Efficient high-dimensional variational data assimilation with machine-learned reduced-order models

Romit Maulik[1], **Vishwas Rao**[1], **Jiali Wang**[1], **Gianmarco Mengaldo**[2], **Emil Constantinescu**[1], **Bethany Lusch**[1], **Prasanna Balaprakash**[1], **Ian Foster**[1], **and Rao Kotamarthi**[1]

[1]240, Argonne National Laboratory, Lemont, IL 60439, USA
[2]Department of Mechanical Engineering, National University of Singapore, Block EA, #07-08,
9 Engineering Drive 1, Singapore

Correspondence: Romit Maulik (rmaulik@anl.gov)

Abstract. Data assimilation (DA) in geophysical sciences remains the cornerstone of robust forecasts from numerical models. Indeed, DA plays a crucial role in the quality of numerical weather prediction and is a crucial building block that has allowed dramatic improvements in weather forecasting over the past few decades. DA is commonly framed in a variational setting, where one solves an optimization problem within a Bayesian formulation using raw model forecasts as a prior and observations as likelihood. This leads to a DA objective function that needs to be minimized, where the decision variables are the initial conditions specified to the model. In traditional DA, the forward model is numerically and computationally expensive. Here we replace the forward model with a low-dimensional, data-driven, and differentiable emulator. Consequently, gradients of our DA objective function with respect to the decision variables are obtained rapidly via automatic differentiation. We demonstrate our approach by performing an emulator-assisted DA forecast of geopotential height. Our results indicate that emulator-assisted DA is faster than traditional equation-based DA forecasts by 4 orders of magnitude, allowing computations to be performed on a workstation rather than a dedicated high-performance computer. In addition, we describe accuracy benefits of emulator-assisted DA when compared to simply using the emulator for forecasting (i.e., without DA). Our overall formulation is denoted AIEADA (Artificial Intelligence Emulator-Assisted Data Assimilation).

1 Introduction

A physical system can be characterized by our existing knowledge of the system plus a set of observations. Existing knowledge of the system is typically formulated in terms of a mathematical model (hereafter, also referred to as a *model* or *computational model*) that usually consists of differential equations. Observations can arise from various sensors, both remote (e.g., satellites) and in place (e.g., weather stations and radiosondes).

Data assimilation (DA) combines existing knowledge of a system, usually in the form of a model, with observations to infer the best estimate of the system state at a given time. Both existing knowledge and observations come with errors that lead to uncertainties about the "true" state of the system being investigated. Hence, when combining model results characterizing our knowledge of the system with observations, it is essential to account for these errors and give an appropriate weight to each source of information available. This leads to statistical approaches, which are the basis for state-of-the-art DA methods. Model results and observations, along with their uncertainties, are encapsulated within a Bayesian framework to provide the best estimate of the state of the system conditioned on the model and observation uncertainties (or error statistics) (Daley, 1993; Kalnay, 2003; Le Dimet and Talagrand, 1986). More specifically, DA can be formulated as a class of inverse problems that combines information from (1) an *uncertain prior* (also referred to as *background*) that encapsulates our best estimate of the system at a given time, (2) an *imperfect computational model* describing the system, and (3) *noisy observations*. These three

sources of information are combined together to construct a posterior probability distribution that is regarded as the best estimate of the system state at the given time(s) of interest and is referred to as *analysis*. The analysis can be used for various tasks, including optimal-state identification, and the selection of appropriate initial conditions for computational models.

Two approaches to DA have gained widespread popularity: variational and ensemble-based estimation methods (Kalnay, 2003). The former are derived from optimal-control theory, while the latter are rooted in statistics. Variational methods formulate DA as a constrained nonlinear optimization problem. Here, the state of the system is adjusted to minimize the discrepancy between the prior or existing knowledge (e.g., in the form of a computational model) and observations, where their associated error statistics are commonly prescribed. Ensemble-based methods use optimal statistical-interpolation approaches, and error statistics for the prior and observations are obtained from an ensemble. Regardless of the approach adopted, DA can be performed in two ways: sequential and continuous. In the sequential way, observations are assimilated in batches as they become available. In the continuous way, one defines a prescribed time window, called an *assimilation window*, and uses all the observations available within the window to obtain the analysis.

DA, originally developed for numerical weather prediction (NWP), can be traced back to the original work of Lewis Fry Richardson (Lynch, 2008). Today, DA is used extensively in NWP to compute accurate states of the atmosphere that, in turn, are used to estimate appropriate initial conditions for NWP models, to compute reanalysis, and to help better understand properties of the atmosphere. We focus here on NWP applications, although the novel methodology proposed can be applied in other contexts.

In NWP, variational approaches are the workhorse and have been employed for the past several years by leading operational weather centers, including the European Centre for Medium-Range Weather Forecasts (ECMWF) and National Centers for Environmental Prediction (NCEP). The two main methods adopted are three-dimensional variational DA (or 3D-Var) and four-dimensional variational DA (or 4D-Var). In both cases, one defines an assimilation window and seeks the system state that best fits the data available, comprising the prior and observations. In 3D-Var, all observations are regarded as if they were obtained at the same time snapshot (i.e., there is no use of the time dimension of the assimilation window). In 4D-Var, the observations retain their time information, and one seeks to identify the state evolution (also referred to as the trajectory) that best fits them within the assimilation window. The state evolution is commonly obtained via a computational model, consisting of a set of high-dimensional partial differential equations (PDEs), which is often considered perfect (without errors) within the assimilation window.

The current state of the art is 4D-Var. This is often implemented as a strong-constraint algorithm (SC4D-Var) where one assumes that the observations over the time window are consistent (within a margin of observation errors) with the model if initialized by the true state, where the model is considered to be perfect. In reality, the error in the model is often non-negligible, in which case the SC4D-Var scheme produces an analysis that is inconsistent with the observations. The effect of model error is even more pronounced when the assimilation window is large. Weak-constraint 4D-Var (WC4D-Var) (Trémolet, 2006, 2007) relaxes the "perfect-model" assumption and assumes that model error is present at each time snapshot of the assimilation window. This requires estimates of model-error statistics that are commonly simplistic (Brajard et al., 2020). Yet, recent efforts have focused on improving estimates for model-error statistics and on better understanding their impact on the analysis accuracy in both variational and statistical approaches (Akella and Navon, 2009; Cardinali et al., 2014; Hansen, 2002; Rao and Sandu, 2015; Trémolet, 2006, 2007; Zupanski and Zupanski, 2006). Indeed, there has been a push to aggregating computational NWP model uncertainties, such as those due to incomplete knowledge of the physics associated with sub-grid modeling, errors in boundary conditions, accumulation of numerical errors, and inaccurate parametrization of key physical processes, into a component called *model error* (Glimm et al., 2004; Orrell et al., 2001; Palmer et al., 2005).

It has been shown that 4D-Var systematically outperforms 3D-Var (Lorenc and Rawlins, 2005), and for this reason it is today the state-of-the-art DA for NWP applications. However, the better accuracy of 4D-Var comes with the price of higher computational costs. Indeed, to exploit the time dimension of the assimilation window, it is necessary to repeatedly solve both the computational model forward in time and the tangent-linear and adjoint problems (Errico, 1997; Errico et al., 1993; Errico and Raeder, 1999). These two additional steps are particularly expensive and lead to a significantly larger computational cost for the 4D-Var algorithm compared to its 3D-Var counterpart. Hence, a significant fraction of the computational cost in NWP is due to DA. This cost can be equivalent to the cost of 30–100 model forecasts, which corresponds to the number of iterations in the optimization procedure. This high cost restricts the amount of data that can be assimilated, and thus only a small fraction of the available observations are typically employed for operational forecasts (Bauer et al., 2015; Gustafsson et al., 2018).

