existing coarse-resolution models (Krasnopolsky et al., 2005a) – for example by extracting the states $x_{1.1}$ and $x_{1.2}$ from the idealised model in Fig. 3 and using them as the features and labels to train an ML emulator. This proves that ML emulators can be incorporated in forecast models, and could help with efficiency savings, but it does not learn new knowledge. It is also possible to train ML emulators using higher resolution models, including cloud resolving models (Rasp et al., 2018; Gentine et al., 2018; Brenowitz and Bretherton, 2018) which could help improve parametrisation quality. However, cloud resolving models are not the truth and still rely on parametrisations of the microphysics, possibly the same as used in the global models. The problem of using real observations is that we do not have regular-gridded vertical profiles of the full state of the atmosphere at the the inputs and outputs of the model timestep (e.g. x_1 and x_2), let alone at the input and output of a single physical process (e.g. $x_{1.1}$ and $x_{1.2}$). To approximate this with in-situ observations, such as colocated radiosondes, aircraft measurements and ground site data, it might take an entire field campaign to gather a handful of input-output pairs for training.

For global coverage and generalisation, we are reliant on satellite observations. However, these are sensitive to broad-layer quantities and a huge range of atmospheric states can all generate the same observation. This is what makes satellite retrieval and DA a classic inverse problem (Eyre et al., 1993; Rodgers, 2000; Eyre et al., 2020). Detailed forecasting of the atmosphere needs a higher vertical resolution than is routinely observable, so DA fills in the high-resolution details (the "null space" of the observations) using the atmospheric model to infer information indirectly from other observations (via forward and backward in time propagation of observational information). A further issue for variables that vary on fine scales, such as clouds or earth surface properties, is that the location represented by the observation can be very different from the grid-box average that a model parametrisation seeks to represent. DA gives the tools for representing this uncertainty as part of the observation error budget

where it is known as representation error (Janjić et al., 2018). Especially for cloud and precipitation, representation error can be a dominant part of the observational error budget (Geer and Bauer, 2011).

A second main issue for ML is how to incorporate existing scientific ('domain') knowledge. A popular proposal is to put physical constraints as additional terms in the loss function (Beucler et al., 2019; Wu et al., 2020) and other approaches exist (Von Rueden et al., 2019). However a Bayesian approach could start from the existing physical knowledge – in Fig. 1 this would be encoded in prior estimates of the parameters w (and in practice in the physical equations these represent), with $P(w)$ describing the level of confidence in these existing models. Parameterised processes result from a mixture of processes and equations, some that are well known, some that are much less so. This would motivate the use of a more fine-grained network structure in the learning, retaining important well-known equations wherever it makes sense, in order to better constrain the unknown parts of the problem. An example from radiative transfer would be to retain the physical solution of the radiative transfer equations, which is fast, physically justified, and accurate. The question whether ML approaches should be used for components inside physical representations, or whether ML should entirely replace a parametrisation, has been under debate since the early days (Chevallier, 2005; Krasnopolsky et al., 2005b).

For a weather forecasting centre that is already operating a data assimilation system that encodes decades of specialist weather forecasting knowledge, the obvious approach would be to extend the existing DA system to learn aspects of the model, whether this is by doing parameter estimation, or by learning new functional representations using neural networks or equivalent approaches (Bocquet et al., 2019). Fig. 3 helps describe some of the benefits of learning within an existing data assimilation system. For example the cloud parameters, w_{clouds}, are surrounded by constraints – implicit training data – that will help reduce their uncertainty. The cloud physics step is the most obvious: the uncertainty of the states $x_{1.1}$ and $x_{1.2}$ is constrained by the better-known physics that surrounds them (such as the atmospheric dynamics) as well as by all the data that is assimilated into the weather forecasting system, whether or not it is directly sensitive to clouds. As explained earlier, in a cycling DA system, the uncertainty in these states is reduced by by forward propagation of information from observations over the past 10 days (for the atmosphere) as well as by backward-in-time propagation of information from observations that are in the future from the perspective of the model timestep. Further information on w_{clouds} is provided by the dependence of the radiation scheme on these parameters, so that states $x_{1.2}$ and $x_{1.3}$ are also constraints on our knowledge of the clouds. Finally, the cloud-sensitive observations, $y_{1\text{clouds}}$, are also directly sensitive to cloud parameters. DA helps provide not just the obvious information at the inputs and outputs of the model component being learnt, but it combines all possible observational information that is relevant, both in space and across time. Further, if the parameters or models that we are learning are constant over the days and years, these models or parameters would learn from thousands of cycles of training data, at every one of millions of grid points in the model.