Our goal in this work is to alleviate the computational burden associated with the 4D-Var approach by replacing the expensive computational model and its adjoint with a data-driven emulator. (We use the terms emulator and surrogate interchangeably in this paper.) Because our emulator is easily differentiable, we can use automatic differentiation (AD) to avoid solving the expensive adjoint problem. AD, in contrast to numerical differentiation, does not introduce any discretization errors such as those encountered in finite differ-

ences. The lack of discretization-based gradient computation also leads to the accurate computation of higher-order derivatives where such errors are more pronounced. Moreover, when gradients are needed with respect to many inputs (such as for partial differentiation), AD is more computationally efficient. Unlike numerical and symbolic differentiation, AD relies on the chain rule to decompose differentials and compute them efficiently. Therefore, they are instrumental in computing gradients with respect to inputs or parameters in neural network applications where the chain rule may be implemented trivially. Therefore, much of the computational burden associated with the 4D-Var method is alleviated by replacing the expensive computational models consisting of high-dimensional PDEs with machine-learning-based (ML) surrogates.

A few efforts toward the integration of DA and ML have been recently undertaken. In Brajard et al. (2021), rather than replacing an entire physics model with an ML emulator, DA is applied to an imperfect physics model, and ML is used to predict the model error. In Frerix et al. (2021), ML is used to improve DA by learning a mapping from observational data to physical states. Le Guen and Thome (2020) propose an ML approach to video prediction that includes a novel neural network cell inspired by DA with component called a "correction Kalman gain". In Hatfield et al. (2021), the original physical parametrization scheme is not replaced by an emulator, but neural networks are used to derive tangent-linear and adjoint models to be used during 4D-Var. Mack et al. (2020) present a new formulation for 3D-Var DA that uses convolutional autoencoders to create a reduced space in which to perform DA. In contrast to our work, they do not create an emulator to step forward in time, as they are performing 3D-Var DA, not 4D-Var. In Brajard et al. (2020), the authors use a method that iterates between training an ML surrogate model for a Lorenz system and applying DA. The output analysis then becomes the new training data to further improve the surrogate. In Penny et al. (2021), they train ML emulators for Lorenz models using a form of recurrent neural networks (RNNs) based on reservoir computing. Then DA is applied (4D-Var and the ensemble transform Kalman filter), estimating the forecast error covariance matrix using their RNN and deriving the corresponding tangent-linear model and its adjoint as linear operators. In Casas et al. (2020), a recurrent neural network is trained to predict the difference between model outputs and a priori performed DA computations during forecasting in a low-dimensional subspace spanned by truncated principal components. Similar methods may also be found in Pawar et al. (2020), Pawar and San (2021), and Popov and Sandu (2021), where ML surrogates are used in lieu of the expensive forward model in ensemble techniques. A recent work (Chennault et al., 2021) incorporates both forward and adjoint information to build the neural surrogates.

Our study is unique in several manners. First, we demonstrate the use of ML emulators for improving variational DA through enabling an acceleration of the outer-loop optimization problem. By using a differentiable ML surrogate instead of an expensive numerical solver, rapid computation of gradients via automatic differentiation allows for a speedup of several orders of magnitude. Moreover, in contrast with Penny et al. (2021), where a reservoir computer was used as the surrogate, our ML emulator is given by a deep recurrent neural network (i.e., a long short-term memory neural network with several stacked cells) for which automatic differentiation is imperative. Our formulation also employs a model-order reduction methodology to forecast dynamics on a reduced space, thereby leading to significant computational gains even for forecasting very high-dimensional systems. In contrast with the formulation in Casas et al. (2020), we perform DA "on the fly" during forecasting instead of learning the mismatch between model predictions and DA-corrected values. This makes our forecasts more generalizable during testing conditions. The specific contributions of this study are summarized as follows.

- We propose a differentiable reduced-order surrogate model using dimensionality reduction coupled with a data-driven time-series forecasting technique.

- We construct a reduced-order variational DA optimization problem that updates the initial condition of any forecast, given observations from random sensors.

- We accelerate this DA by several orders of magnitude by using gradients of the differentiable low-order surrogate model. Our a posteriori assessments show that the reduced-order DA significantly improves the accuracy of the forecast using the updated initial conditions.

We highlight that in contrast to a vast majority of previous DA and ML studies, the proposed technique is demonstrated for a forecasting problem that is of real-world importance (geopotential height). Our overall formulation, AIEADA (Artificial Intelligence Emulator-Assisted Data Assimilation) will comprise a code base that will ultimately develop emulators and data assimilation for climate/weather models.

2 Surrogate modeling

In this section, we will first introduce our surrogate model strategy, which may be used for direct forecasting of a geophysical quantity from data. Subsequently, we introduce the DA procedure for real-time forecasting correction using this surrogate model. The surrogate model relies on a dimensionality reduction given by proper orthogonal decomposition and neural network time-series forecasting of the reduced representation. We review dimensionality reduction, time-series forecasting, and surrogate-based DA in the following.

2.1 Proper orthogonal decomposition (POD)

The first step in the surrogate construction is to reduce the degrees of freedom of the original dataset to enable rapid forecasts. Projection-based reduced-order models (ROMs) are effective at compressing high-dimensional dynamical systems without the loss of spatiotemporal structure. The compression is performed by projecting the high-dimensional model onto a set of optimally chosen basis functions with respect to the L_2 norm (Berkooz et al., 1993). This process can be illustrated for a state variable $x \in \mathbb{R}^N$, where N represents the size of the computational grid. Here, we use a projection to approximate x as the linear expansion on a finite number of k orthonormal basis vectors $\phi_i \in \mathbb{R}^N$, a subset of the *proper orthogonal decomposition (POD) basis*:

$$x \approx \sum_{i=1}^{k} r_i \phi_i, \tag{1}$$

where $r_i \in \mathbb{R}$ is the ith component of the $r \in \mathbb{R}^k$ values, which are the coefficients of the basis expansion. The $\{\phi_i\}$, $i = 1, \ldots, K$, $\phi_i \in \mathbb{R}^N$ values are the POD *modes*. POD modes in Eq. (1) can be shown to be the left singular vectors of the snapshot matrix (obtained by stacking M snapshots of x) of $\mathbf{X} = [x_1, \ldots, x_M]$, extracted by performing a compact singular value decomposition (SVD) on $\mathbf{X}$ (Holmes et al., 2012; Chatterjee, 2000). That is,

$$\mathbf{X} \underset{\text{svd}}{=} \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^\top, \tag{2}$$

where $\mathbf{U} \in \mathbb{R}^{N \times M}$ is the left singular vector matrix and $\boldsymbol{\Phi}_k$ values are the first K columns of the right singular matrix $\mathbf{V}$ (obtained after truncating the last $M - K$ columns based on the relative magnitudes of the cumulative sum of their singular values; Maulik and Mengaldo, 2021). Note that due to the compact nature of the SVD, $\mathbf{U}$ and $\mathbf{V}$ are semi-unitary matrices which need not be square. The singular values of the SVD are available through the elements of the diagonal matrix $\boldsymbol{\Sigma}$. The total L_2 error in approximating the snapshots via the truncated POD basis is then

$$\sum_{j=1}^{M} \left\| x_j - (\boldsymbol{\Phi}_k \boldsymbol{\Phi}_k^\top) x_j \right\|_2^2 = \sum_{i=k+1}^{M} \sigma_i^2, \tag{3}$$

where σ_i is the singular value corresponding to the ith column of $\mathbf{V}$ and is also the ith diagonal element of $\boldsymbol{\Sigma}$. It is well known that POD bases are L_2 optimal and present a good choice for efficient compression of high-dimensional data (Berkooz et al., 1993). A recent alternative for compressing the data consists of spectral POD (Schmidt et al., 2019; Mengaldo and Maulik, 2021; Lario et al., 2021).