Incorporating model learning into the wider DA system has several other potential benefits. Modelling systems contain compensating errors coming from the different parts of the model. It might not be possible to improve on a cloud physics scheme that produced an excessive warming, if it were already compensated by a radiation scheme that generated excessive cooling. A data assimilation system could allow many parameters to be adjusted simultaneously, within their estimated uncertainties, and it could use other observational or physical constraints to help resolve any ambiguities. A parametrisation that was in constant training inside a DA system could adapt to large, otherwise unmodelled changes in the earth system, such as a volcanic eruption, or long-term trends in air pollution (which could for example affect the microstructure of clouds). Even without explicit parameter training, DA systems implicitly incorporate knowledge from observations to compensate for modelling errors. One example comes from long-term climate reanalyses: even if their models use a static level of CO_2 in the atmosphere, they can

still exhibit realistic global warming trends imposed by the observations themselves. This is because DA directly warms the atmosphere, taking the place of the missing physics (Cai and Kalnay, 2005).

However, DA systems need to be careful about attributing sources of systematic error. Sometimes these come from biases in the observations themselves (Eyre, 2016). We may suspect inadequate knowledge of the fundamental physics behind the models, but this is hard to determine when observation calibrations and biases are themselves uncertain (Brogniez et al., 2016). It will be an ongoing challenge to quantify systematic uncertainty in both the existing physical knowledge and in the observations (Carminati et al., 2019) to support physically-based model and parameter learning, to avoid attributing errors to the wrong source.

Model learning (or parameter estimation) has not been widely done in DA for the earth sciences. It is not clear if this is because it is too hard or because not enough effort has been invested – certainly it has not been a priority at weather forecasting centres. Possible problems include the difficulty of simultaneously estimating multiple parameters, the presence of strong nonlinearities and state dependencies in the parameter response, and whether the available observations are even capable of constraining the parameters (if not, this is known as non-identifiability) (Aksoy et al., 2006; Posselt and Vukicevic, 2010; Posselt, 2016). However, for cloud physics, hopes are rising for a future 'microphysical closure' (Geer et al., 2017) when cloud-sensitive observations are assimilated from all-sky satellite radiances, cloud and precipitation radars and lidars, lighting imagers, and in-situ instrumentation, all combined with the physical constraints from the rest of the forecast model and the non-cloud observations in the system. There is similar scope for DA systems to use satellite data to help learn models in areas like sea-ice and land surface processes, where current scientific knowledge is limited, or where parameters vary at fine scales, such as variations in land cover.

A halfway strategy is also possible, taking inspiration from the use of ML in postprocessing forecasts (McGovern et al., 2017), where ML is a kind of situation-dependent bias correction for a partly-erroneous forecast made by a physical model. Weak-constraint DA (Trémolet, 2006) is similar, in that it does not improve the forward model, but estimates a spatial field of model errors. ML could be equally applicable to learning this kind of model error (Bonavita and Laloyaux, 2020). However, in weak-constraint DA it can be hard to separate these errors from errors in the state, if they occur on similar spatial scales (Laloyaux et al., 2020). Hence learning the actual models (or their parameters) would still be preferred as it can focus the learning more precisely where it is needed, truly learning new knowledge from observations.

6 Computational aspects

The computational forms of current DA and ML approaches are shaped by available resources, and by their applications. Broadly, earth science applications of DA tend to use supercomputers, which are required for the forward modelling of the atmosphere and ocean, as well as the background error modelling, both of which have non-local dependencies (and hence a lot of inter-process communication, which is optimised on supercomputers). ML approaches typically use cloud computing, taking advantage of algorithms that require less communication such as stochastic gradient descent (SGD) (Kingma and Ba, 2014), along with the compatibility of neural network processing with graphics or tensor processing units (GPUs or TPUs). One practical problem is to combine typical DA or science applications in Fortran with ML software such as Keras and TensorFlow, which are a Python frontend and a C++ backend (Ott et al., 2020).

But many aspects of ML and DA computing are similar, including the use of adjoint methods or back-

propagation to compute the gradients of the cost/loss function. There is a link to Bayesian networks here too: the factorisation process described earlier requires these networks to be directed acyclic graphs (DAGs). On the ML side, DAGs are also the basis of TensorFlow, which represents each neural network layer as a node and the communications between them as edges. On the DA side, we could follow the Bayesian hierarchy down to the individual line of code, the atomic level of the Bayesian factorisation. DA is deterministic, not probabilistic, on this level, but by line-by-line code differentiation (differentiable programming) it is possible to form the adjoint and TL models that propagate knowledge and uncertainty from one line to the next. TensorFlow allows layers with free-form algebraic computations, and can automatically differentiate these layers to allow them to be part of the backpropagation process. It is possible to automatically differentiate C++ and other code (Hogan, 2014) but most code for DA applications is hand-differentiated – separate lines of code (or separate subroutines) are typed in to represent the nonlinear (direct) computation, for the TL, and for the adjoint. This is a conscious choice, based on the difficulty of using automatic differentiation tools, and the possibility to hand-optimise and hand-regularise the code – for the purposes of differentiation, nonlinear or non-differentiable steps can be modelled by smooth functions (Janisková and Lopez, 2013). One potential application for an ML emulator of an existing physical parametrisation is to use its backpropagation gradients as a replacement for an adjoint model, and hence to use ML as another type of automatic differentiation for existing code.