2.2 Time-series forecasting

For forecasting a dynamical system from examples of time-series data, we utilize an encoder–decoder framework coupled with long short-term memory neural networks (LSTMs)

(Hochreiter and Schmidhuber, 1997). The encoder–decoder formulation is given by a first step, where a latent representation is derived through information from historical data (in our case, the compressed representation of the full state), i.e.,

$$h^{\text{input}} = h(r_{t-T}, r_{t-T+1}, \ldots, r_t), \tag{4}$$

where h is an LSTM and r_i values are the POD coefficients at a particular time step in the input window of state values from $i = t - T \ldots, t$. The LSTM neural architecture is devised to account for long- and short-term correlations in time-series data through the specification of a hidden state that evolves over time and is affected by each observation of the dynamical system. The basic equations of the LSTM in our context are given by

input gate: $G_i = \varphi_S \circ \mathcal{F}_i^{N_c}(r_t),$

forget gate: $G_f = \varphi_S \circ \mathcal{F}_f^{N_c}(r_t),$

output gate: $G_o = \varphi_S \circ \mathcal{F}_o^{N_c}(r_t),$

internal state: $s_t = G_f \odot s_{t-1} + G_i \odot \left(\varphi_T \circ \mathcal{F}_{r_t}^{N_c}(r_t)\right),$

output: $h_t = G_o \circ \varphi_T(s_t),$ $\tag{5}$

where φ_S and φ_T refer to tangent sigmoid and tangent hyperbolic activation functions, respectively; N_c is the number of hidden layer units in the LSTM network; and $a \odot b$ refers to a Hadamard product of two vectors. Here, $\mathcal{F}^n$ refers to a linear operation given by a matrix multiplication and subsequent bias addition, i.e.,

$$\mathcal{F}^n(x) = \mathbf{W}x + B, \tag{6}$$

where $\mathbf{W} \in \mathbb{R}^{n \times m}$ and $\mathbf{B} \in \mathbb{R}^n$ for $x \in \mathbb{R}^m$. The LSTM implementation is used to encode a sequence of inputs, r_i with $i = t - T \ldots t$ as described in Eq. (4). In other words, the value of $h_t \in \mathbb{R}^n$ at the end of encoding a sequence is h^{input}. The LSTM network's primary utility is the ability to control information flow through time with the use of the gating mechanisms. A quantity that preserves information of past inputs and predictions is the internal state s_t which is updated using the result of the input and forget gates every time the LSTM operations are performed. A greater value of the forget gate (after sigmoidal activation) allows for a greater preservation of past state information through the sequential inference of the LSTM, whereas a smaller value suppresses the influence of the past.

The h^{input} obtained from the previous steps becomes the input to another neural network (the decoder) for making forecasts. In this study, the "decoding" component of the architecture is also given by an LSTM cell that is provided the encoded state information for each time step of the output, i.e.,

$$\left[r_{t+1}, \ldots, r_{t+T_o}\right] = \tilde{h}(h_{t+1}, \ldots, h_{t+T_o}),$$
$$h_{t+1} = h_{t+2} = \ldots = h_{t+T_o} = h^{\text{input}}, \tag{7}$$
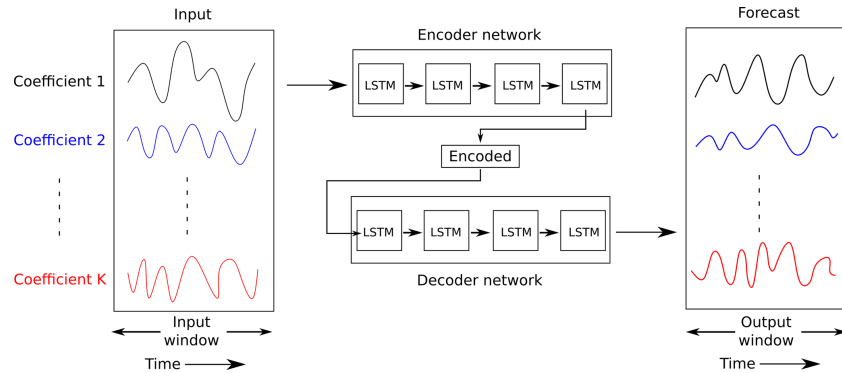
Figure 1. A schematic for forecasting POD coefficients using an encoder–decoder LSTM neural network. The various POD coefficients of an input window are encoded to a latent state on which the forecast window is conditioned.

where $\tilde{h}$ is a different LSTM cell and T_o is the length of the output window. Here, instead of discarding intermediate values of the output operation in the LSTM and focusing solely on the final time step, each output at a particular time step (i.e., h_t) is retained as the forecast of the POD coefficients r at one step in the future. Subsequently, the full-order state variable can be reconstructed by using the precomputed basis functions. A schematic of the architecture is shown in Fig. 1. In the past, several dynamical system forecasts were performed solely with the use of POD–LSTM type learning (Pawar et al., 2019; Mohan and Gaitonde, 2018; Maulik et al., 2021, 2020). We note that several methodological improvements are possible for increasing the quality of the surrogate model, a priori without data assimilation. These include the use of alternate forecast architectures in latent space, such as transformers, neural ordinary differential equations, and gated recurrent units. In addition, the use of nonlinear compression techniques such as autoencoders, instead of linear methods such as proper orthogonal decomposition, may lead to compressive benefits. Finally, the use of adversarial training for forecasting in latent space is also an attractive option for improving accuracy. We note that combining these techniques with our data assimilation formulation would ultimately improve the fidelity of real-time forecasts significantly. However, as will be demonstrated shortly, such approaches can be enhanced significantly with the use of real-time DA from sparse observations.

3 Data assimilation

Data assimilation (DA) is the process of combining information from prior data, imperfect model predictions, and noisy observations to obtain an improved description of the true state x^{true} of a physical system.

The resulting estimate represents the maximum a posteriori estimate, within a Bayesian setting and is referred to as the analysis x^{a}.

The *prior* value represents the current knowledge of the system and is frequently referred to as background in the DA community (Kalnay, 2003). The background is usually an estimate of the state x^{b}, along with the corresponding error covariance matrix $\mathbf{B}$.

The *imperfect predictions* are generated by a model that approximates the physical laws that govern system evolution. The model evolution uses an initial state $x_0 \in \mathbb{R}^n$ at initial time t_0 to obtain states $x_i \in \mathbb{R}^n$ at future times t_i, i.e.,

$$x_i = \mathcal{M}_{t_0 \to t_i}(x_0), \quad i = 1, \cdots, N. \tag{8}$$

The *noisy data* are partial observations of the true state available at discrete time instances. Specifically, measurements $y_i \in \mathbb{R}^m$ of the true physical state $x^{\text{true}}(t_i)$ are taken at discrete times t_i:

$$y_i = \mathcal{H}(x_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_i), \quad i = 1, \cdots, N, \tag{9}$$

where the observation operator $\mathcal{H} : \mathbb{R}^n \to \mathbb{R}^m$ maps the high-dimensional model state space onto the partial (and potentially sparse) observation space. The random observation errors ε_i are assumed to be normally distributed. In general, both the model and the observation operator are nonlinear. These concepts are detailed in the following references: Daley (1993), Kalnay (2003), Sandu and Chai (2011), Sandu et al. (2005), and Carmichael et al. (2008).

Through variational methods, one may solve the DA problem by adjusting a control variable (e.g., model parameters or initial conditions) in order to minimize the discrepancy between model forecasts and observations, in a manner similar to an optimal-control framework. We next review the variational DA formulation.

3.1 Four-dimensional variational data assimilation

Strong-constraint four-dimensional variational (4D-Var) DA processes simultaneously all observations at all times $t_1, t_2, \ldots, t_T$ within the assimilation window (see Fig. 2 for a schematic representation). The control parameters are
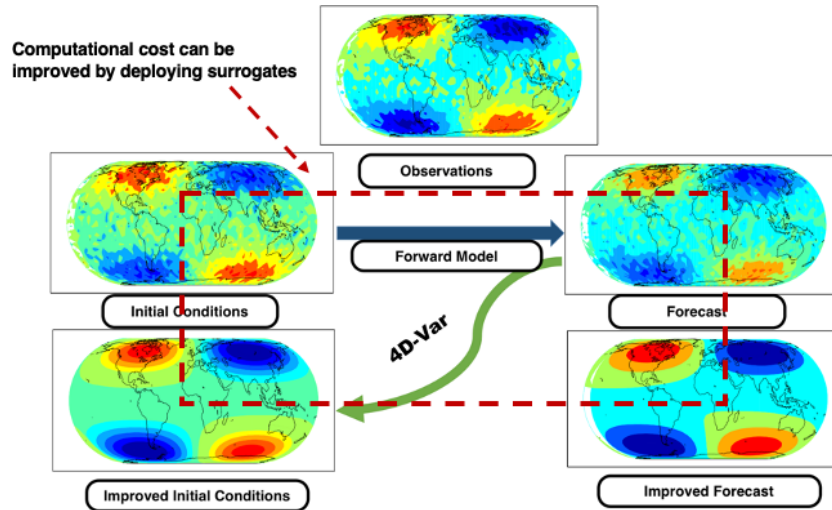
Figure 2. A schematic for 4D-Var. Performing 4D-Var improves the initial conditions of the system, which in turn improves the forecasts. However, 4D-Var requires simulating the expensive model multiple times, and this cost can be mitigated by deploying surrogates.

typically given by the initial conditions x_0, which uniquely determine the state of the system at discrete instances in the future under the assumption that the model of Eq. (8) perfectly represents reality. The background state (i.e., the prior state) is the best estimate of the initial conditions x_0^b and comes with a background error covariance matrix $\mathbf{B}_0$. The 4D-Var method obtains an estimate x_0^a of the true initial conditions by solving the following optimization problem:

$$x_0^a = \arg\min_{x_0} \ \mathcal{J}(x_0) \qquad \text{subject to Eq. (8)}, \tag{10a}$$

$$\mathcal{J}(x_0) = \frac{1}{2}\|x_0 - x_0^b\|_{\mathbf{B}_0^{-1}}^2 + \frac{1}{2}\sum_{i=1}^{T}$$
$$\|\mathcal{H}(x_i) - y_i\|_{\mathbf{R}_i^{-1}}^2. \tag{10b}$$

The first term of the sum of Eq. (10b) captures the mismatch of the solution x_0 from the background x_0^b at t_0. The second term measures the discrepancy between the forecast trajectory (model solutions x_i) and partial observations y_i at future times t_i in the assimilation window. The covariance matrices $\mathbf{B}_0$ and $\mathbf{R}_i$ are usually specified a priori, and their quality influences the accuracy of the resulting solution.

3.2 Data assimilation with a surrogate model

Given a low-dimensional differentiable surrogate model, for instance, the proposed POD–LSTM, which approximates the dynamics of the full-order numerical model, the variational DA process can be accelerated dramatically. If such a surrogate model has predicted $\hat{x}_1$ to $\hat{x}_T$ for a range of time steps in the reduced space (i.e., $\hat{x} \approx Px$, $\hat{x} \in \mathbb{R}^K$, with $K \ll N$) and observations from the real system, y_1 to y_T are available; the objective function of the DA problem may be obtained

by reconstructing the model state from the compressed representation. Here, $P : \mathbb{R}^N \to \mathbb{R}^K$ is the function that maps from physical space to reduced space. In the case of POD-based compression, P is a linear operation that projects from the physical space to the basis spanned by the truncated set of the left singular vectors of an SVD performed on snapshots from full-state model evaluation. If inversion from the truncated subspace is represented as $P^\dagger$, this gives us

$$\mathcal{J}(\hat{x}_0^*) = \frac{1}{2}\|P\hat{x}_0^* - P\hat{x}_0^b\|_{\mathbf{B}_0^{-1}}^2$$
$$+ \frac{1}{2}\sum_{i=1}^{T}\|\mathcal{H}(P^\dagger\hat{x}_i) - y_i\|_{\mathbf{R}_i^{-1}}^2. \tag{11}$$

Note that $\hat{x}_i$, which is an approximation to the compressed state at time t_i, is evaluated using the surrogate model – that is, by propagating $\hat{x}_0^*$ using the surrogate model (LSTM in this case). The operation $P^\dagger\hat{x}_i$ represents reconstruction of the full-order state from its compressed representation. Given a projection operation P with an effective compression ratio (obtained from our dimensionality reduction technique), Eq. (10b) becomes a cost function as shown in Eq. (11) expressed in K dimensions which is amenable to rapid updates of the initial conditions. Moreover, gradients of this objective function with respect to the initial conditions are trivially computable because of the use of automatic differentiation of our LSTM neural network. The overall approach is as follows:

- For each forecast window, collect random observations from the true state.

- Perform an optimization of Eq. (11) by perturbing the inputs to the LSTM neural network. This input is the projected version of the initial-condition state.

– If optimization has converged, perform one forecast with the optimized initial condition. This is the DA-improved forecast.

The optimization methodology used in this article is the sequential least-squares programming approach (Nocedal and Wright, 2006) implemented in SciPy. The neural architecture was deployed using TensorFlow 2.4. Our overall implementation is in Python.

4 Results

We describe the dataset used in our experiments and then present our experimental results.

4.1 Dataset

The data used in this study are a subset of 20 years of output from the regional climate model Weather Research and Forecasting (WRF) version 3.3.1, prepared with methods and configurations described by Wang and Kotamarthi (2014). WRF is a fully compressible and nonhydrostatic regional numerical weather prediction system with proven suitability for a broad range of applications. The WRF simulations used by this study are driven by reanalysis data of NCEP Reanalysis 2 (NCEP-R2) for the period 1984–2003. The NCEP-R2 dataset assimilates many of the observational datasets available to build dynamically consistent gridded fields that are typically used for initialization and the boundary condition setting for forecasting models. The WRF simulation domain is centered at $52.24°$ N, $105.5°$ W and has dimensions of 600×516 horizontal grid points in the west–east and south–north directions with a grid spacing of 12 km, covering most of North America. A spectral-nudging technique is applied for zonal and meridional winds, temperature, and geopotential height at each vertical level above 850 hPa (e.g., around 1.5 km).

We use geopotential height of the 500 hPa pressure surface (Z500 hereafter) from WRF output to demonstrate the approach developed in this study. The geopotential height represents the height of pressure surface above sea level; for Z500, it is around 5.5 km, and it is often referred to as a steering level. The weather systems beneath, near to Earth's surface, roughly move in the same direction as the winds at the 500 hPa level. This is also the level where one can look for vertical motions. Z500 has been used as one of the standard fields for weather forecasting because it does not have very strong local gradients (in contrast to fields such as humidity or precipitation) and does not depend on local conditions such as topography, yet most of the important global flow patterns – such as midlatitude jets and a gradient between poles and Equator – are visible in Z500. Low Z500 values indicate troughs and cyclones (e.g., favorable to precipitation) in the middle troposphere, while high Z500 values indicate ridges and anticyclones.