As mentioned in the introduction, most operational DA codes are based on proprietary software, but there are attempts to provide more re-usable and open-source DA code (English et al., 2017) and `https://www.image.ucar.edu/DAReS/DART/` and `https://www.jcsda.org/jcsda-project-jedi`. An attractive future possibility would be to use the principles of Bayesian networks and a DAG implementation like in TensorFlow, to organise earth system models as networks of much smaller modular components. Existing Fortran subroutines would be packaged up, along with their pre-differentiated counterparts (TL and adjoint models) and a description of their uncertainties, representing prior knowledge to be incorporated in the wider network. In this framework it would be easy to mix neural networks and physical algorithms, making it easier to do DA and ML with them – and hence to learn new knowledge from observations. For ML applications already based around software like TensorFlow, it may also be possible to implement DA algorithms using the tools already available (such as free-form algebraic layers and new terms in the loss function) and to incorporate external physical models (such as observation operators) as additional layers.

Part of the drive towards using ML to learn from earth-system observations is the assumption that ML emulators will be faster than physical models of the observations (Boukabara et al., 2019). However, the runtime cost of observations in one 4D-Var DA system is just 2%, with the main cost coming from the DA algorithm and from the physical models of the earth system (English et al., 2020). Observation computations are easy to parallelise, since the simulation of each observation from the model fields can be done independently after an initial interpolation and communication step. This also suggests that the most promising venue for incorporating ML techniques in earth sciences lies in the physical models themselves.

7 Conclusion

The earth sciences are trying to improve model representations of difficult areas such as clouds and precipitation, and to include new modelling areas, such as earth surface and biological processes. All these areas are difficult to represent on the grid scale of current global models, making it hard to use physical laws to describe these processes, and in some cases the physical laws are not known. The development of existing models through human effort seems to become ever trickier – for example,

despite many important subsequent refinements, the core physical parametrisations in many models are around 30 years old (Tiedtke, 1989, 1993). Parametrisations can themselves be just compact functional summaries of limited observational data – for example widely used assumptions on snow fallspeeds and shapes are based on measurements made in the Cascade mountains during the winters of 1972 and 1973 (Locatelli and Hobbs, 1974). For all these reasons it is attractive to start learning model improvements directly from the vast numbers of observations that have been made over decades, and that are continuing to be made.

Any plan to learn directly from observations will have to deal with the same challenges that have driven the development of DA in the earth sciences – among them the need to represent uncertainty in a quantitative way, and the need to make use of observations that are sparsely distributed in time and space, indirectly related to the processes of interest, and giving incomplete information. Further, although it is tempting to discard existing physical knowledge and start from scratch with ML (Dueben and Bauer, 2018; Pathak et al., 2018; Sønderby et al., 2020), any Bayes-respecting approach would try to benefit from that prior knowledge. In areas where there has been a lot of investment in DA frameworks, model learning is likely to be most achievable within the existing framework, whether as classic DA parameter estimation or ML-like learning of functional forms. However, parameter estimation has made little progress so far in these areas and the feeling is that it will be very difficult. Nevertheless, given the potential benefits, and whatever the methods, the automatic learning of models from observations must be one of the grand challenges of earth sciences.

There are many ways ML could benefit from approaches used in DA, particularly where ML aims to replace the DA process:

- Real observations are sparse and irregular, and indirectly and ambiguously dependent on the geophysical state. DA uses physical observation operators in the forward model and Bayesian inverse methods to extract optimal information on the geophysical state. To replicate this, ML could incorporate physical observation models as an output layer.
- Where our physical knowledge is good, physical models allow us to constrain parts of the problem and localise uncertainties to their correct sources. Hence, physical model layers could also be incorporated in neural networks for constraining the geophysical state.
- The Bayesian framework requires physical quantification of the uncertainty in the observations, the state and the model. The equivalent feature, label, and parameter uncertainty could be similarly quantified in the ML loss function, and might replace ad-hoc regularisation approaches like data normalisation.
- High quality earth system forecasts aggregate global observational information from at least the past 10 days (and longer in the ocean). ML approaches must aim to use all this data.
- The Bayesian equivalence between cycling DA and Recursive Neural Networks (RNN) has been established; this shows that the main job of a recursive network in geophysical state forecasting is to propagate this state forward in time, as a way to retain information from past observations.

If these approaches were followed, ML could end up looking a lot like DA.