We retrieve the Z500 data from the WRF output at a grid spacing of 12 km and at 3 h intervals with 515×599 grid cells in the south–north and west–east directions, respectively. We then calculate per-grid-point daily averages to obtain one data value per grid point per day. Spatially, we coarsen the data by five strides (uniform subsampling), which still maintains the spatial patterns of Z500 but reduces the data size significantly. Each daily Z500 snapshot therefore comprises 102×119 grid points of 60×60 km. We have 3287 such daily snapshots for the period 1 January 1984 to 31 December 1991. Despite the coarsening of the original 12 km data, the typical eastward-propagating waves are clearly visible in the Northern Hemisphere. Such waves are also observed in the real atmosphere and are one of the main features of midlatitude weather variability on timescales of several days.

We use the first 6 years of the daily averaged Z500 WRF data (1984–1989) for surrogate training and optimization. Of those data, we select 70 % at random as *training data*, for use in training the supervised ML formulation, and keep the remaining 30 % as *validation data*, for use in tuning the neural-architecture hyperparameters and to control overfitting (a situation in which the trained network predicts well on the training data but not on the test data). We also construct a set of *test data* consisting of 1 year of records (1991) for prediction and evaluation. We skip 1990 so as to ensure that there is no overlap in input windows.

We assume that the observation errors are uncorrelated. This is realistic, as the sensors can be assumed to be independent. However, this assumption might not hold in the case of certain instruments such as lidar, where the errors can be spatially and temporally correlated. We will explore the use of correlated observation errors in our future research. We assume the standard deviation of the observation errors to be approximately 1.5 % of the mean value. In our future studies, in addition to using a realistic observation error covariance matrix, we will also pursue using a flow-dependent background error covariance matrix, as described in Buehner (2005).

4.2 Experiments

We conduct a number of experiments to evaluate the performance of our emulator-assisted DA approach. In the first one, we perform a grid-based search in which we vary both the number of POD modes (5, 10, and 15) and the size of the input window (7, 14, 28, or 42 d of lead-time information) to identify the optimal combination for forecasting. Other hyperparameters such as learning rate, learning rate scheduler, number of LSTM cells, and number of neurons are determined by a combination of experience in previous modeling tasks, considerations of computational efficiency, and limited manual tuning. We set the output forecast window to 20 d to represent a difficult forecast task for the emulator.

We give the complete set of hyperparameters for our problem in Table 1. We found that the lowest validation errors

were achieved when just five modes were retained, a result that matches earlier studies (Lario et al., 2021), where increasing the number of modes is seen to cause difficulties in long-term forecasting (for example, for more than 2 weeks). The training and validation data are used in this phase of our experimentation to identify the best model for performing accelerated DA. We train the neural network architecture by penalizing the loss function on the training data while using the validation data to enforce an early stopping criterion (i.e., prevent overfitting). Once the different models have been trained, the best model is determined by studying the validation losses of all the different architectures. This model is then used for testing on unseen data and for DA. For a further sensitivity analysis of this model, we trained four other models with varying input window sizes (7, 28, 35, and 42) and different random seeds, with other hyperparameters fixed, and utilized them for obtaining ensemble test results. We trained and validated architectures using multiple hyperparameters in parallel by using Ray (Moritz et al., 2018), a TCP/IP-based (Transmission Control Protocol–Internet Protocol) parallelization protocol, on Nvidia A100 graphics processing units (GPUs) of the Argonne Leadership Computing Facility's Theta supercomputer.

Table 1. LSTM hyperparameters used when training the low-dimensional surrogate. The quantity with an asterisk is varied, in addition to random seeds, for sensitivity analyses of the surrogate model where, in addition to the best chosen model here, several other models are trained with different weight initializations and input window sizes.

Neurons per cell	20
Number of stacked cells	2
Initial learning rate	1×10^{-3}
Learning rate decay rate	0.5 (based on 10 epochs patience)
Activation function	ReLU (rectified linear unit)
Input window*	7, 14, 28, or 42 d
Number of modes retained	5
Output window	20 d
Weight initialization	Glorot
Optimizer	Adam

Our first set of assessments test the POD–LSTM emulator without any DA. We show in Fig. 3 (in the left column) how emulator predictions for day 15 (for all forecasts in the test dataset) compare to climatology. Climatology, here, refers to the average forecast on a specific forecast day based on averaged training data. For instance, if geopotential height is to be forecast on 7 December 1991, the climatology prediction would be the average of 7 December geopotential height values for 1984–1989. The metrics we use for comparison first include the cosine similarity improvement as shown in Fig. 3a. Here, the cosine similarity (which captures the alignment between prediction and truth) obtained from climatology is subtracted from that obtained from the emulator fore-

cast. Thus, in this case, negative (blue) regions are where climatology captures the temporal trend of the forecast better than our emulator. In Fig. 3c, we subtract the mean absolute error (MAE) of the emulator predictions from that of climatology. Here, the blue regions are where climatology is more accurate on average than the emulator. Figure 3e merely shows the MAE for the emulator. From this analysis, it is clear that for large regions in the data domain (particularly in the north), the emulator performs quite poorly.

We next assess results from applying DA through random observations at 5000 locations of the domain to see if results obtained with the emulator can be improved. Here, sparse observations at each time step of the output window are used within the optimization statement introduced in Eq. (11) to update the initial conditions (i.e., the input window). We emphasize that the observations are obtained randomly from the true state of the system during forecasts, which corresponds to the test data introduced previously.
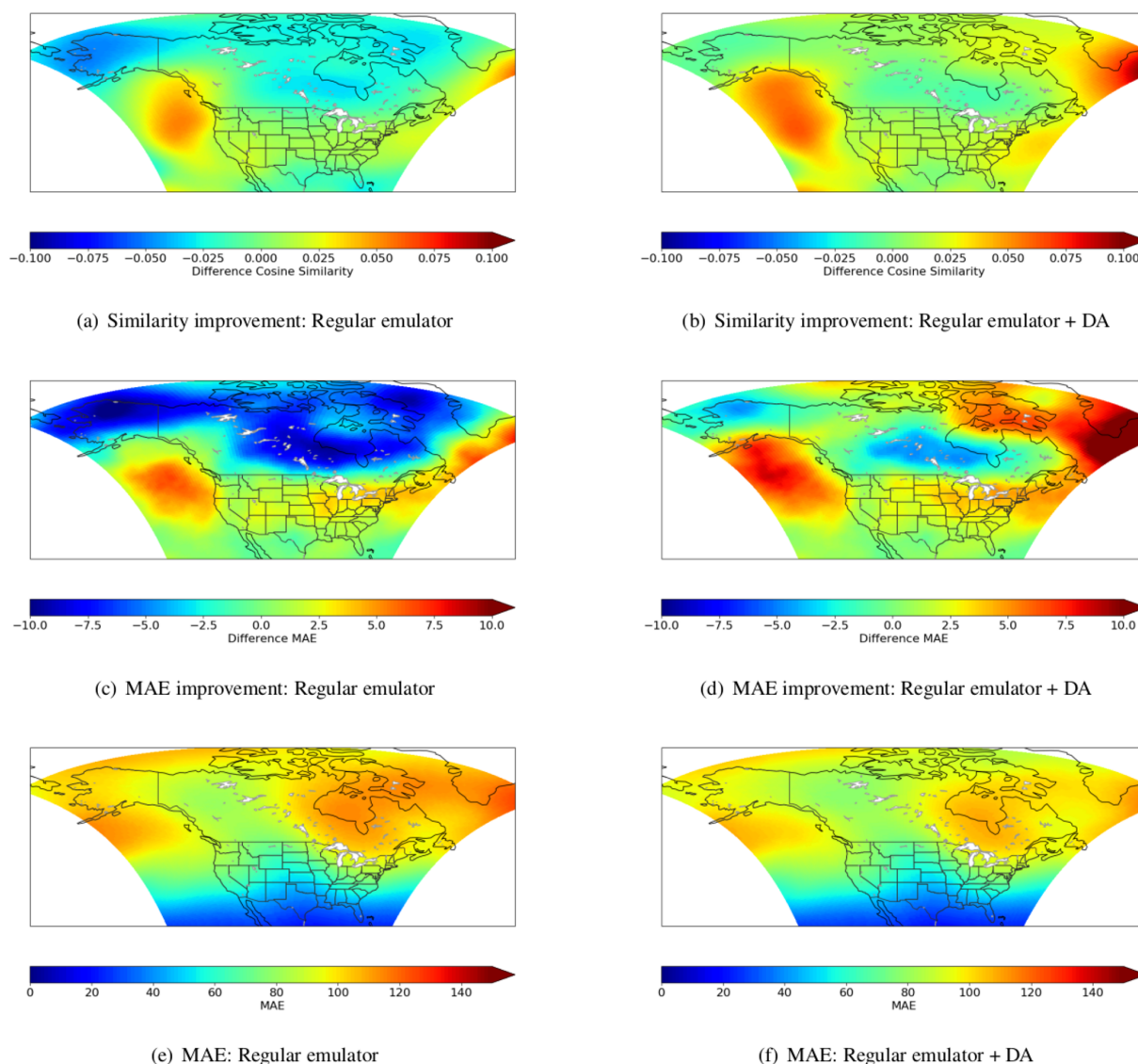
The results for using DA, in the right column of Fig. 3, show that the use of sensor information aids the forecast immensely, improving performance in all metrics significantly by virtue of DA. In particular, the proposed augmentation (i.e., the use of DA during forecasts) reduces forecast errors considerably in regions where the sole use of the emulator was not competitive. We provide MAE assessments for our 20 d output averaged over the testing data range in Fig. 4, where we compare the raw emulator outputs, climatology, persistence, and the DA-corrected outputs. We also provide confidence intervals for the regular (i.e., without DA) and DA-corrected emulators, where the five different emulators are trained with different random seeds and input window durations to assess the sensitivity to the initialization of the training as well as the automatic-differentiation-based optimization. The results indicate that the DA-corrected emulation is the most accurate and consistent forecast technique.

We also provide in Fig. 5 a comparison with another benchmark, persistence, which uses the state of the last day in the input window as the forecast for each day of the output window. Persistence is seen to be more accurate for short lead times but is outperformed by both climatology and data-driven methods for longer forecast durations.

Our choice of 20 d for the forecast window was motivated by the results of Rasp et al. (2020) and Rasp and Thuerey (2021), who performed analyses for 5 d forecasts. Our study increased the forecast window to 4 times the original assessments for geopotential height to demonstrate the possibility of improved forecasting of *any emulator* using automatic-differentiation-enabled variational DA. In many subregions of the computational domain, the use of DA with the emulator is also seen to improve on climatology predictions – even at large lead times (20 d out), where classical data-driven and numerical forecast models are typically less accurate than climatology.

Our method also delivers large computational gains. A 1-year forecast with the emulator-assisted DA performed each

(a) Similarity improvement: Regular emulator

(b) Similarity improvement: Regular emulator + DA

(c) MAE improvement: Regular emulator

(d) MAE improvement: Regular emulator + DA

(e) MAE: Regular emulator

(f) MAE: Regular emulator + DA

Figure 3. Improvements over climatology for the regular emulator **(a, c, e)** and DA-corrected emulator **(b, d, f)** when using the best set of hyperparameters as determined on validation data. Here cosine similarity captures an agreement in the trend between forecast and truth, and MAE refers to the mean absolute error between the same. These contours are averaged over all forecasts.

day takes approximately 1 h on a single processor core without any accelerator hardware. In contrast, the original PDE-based simulation required 21 600 core hours for a 1-year forecast with a 515×599 grid. (A coarse-grained forecast for 1 year on the 103×120 grid used here takes 172 core hours, but the resulting flow field does not adequately reproduce the fidelity of the 515×599 grid.) One can observe significant speedup ($\sim 10^4$ times) for the emulation of the geopotential height, even without factoring the cost of an additional variational DA step. Furthermore, the solution to the 4D-Var problem, which yields an improved initial condition, requires on the order of 100 model runs, where each model run can be several orders of magnitude more expensive than our emulator.

5 Conclusions

We have described how a differentiable reduced-order surrogate geophysical forecasting model may be integrated into an outer-loop optimization technique whereby real-time observations of the true solution are used to improve the forecast of the surrogate. We use such observations to improve the initial condition of the surrogate model such that an optimization statement given by the classical four-dimensional variational DA objective function is minimized. The use of the reduced-order surrogate converts a high-dimensional optimization to one that is given by the dimensionality of the compressed representation and the duration of the input window for forecasting. Our optimization is thus performed
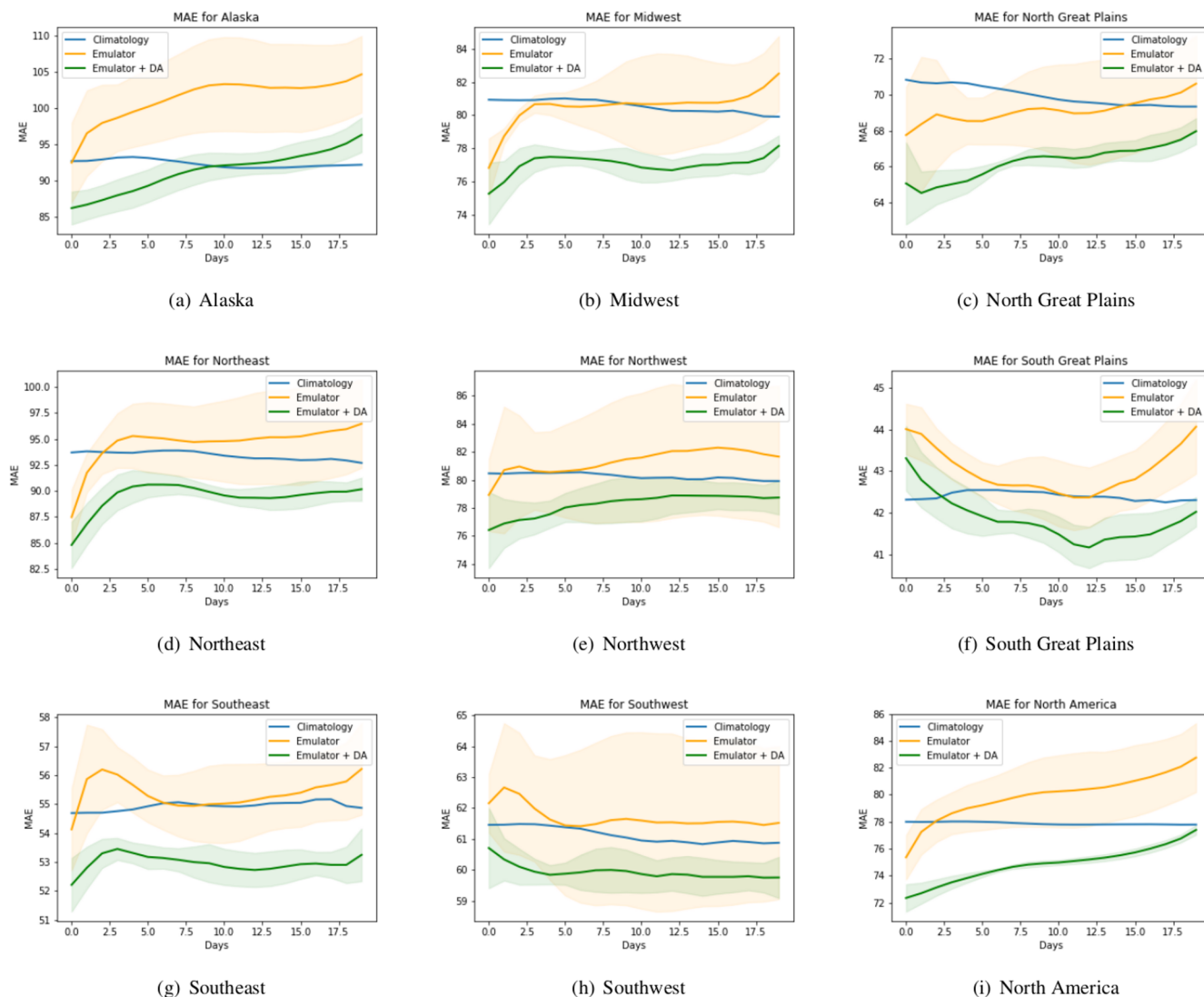
Figure 4. Geopotential height forecast MAEs for 20 d, relative to the true test data, for seven National Climate Assessment subregions of continental North America (Reidmiller et al., 2018), from both a regular ML emulator and the same emulator corrected by variational DA. Confidence intervals, for five separately trained emulators, encapsulate the effects of perturbations to the random seed and the input window. DA-based correction gives both lower MAEs and narrower confidence intervals.