Conversely, ML shows DA the huge possibilities for automatically improving or learning models in areas where human-driven approaches are struggling. DA has relied on a perfect model assumption that is increasingly untenable. A fully generalised Bayesian approach to both observational and model uncertainty is just as needed in DA. There are also useful ways that ML emulator models could be incorporated in all

parts of the DA process: as learnable modules in an otherwise physical model framework; for learning model and observation systematic errors; as accelerators for making parts of the process faster (particularly in areas where models need to be run many times, which occurs in both ensemble and variational approaches); and as an alternative tool for automatic differentiation in variational DA. Further, DA needs improve its software tools and make them more generally available, re-usable and documented, in the way that ML already has done. Approaches developed for ML, such as stochastic gradient descent, could also be adopted.

In this article, Bayesian networks were used to graphically describe the processes of DA and ML at a general level, as well as to provide a unifying mathematical basis for comparing them. Although a full Bayesian network solution would be unfeasible, they are the general solution to which practical techniques approximate. Bayesian networks are a directed acyclic graph (DAG), which also provides a way for earth system models to be broken into modules, and combined with ML models in a way that permits diverse learning methods such as DA and ML. With the assumption of Gaussian errors leading to the variational DA framework, these networks can be implemented and solved using differentiable programming, in other words backpropagation and adjoint techniques. The graph framework, similar to what is implemented in TensorFlow, could thus provide an overarching infrastructure for continually updating our knowledge and our estimates of its uncertainty. Ultimately this is a vision that combines both machine learning and data assimilation.

Acknowledgements

The internal reviewers, Niels Bormann, Nils Wedi, Peter Dueben, Massimo Bonvita and Stephen English, are thanked for their invaluable help. These ideas have been shaped through discussions with many people including the internal reviewers and Richard Forbes, Elias Hólm, Patricia de Rosnay, Marcin Chrust, Peter Lean, Peter Bauer, Philippe Lopez and Katrin Lonitz.

A Mathematical notes

The chain rule of probability allows the factorisation of a joint probability distribution in terms of conditional probabilities. For example the joint probability distribution of observations y, state x and parameters w is $P(y,x,w)$ and can be factorised in six ways including:

$$P(y,x,w) = P(x|w,y)P(w|y)P(y); \tag{6}$$
$$P(y,x,w) = P(y|x,w)P(x|w)P(w). \tag{7}$$

By equating the two right hand sides:

$$P(x|w,y)P(w|y) = \frac{P(y|x,w)P(x|w)P(w)}{P(y)} \tag{8}$$

The left hand side gives the conditional joint probability distribution that can be rewritten $P(x,w|y)$. In the initial example of the Bayesian network in Fig. 1, x and w are independent (before the observation of y) so $P(x|w) = P(x)$ and this leads to the simplified form given as Eq. 2, which was used to emphasise the symmetry between parameter and state estimation. However we also now have a form that can be used repeatedly to update our knowledge of the joint PDF of the state and the parameters, as new sets

of observations come in. This is done by noting that from the chain rule the joint PDF of x and w, $P(x,w) = P(x|w)P(w)$:

$$P(x,w|y) = \frac{P(y|x,w)P(x,w)}{P(y)} \tag{9}$$

The posterior on the left hand side then gives the prior for the application of Bayes rule to a new set of observations. For convenience of notation, going forward we can drop the conditionality on the older observations.

The term $P(y)$ is not needed when we assume Gaussian errors and start to solve Bayesian problems variationally as in Eqs. 3 – 5. However it can be calculated if needed using the sum rule of probability, so that $P(y) = \int_x \int_w P(y,x,w)\,dw\,dx$, in a process referred to as marginalisation, in this case over w and x. Marginalisation is a key part of the process in solving Bayesian networks (Needham et al., 2007; Ghahramani, 2015) but for high-dimensional x and w in typical earth science problems, integration over the whole state and parameter domain is unfeasible.

Now we need to work out how to go forward in time and to cycle 4D-Var, as represented in the Bayesian network in Fig. 2 (ignoring z_t). This figure describes a joint PDF $P(y_t, x_{t+1}, x_t, w)$ where we now have two time-levels of the state. We can factorise this is in two helpful ways, bearing in mind how 4D-Var is solved, first as in Eq. 9 estimating the updated state and if necessary parameters from the observations, then running the forward model that gives x_{t+1}:

$$P(y_t, x_{t+1}, x_t, w) = \underbrace{P(x_{t+1}|w,x_t,y_t)P(w|x_t,y_t)P(x_t|y_t)}_{P(x_{t+1},w,x_t|y_t)}P(y_t); \tag{10}$$

$$P(y_t, x_{t+1}, x_t, w) = P(x_{t+1}|w,x_t,y_t)\underbrace{P(x_t|w,y_t)P(w|y_t)}_{P(x_t,w|y_t)}P(y_t). \tag{11}$$

The Bayesian network represents that $x_t + 1$ is conditionally independent on all variables other than its parents, $P(x_{t+1}|w,x_t,y_t)$ is identical to $P(x_{t+1}|w,x_t)$. We expect this from a (probabilistic) dynamical model of the atmosphere: the forward evolution of the system depends only on the initial state x_t and the parameters w, in other words we assume the atmosphere has the first-order Markov property. We now have a general description of the DA problem with, from the right, an update of the state and parameters from the observations, $P(x_t, w|y_t)$, solved by Eq. 9, and then the probabilistic forward model $P(x_{t+1}|w,x_t)$:

$$P(x_{t+1}, w, x_t|y_t) = P(x_{t+1}|w,x_t)P(x_t,w|y_t) \tag{12}$$

A problem is that the prior probability distribution needed by the next cycle of DA is $P(x_{t+1}, w|y_t)$ and we have an extra "nuisance variable" x_t, which we know imperfectly. Hence we would need to integrate (marginalise) over x_t:

$$P(x_{t+1}, w|y_t) = \int P(x_{t+1}, w, x_t|y_t)\,dx_t = \int P(x_{t+1}|w,x_t)P(x_t,w|y_t)\,dx_t \tag{13}$$

In practically feasible versions of DA, for example in Ensemble Kalman filters (Evensen, 2009) and in hybrid 4D-Var (Bonavita et al., 2016) this step is represented by running an ensemble of deterministic forward models, from which the parameters of the prior PDF are estimated.

References

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser,

M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abarbanel, H. D., P. J. Rozdeba, and S. Shirman (2018). Machine learning: deepest learning as statistical data assimilation problems. *Neural Computation 30*(8), 2025–2055.

Aksoy, A., F. Zhang, and J. W. Nielsen-Gammon (2006). Ensemble-based simultaneous state and parameter estimation in a two-dimensional sea-breeze model. *Mon. Weath. Rev. 134*(10), 2951–2970.

Alcorn, M. A., Q. Li, Z. Gong, C. Wang, L. Mai, W.-S. Ku, and A. Nguyen (2019). Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4845–4854.

Andersson, E., J. Pailleux, J. N. Thépaut, J. R. Eyre, A. P. McNally, G. A. Kelly, and P. Courtier (1994). Use of cloud-cleared radiances in three/four-dimensional variational data assimilation. *Quart. J. Roy. Meteorol. Soc. 120*, 627–653.

Ball, J. E., D. T. Anderson, and C. S. Chan (2017). Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of Applied Remote Sensing 11*(4), 042609.

Bannister, R. N. (2008). A review of forecast error covariance statistics in atmospheric variational data assimilation. I: Characteristics and measurements of forecast error covariances. *Quart. J. Roy. Meteorol. Soc. 134*(637), 1951–1970.

Bauer, P., A. J. Geer, P. Lopez, and D. Salmond (2010). Direct 4D-Var assimilation of all-sky radiances: Part I. Implementation. *Quart. J. Roy. Meteorol. Soc. 136*, 1868–1885.

Bauer, P., A. Thorpe, and G. Brunet (2015). The quiet revolution of numerical weather prediction. *Nature 525*(7567), 47–55.

Beucler, T., M. Pritchard, S. Rasp, P. Gentine, J. Ott, and P. Baldi (2019). Enforcing analytic constraints in neural-networks emulating physical systems. *arXiv preprint arXiv:1909.00912*.

Bocquet, M., J. Brajard, A. Carrassi, and L. Bertino (2019). Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models. *Nonlinear Processes in Geophysics 26*(3), 143–162.

Bonavita, M., L. Isaksen, E. Hólm, and M. Fisher (2016). The evolution of the ECMWF hybrid data assimilation system. *Quart. J. Roy. Meteorol. Soc. 142*, 287–303.

Bonavita, M. and P. Laloyaux (2020). Machine learning for model error inference and correction. *J. App. Meterol. Earth. Sys.*, to be submitted.

Bormann, N. and P. Bauer (2010). Estimates of spatial and interchannel observation-error characteristics for current sounder radiances for numerical weather prediction. I: Methods and application to ATOVS data. *Quart. J. Roy. Meteorol. Soc. 136*, 1036–1050.

Boukabara, S.-A., V. Krasnopolsky, J. Q. Stewart, E. S. Maddy, N. Shahroudi, and R. N. Hoffman (2019). Leveraging modern artificial intelligence for remote sensing and NWP: Benefits and challenges. *Bull. Am. Meteorol. Soc. 100*(12), ES473–ES491.

Boukabara, S.-A., V. Krasnopolsky, J. Q. Stewart, A. McGovern, D. Hall, J. E. T. Hoeve, J. Hickey, H.-L. A. Huang, J. K. Williams, K. Ide, P. Tissot, S. E. Haupt, E. Kearns, K. S. Casey, N. Oza, P. Dolan, P. Childs, S. G. Penny, A. J. Geer, E. Maddy, and R. N. Hoffman (2020). Outlook for exploiting artificial intelligence in earth science. *Bull. Am. Meteorol. Soc.*, submitted.