rapidly, given sparse and random observations from the true flow field, without any access to high-performance computing resources. Computational costs are reduced by 4 orders of magnitude, by virtue of the surrogate-assisted forecasting and variational DA, when compared to a classical equation-based forecasting of the dynamics. We assess our model on a real-world forecasting task for geopotential height in the continental US and obtain competitive results with respect to climatology and persistence baselines for mean absolute error and cosine similarity.

Author contributions. RM designed the study, wrote the code, performed the experiments, and analyzed the results. VR and EC developed the variational data assimilation. JW and RK developed the model output used for developing the surrogate. PB and BL assisted with the development of the surrogate model and algorithms. IF participated in the model analysis and development of the manuscript.
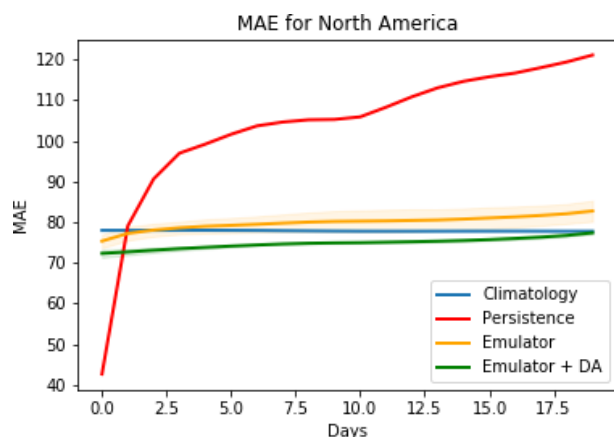
Figure 5. Geopotential height forecast MAEs for 20 d for continental North America, relative to the true test data. Results describe performance of a regular ML emulator and the same emulator corrected by variational data assimilation. Here we also show the results of persistence, which is outperformed in the long-term prediction regime as expected.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

Akella, S. and Navon, I.: Different approaches to model error formulation in 4D-Var: A study with high-resolution advection schemes, Tellus A, 61, 112–128, 2009.

Bauer, H.-S., Schwitalla, T., Wulfmeyer, V., Bakhshaii, A., Ehret, U., Neuper, M., and Caumont, O.: Quantitative precipitation estimation based on high-resolution numerical weather prediction and data assimilation with WRF – a performance test, Tellus A, 67, 25047, https://doi.org/10.3402/tellusa.v67.25047, 2015.

Berkooz, G., Holmes, P., and Lumley, J. L.: The proper orthogonal decomposition in the analysis of turbulent flows, Annu. Rev. Fluid Mech., 25, 539–575, 1993.

Brajard, J., Carrassi, A., Bocquet, M., and Bertino, L.: Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the Lorenz 96 model, J. Comput. Sci., 44, 101171, https://doi.org/10.1016/j.jocs.2020.101171, 2020.

Brajard, J., Carrassi, A., Bocquet, M., and Bertino, L.: Combining data assimilation and machine learning to infer unresolved scale parametrization, Philos. T. R. Soc. A, 379, 20200086, https://doi.org/10.1098/rsta.2020.0086, 2021.

Buehner, M.: Ensemble-derived stationary and flow-dependent background-error covariances: Evaluation in a quasi-operational NWP setting, Q. J. Roy. Meteor. Soc., 131, 1013–1043, 2005.

Cardinali, C., Žagar, N., Radnoti, G., and Buizza, R.: Representing model error in ensemble data assimilation, Nonlinear Proc. Geophys., 21, 971–985, 2014.

Carmichael, G. R., Sandu, A., Chai, T., Daescu, D. N., Constantinescu, E. M., and Tang, Y.: Predicting air quality: Improvements through advanced methods to integrate models and measurements, J. Comput. Phys., 227, 3540–3571, 2008.

Casas, C. Q., Arcucci, R., Wu, P., Pain, C., and Guo, Y.-K.: A reduced order deep data assimilation model, Physica D: Nonlinear Phenomena, 412, 132615, https://doi.org/10.1016/j.physd.2020.132615, 2020.

Chatterjee, A.: An introduction to the proper orthogonal decomposition, Current Science, 78, 808–817, 2000.

Chennault, A., Popov, A. A., Subrahmanya, A. N., Cooper, R., Karpatne, A., and Sandu, A.: Adjoint-Matching Neural Network Surrogates for Fast 4D-Var Data Assimilation, CoRR, abs/2111.08626, https://doi.org/10.48550/ARXIV.2111.08626, 2021.

Daley, R.: Atmospheric Data Analysis, Cambridge University Press, 2, https://books.google.com/books (last access: 27 April 2022), 1993.

Errico, R. M.: What is an adjoint model?, B. Am. Meteorol. Soc., 78, 2577–2592, 1997.

Errico, R. M. and Raeder, K. D.: An examination of the accuracy of the linearization of a mesoscale model with moist physics, Q. J. R. Meteor. Soc., 125, 169–195, 1999.

Errico, R. M., Vukicevic, T., and Raeder, K.: Examination of the accuracy of a tangent linear model, Tellus A, 45, 462–477, 1993.

Frerix, T., Kochkov, D., Smith, J. A., Cremers, D., Brenner, M. P., and Hoyer, S.: Variational Data Assimilation with a Learned Inverse Observation Operator, in: Proceedings of the 38th International Conference on Machine Learning (ICML), Proceedings of Machine Learning Research (PMLR), 139, 3449–3458, https://proceedings.mlr.press/v139/frerix21a.html (last access: 27 April 2022), 2021.

Glimm, J., Hou, S., Lee, Y., Sharp, D., and Ye, K.: Sources of uncertainty and error in the simulation of flow in porous media, Comput. Appl. Math., 23, 109–120, 2004.

Gustafsson, N., Janjić, T., Schraff, C., Leuenberger, D., Weissmann, M., Reich, H., Brousseau, P., Montmerle, T., Wattrelot, E., Bučánek, A., Mile, M., Hamdi, R., Lindskog, M., Barkmeijer, J., Dahlbom, M., Macpherson, B., Ballard, S., Inverarity, G., Carley, J., Alexander, C., Dowell, D., Liu, S., Ikuta, Y., and Fujita, T.: Survey of data assimilation methods for convective-scale numerical weather prediction at operational centres, Q. J. R. Meteor. Soc., 144, 1218–1256, https://doi.org/10.1002/qj.3179, 2018.

Hansen, J. A.: Accounting for model error in ensemble-based state estimation and forecasting, Mon. Weather Rev., 130, 2373–2391, 2002.

Hatfield, S., Chantry, M., Dueben, P., Lopez, P., Geer, A., and Palmer, T.: Building Tangent-Linear and Adjoint Models for Data Assimilation With Neural Networks, J. Adv. Model. Earth Sy., 13, e2021MS002521, https://doi.org/10.1029/2021MS002521, 2021.