Brajard, J., A. Carassi, M. Bocquet, and L. Bertino (2020). Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the Lorenz 96 model. *arXiv preprint arXiv:2001.01520*.

Brenowitz, N. D. and C. S. Bretherton (2018). Prognostic validation of a neural network unified physics parameterization. *Geophys. Res. Let. 45*(12), 6289–6298.

Brogniez, H., S. English, J.-F. Mahfouf, A. Behrendt, W. Berg, S. Boukabara, S. A. Buehler, P. Chambon, A. Gambacorta, A. Geer, W. Ingram, E. R. Kursinski, M. Matricardi, T. A. Odintsova, V. H. Payne, P. W. Thorne, M. Y. Tretyakov, and J. Wang (2016). A review of sources of systematic errors and uncertainties in observations and simulations at 183 GHz. *Atmos. Meas. Tech. 9*, 2207–2221.

Cai, M. and E. Kalnay (2005). Can reanalysis have anthropogenic climate trends without model forcing? *J. Clim. 18*(11), 1844–1849.

Carminati, F., S. Migliorini, B. Ingleby, W. Bell, H. Lawrence, S. Newman, J. Hocking, and A. Smith (2019). Using reference radiosondes to characterise NWP model uncertainty for improved satellite calibration and validation. *Atmos. Meas. Tech. 12*(1), 83.

Chevallier, F. (2005). Comments on New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of longwave radiation in a climate model. *Mon. Weath. Rev. 133*(12), 3721–3723.

Chevallier, F., J.-J. Morcrette, F. Chéruy, and N. Scott (2000). Use of a neural-network-based long-wave radiative-transfer scheme in the ECMWF atmospheric model. *Quart. J. Roy. Meteorol. Soc. 126*(563), 761–776.

Chollet, F. et al. (2015). Keras. https://keras.io.

Cireşan, D. C., U. Meier, and J. Schmidhuber (2012). Transfer learning for Latin and Chinese characters with deep neural networks. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6. IEEE.

Cleary, E., A. Garbuno-Inigo, S. Lan, T. Schneider, and A. M. Stuart (2020). Calibrate, Emulate, Sample. *arXiv preprint arXiv:2001.03689*.

Dee, D. (2004). Variational bias correction of radiance data in the ECMWF system. In *ECMWF workshop proceedings: Assimilation of high spectral resolution sounders in NWP, 28 June – 1 July, 2004*, pp. 97–112. Eur. Cent. for Med. Range Weather Forecasts, Reading, UK, available from http://www.ecmwf.int.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee.

Dueben, P. D. and P. Bauer (2018). Challenges and design choices for global weather and climate models based on machine learning. *Geosci. Mod. Dev. 11*(10), 3999–4009.

English, S., P. Lean, and A. Geer (2020). How radiative transfer models can support the future needs of earth-system forecasting and re-analysis. *J. Quant. Spectrosc. Rad. Trans.*, accepted.

English, S., D. Salmond, M. Chrust, O. Marsden, A. Geer, E. Hólm, S. Massart, M. Hamrud, R. Stappers, and R. E. Khatib (2017). Progress with running IFS 4D-Var under OOPS. *ECMWF Newsletter 153*, 13–14.

Errico, R. M. (1997). What is an adjoint model? *Bulletin of the American Meteorological Society 78*(11), 2577–2592.

Evensen, G. (2009). The ensemble Kalman filter for combined state and parameter estimation. *IEEE Control Systems Magazine 29*(3), 83–104.

Eyre, J. (2016). Observation bias correction schemes in data assimilation systems: A theoretical study of some of their properties. *Quart. J. Roy. Meteorol. Soc. 142*(699), 2284–2291.

Eyre, J. R., S. J. English, and M. Forsythe (2020). Assimilation of satellite data in numerical weather prediction. Part I: The early years. *Quart. J. Roy. Meteorol. Soc. 146*(726), 49–68.

Eyre, J. R., G. A. Kelly, A. P. McNally, E. Andersson, and A. Persson (1993). Assimilation of TOVS radiance information through one-dimensional variational analysis. *Quart. J. Roy. Meteorol. Soc. 119*, 1427–1463.

Fisher, M., M. Leutbecher, and G. Kelly (2005). On the equivalence between Kalman smoothing and weak-constraint four-dimensional variational data assimilation. *Quart. J. Roy. Meteorol. Soc. 131*(613), 3235–3246.

Gagne, I., D. John, H. M. Christensen, A. C. Subramanian, and A. H. Monahan (2019). Machine learning for stochastic parameterization: Generative adversarial networks in the Lorenz'96 model. *arXiv preprint arXiv:1909.04711*.

Gal, Y. and Z. Ghahramani (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International conference on machine learning*, pp. 1050–1059.

Geer, A., M. Ahlgrimm, P. Bechtold, M. Bonavita, N. Bormann, S. English, M. Fielding, R. Forbes, E. H. Robin Hogan, M. Janisková, K. Lonitz, P. Lopez, M. Matricardi, I. Sandu, and P. Weston (2017). Assimilating observations sensitive to cloud and precipitation. Tech. Memo. 815, ECMWF, Reading, UK.

Geer, A. J. and P. Bauer (2011). Observation errors in all-sky data assimilation. *Quart. J. Roy. Meteorol. Soc. 137*, 2024–2037.

Geer, A. J., K. Lonitz, P. Weston, M. Kazumori, K. Okamoto, Y. Zhu, E. H. Liu, A. Collard, W. Bell, S. Migliorini, P. Chambon, N. Fourrié, M.-J. Kim, C. Köpken-Watts, and C. Schraff (2018). All-sky satellite data assimilation at operational weather forecasting centres. *Quart. J. Roy. Meteorol. Soc. 144*, 1191–1217.

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin (2013). *Bayesian data analysis*. CRC press.

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

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature 521*(7553), 452–459.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680.

Hogan, R. J. (2014). Fast reverse-mode automatic differentiation using expression templates in C++. *ACM Transactions on Mathematical Software (TOMS) 40*(4), 1–16.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks 4*(2), 251–257.

Hsieh, W. W. and B. Tang (1998). Applying neural network models to prediction and data analysis in meteorology and oceanography. *Bulletin of the American Meteorological Society 79*(9), 1855–1870.

Janisková, M. and P. Lopez (2013). Linearized physics for data assimilation at ECMWF. In *S.K. Park and L. Xu (Eds), Data Assimilation for Atmospheric, Ocean and Hydrological Applications (Vol II), Springer-Verlag Berline Heidelberg*, pp. 251–286, doi:10.1007/978–3–642–35088–7–11.

Janjić, T., N. Bormann, M. Bocquet, J. Carton, S. Cohn, S. Dance, S. Losa, N. Nichols, R. Potthast, J. Waller, and P. Wseton (2018). On the representation error in data assimilation. *Quart. J. Roy. Meteorol. Soc. 144*, 1257–1278.

Jindal, I., M. Nokleby, and X. Chen (2016). Learning deep networks from noisy labels with dropout regularization. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 967–972. IEEE.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krasnopolsky, V. M., M. S. Fox-Rabinovitz, and D. V. Chalikov (2005a). New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of longwave radiation in a climate model. *Monthly Weather Review 133*(5), 1370–1383.

Krasnopolsky, V. M., M. S. Fox-Rabinovitz, and D. V. Chalikov (2005b). Reply. *Mon. Weath. Rev. 133*(12), 3724–3728.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105.

Lakshminarayanan, B., A. Pritzel, and C. Blundell (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pp. 6402–6413.

Laloyaux, P., M. Bonavita, M. Dahoui, J. Farnan, S. Healy, E. Hólm, and S. Lang (2020). Towards an unbiased stratospheric analysis. *Quart. J. Roy. Meteorol. Soc.*.

Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 8595–8598. IEEE.

LeCun, Y., Y. Bengio, and G. Hinton (2015). Deep learning. *Nature 521*(7553), 436–444.

Lee, J., Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein (2017). Deep neural networks as Gaussian processes. *arXiv preprint arXiv:1711.00165*.

Locatelli, J. D. and P. V. Hobbs (1974). Fall speeds and masses of solid precipitation particles. *J. Geophys. Res. 79*, 2185–2197.

Lorenc, A. C. (1986). Analysis methods for numerical weather prediction. *Quart. J. Roy. Meteorol. Soc. 112*(474), 1177–1194.

Lorenc, A. C. and M. Jardak (2018). A comparison of hybrid variational data assimilation methods for global NWP. *Quart. J. Roy. Meteorol. Soc. 144*(717), 2748–2760.

McGovern, A., K. L. Elmore, D. J. Gagne, S. E. Haupt, C. D. Karstens, R. Lagerquist, T. Smith, and J. K. Williams (2017). Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bulletin of the American Meteorological Society 98*(10), 2073–2090.

McNally, T., M. Bonavita, and J.-N. Thépaut (2014). The role of satellite data in the forecasting of Hurricane Sandy. *Mon. Weath. Rev. 142*(2), 634–646.

Neal, R. M. (1995). *Bayesian Learning for Neural Networks*. Ph. D. thesis, University of Toronto.

Needham, C. J., J. R. Bradford, A. J. Bulpitt, and D. R. Westhead (2007). A primer on learning in Bayesian networks for computational biology. *PLoS computational biology 3*(8).