Hochreiter, S. and Schmidhuber, J.: Long short-term memory, Neural computation, 9, 1735–1780, 1997.

Holmes, P., Lumley, J. L., Berkooz, G., and Rowley, C. W.: Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, p. 386, ISBN 9781107008250, 2012.

Kalnay, E.: Atmospheric Modeling, Data Assimilation and Predictability, Cambridge University Press, p. 341, ISBN 9780521796293, 2003.

Lario, A., Maulik, R., Rozza, G., and Mengaldo, G.: Neural-network learning of SPOD latent dynamics, arXiv preprint arXiv:2110.09218, p. 27, https://doi.org/10.48550/arXiv.2110.09218, 2021.

Le Dimet, F. and Talagrand, O.: Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects, Tellus A, 38, 97–110, 1986.

Le Guen, V. and Thome, N.: Disentangling physical dynamics from unknown factors for unsupervised video prediction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 13–19 June 2020, Seattle, WA, USA, 11474–11484, https://doi.org/10.1109/CVPR42600.2020.01149, 2020.

Lorenc, A. C. and Rawlins, F.: Why does 4D-Var beat 3D-Var?, Quarterly J. Roy. Meteorol. Soc., 131, 3247–3257, 2005.

Lynch, P.: The origins of computer weather prediction and climate modeling, J. Comput. Phys., 227, 3431–3444, 2008.

Mack, J., Arcucci, R., Molina-Solana, M., and Guo, Y.-K.: Attention-based convolutional autoencoders for 3D-variational data assimilation, Comput. Method. Appl. M., 372, 113291, https://doi.org/10.1016/j.cma.2020.113291, 2020.

Maulik, R.: AIEADA/LSTM_Var_Prototype: GMD-2021-415: AIEADA 1.0: Efficient high-dimensional variational data assimilation with machine-learned reduced-order models (GMD_v1), Zenodo [data set] [code], https://doi.org/10.5281/zenodo.6382921, 2022.

Maulik, R. and Mengaldo, G.: PyParSVD: A streaming, distributed and randomized singular-value-decomposition library, 2021 7th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD-7), p. 19-25, https://doi.org/10.1109/DRBSD754563.2021.00007, 2021.

Maulik, R., Egele, R., Lusch, B., and Balaprakash, P.: Recurrent neural network architecture search for geophysical emulation, in: SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, Atlanta, Georgia, IEEE, p. 14, ISBN 9781728199986, 2020.

Maulik, R., Lusch, B., and Balaprakash, P.: Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders, Physics of Fluids, 33, 037106, https://doi.org/10.1063/5.0039986, 2021.

Mengaldo, G. and Maulik, R.: PySPOD: A Python package for Spectral Proper Orthogonal Decomposition (SPOD), Journal of Open Source Software, 6, 2862, https://doi.org/10.21105/joss.02862, 2021.

Mohan, A. T. and Gaitonde, D. V.: A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks, arXiv, preprint arXiv:1804.09269, https://doi.org/10.48550/arXiv.1804.09269, 2018.

Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M. I., and Stoica, I.: Ray: A distributed framework for emerging AI applications, in: 13th USENIX Symposium on Operating Systems Design and Implementation, 561–577, ISBN 9781931971478, 2018.

Nocedal, J. and Wright, S. J.: Sequential quadratic programming, Numerical Optimization, 529–562, https://doi.org/10.1007/978-0-387-40065-5_18, 2006.

Orrell, D., Smith, L., Barkmeijer, J., and Palmer, T. N.: Model error in weather forecasting, Nonlin. Processes Geophys., 8, 357–371, https://doi.org/10.5194/npg-8-357-2001, 2001.

Palmer, T., Shutts, G., Hagedorn, R., Doblas-Reyes, F., Jung, T., and Leutbecher, M.: Representing model uncertainty in weather and climate prediction, Annu. Rev. Earth Planet. Sci, 33, 163–93, 2005.

Pawar, S. and San, O.: Data assimilation empowered neural network parametrizations for subgrid processes in geophysical flows, Physical Review Fluids, 6, 050501, https://doi.org/10.1103/PhysRevFluids.6.050501, 2021.

Pawar, S., Rahman, S., Vaddireddy, H., San, O., Rasheed, A., and Vedula, P.: A deep learning enabler for nonintrusive reduced order modeling of fluid flows, Physics of Fluids, 31, 085101, https://doi.org/10.1063/1.5113494, 2019.

Pawar, S., Ahmed, S. E., San, O., Rasheed, A., and Navon, I. M.: Long short-term memory embedded nudging schemes for nonlinear data assimilation of geophysical flows, Physics of Fluids, 32, 076606, https://doi.org/10.1063/5.0012853, 2020.

Penny, S. G., Smith, T. A., Chen, T.-C., Platt, J. A., Lin, H.-Y., Goodliff, M., and Abarbanel, H. D. I.: Integrating recurrent neural networks with data assimilation for scalable data-driven state estimation, arXiv preprint, arXiv:2109.12269, 14, e2021MS002843, https://doi.org/10.1029/2021MS002843, 2021.

Popov, A. A. and Sandu, A.: Multifidelity ensemble Kalman filtering using surrogate models defined by physics-informed autoencoders, arXiv preprint, arXiv:2102.13025, https://doi.org/10.48550/arXiv.2102.13025, 2021.

Rao, V. and Sandu, A.: A posteriori error estimates for the solution of variational inverse problems, SIAM/ASA, Journal on Uncertainty Quantification, 3, 737–761, 2015.

Rasp, S. and Thuerey, N.: Data-Driven Medium-Range Weather Prediction With a Resnet Pretrained on Climate Simulations:

A New Model for WeatherBench, J. Adv. Model. Earth Sy., 13, e2020MS002405, https://doi.org/10.1029/2020MS002405, 2021.

Rasp, S., Dueben, P. D., Scher, S., Weyn, J. A., Mouatadid, S., and Thuerey, N.: WeatherBench: A benchmark data set for data-driven weather forecasting, J. Adv. Model. Earth Sy., 12, e2020MS002203, https://doi.org/10.1029/2020MS002203, 2020.

Reidmiller, D., Avery, C., Easterling, D., Kunkel, K., Lewis, K., Maycock, T., and Stewart, B.: Fourth national climate assessment, Volume II: Impacts, Risks, and Adaptation in the United States, U.S. Global Change Research Program, Washington, DC, USA, 1515 pp., https://doi.org/10.7930/NCA4.2018, 2018.

Sandu, A. and Chai, T.: Chemical data assimilation – An overview, Atmosphere, 2, 426–463, 2011.

Sandu, A., Daescu, D. N., Carmichael, G. R., and Chai, T.: Adjoint sensitivity analysis of regional air quality models, J. Comput. Phys., 204, 222–252, 2005.

Schmidt, O. T., Mengaldo, G., Balsamo, G., and Wedi, N. P.: Spectral empirical orthogonal function analysis of weather and climate data, Mon. Weather Rev., 147, 2979–2995, 2019.

Trémolet, Y.: Accounting for an imperfect model in 4D-Var, Q. J. R. Meteor. Soc., 132, 2483–2504, https://doi.org/10.1256/qj.05.224, 2006.

Trémolet, Y.: Model-error estimation in 4D-Var, Q. J. R. Meteor. Soc., 133, 1267–1280, https://doi.org/10.1002/qj.94, 2007.

Wang, J. and Kotamarthi, V. R.: Downscaling with a nested regional climate model in near-surface fields over the contiguous United States, J. Geophys. Res.-Atmos., 119, 8778–8797, 2014.

Zupanski, D. and Zupanski, M.: Model error estimation employing an ensemble data assimilation approach, Mon. Weather Rev., 134, 1337–1354, 2006.