Norris, P. M. and A. M. Da Silva (2007). Assimilation of satellite cloud data into the GMAO finite-volume data assimilation system using a parameter estimation method. Part I: Motivation and algorithm description. *J. Atmos. Sci. 64*(11), 3880–3895.

Ott, J., M. Pritchard, N. Best, E. Linstead, M. Curcic, and P. Baldi (2020). A Fortran-Keras deep learning bridge for scientific computing. *arXiv preprint arXiv:2004.10652*.

Park, D. C. and Y. Zhu (1994). Bilinear recurrent neural network. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, Volume 3, pp. 1459–1464. IEEE.

Pathak, J., B. Hunt, M. Girvan, Z. Lu, and E. Ott (2018). Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Let. 120*(2), 024102.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Peubey, C. and A. P. McNally (2009). Characterization of the impact of geostationary clear-sky radiances on wind analyses in a 4D-Var context. *Quart. J. Roy. Meteorol. Soc. 135*, 1863 – 1876.

Posselt, D. J. (2016). A Bayesian examination of deep convective squall-line sensitivity to changes in cloud microphysical parameters. *J. Atmos. Sci. 73*(2), 637–665.

Posselt, D. J. and T. Vukicevic (2010). Robust characterization of model physics uncertainty for simulations of deep moist convection. *Mon. Weath. Rev. 138*(5), 1513–1535.

Rabier, F., H. Järvinen, E. Klinker, J.-F. Mahfouf, and A. Simmons (2000). The ECMWF operational implementation of four-dimensional variational assimilation. I: Experimental results with simplified physics. *Quart. J. Roy. Meteorol. Soc. 126*, 1148–1170.

Rasp, S., M. S. Pritchard, and P. Gentine (2018). Deep learning to represent subgrid processes in climate models. *Proc. Nat. Acad. Sci. 115*(39), 9684–9689.

Reichstein, M., G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, et al. (2019). Deep learning and process understanding for data-driven Earth system science. *Nature 566*(7743), 195–204.

Rodgers, C. D. (2000). *Inverse methods for atmospheric sounding: Theory and Practice*. Singapore: World Scientific.

Satterfield, E. A., D. Hodyss, D. D. Kuhl, and C. H. Bishop (2018). Observation-informed generalized hybrid error covariance models. *Mon. Weath. Rev. 146*(11), 3605–3622.

Schneider, T., S. Lan, A. Stuart, and J. Teixeira (2017). Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophys. Res. Let. 44*(24), 12–396.

Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature 529*(7587), 484.

Silver, D., J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. (2017). Mastering the game of go without human knowledge. *Nature 550*(7676), 354–359.

Sønderby, C. K., L. Espeholt, J. Heek, M. Dehghani, A. Oliver, T. Salimans, S. Agrawal, J. Hickey, and N. Kalchbrenner (2020). MetNet: A neural weather model for precipitation forecasting. *arXiv preprint arXiv:2003.12140*.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research 15*(1), 1929–1958.

Stuart, A. M. (2010). Inverse problems: a Bayesian perspective. *Acta numerica 19*, 451–559.

Su, J., D. V. Vargas, and K. Sakurai (2019). One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation 23*(5), 828–841.

Sutskever, I., O. Vinyals, and Q. V. Le (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112.

Tang, Y. and W. W. Hsieh (2001). Coupling neural networks to incomplete dynamical systems via variational data assimilation. *Mon. Weath. Rev. 129*(4), 818–834.

Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*, Volume 89. SIAM.

Tiedtke, M. (1989). A comprehensive mass flux scheme for cumulus parameterization in large-scale models. *Monthly Weather Review 117*(8), 1779–1800.

Tiedtke, M. (1993). Representation of clouds in large-scale models. *Mon. Wea. Rev. 128*, 1070–1088.

Trémolet, Y. (2006). Accounting for an imperfect model in 4D-Var. *Quart. J. Roy. Meteorol. Soc. 132*, 2483–2504.

Vlachas, P., J. Pathak, B. Hunt, T. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos (2020). Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks 126*, 191–217.

Von Rueden, L., S. Mayer, J. Garcke, C. Bauckhage, and J. Schuecker (2019). Informed machine learning–towards a taxonomy of explicit integration of knowledge into machine learning. *Learning 18*, 19–20.

Wikle, C. K. and L. M. Berliner (2007). A Bayesian tutorial for data assimilation. *Physica D: Nonlin. Phenom. 230*(1-2), 1–16.

Wu, J.-L., K. Kashinath, A. Albert, D. Chirila, Prabhat, and H. Xiao (2020). Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. *Journal of Computational Physics 406*, 109209.

Wu, Y., M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zelinka, M. D., T. A. Myers, D. T. McCoy, S. Po-Chedley, P. M. Caldwell, P. Ceppi, S. A. Klein, and K. E. Taylor (2020). Causes of higher climate sensitivity in CMIP6 models. *Geophysical Research Letters 47*(1), e2019GL085782.