Journal of Computational Physics 477 (2023) 111918

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Deep learning-enhanced ensemble-based data assimilation for high-dimensional nonlinear dynamical systems

Ashesh Chattopadhyay [a,*], Ebrahim Nabizadeh [a], Eviatar Bach [b,c], Pedram Hassanzadeh [a,d,*]

[a] Department of Mechanical Engineering, Rice University, Houston, TX, United States of America
[b] Geosciences Department and Laboratoire de Météorologie Dynamique (CNRS and IPSL), École Normale Supérieure and PSL University, Paris, France
[c] Division of Geological and Planetary Sciences, California Institute of Technology, Pasadena, CA, United States of America
[d] Department of Earth, Environmental and Planetary Sciences, Rice University, Houston, TX, United States of America

ARTICLE INFO

Article history:
Received 9 June 2022
Received in revised form 7 December 2022
Accepted 4 January 2023
Available online 10 January 2023

Keywords:
Data assimilation
High-dimensional systems
Ensemble Kalman filter
Deep learning
Forecasting
Turbulence

ABSTRACT

Data assimilation (DA) is a key component of many forecasting models in science and engineering. DA allows one to estimate better initial conditions using an imperfect dynamical model of the system and noisy/sparse observations available from the system. Ensemble Kalman filter (EnKF) is a DA algorithm that is widely used in applications involving high-dimensional nonlinear dynamical systems. However, EnKF requires evolving large ensembles of forecasts using the dynamical model of the system. This often becomes computationally intractable, especially when the number of states of the system is very large, e.g., for weather prediction. With small ensembles, the estimated background error covariance matrix in the EnKF algorithm suffers from sampling error, leading to an erroneous estimate of the analysis state (initial condition for the next forecast cycle). In this work, we propose hybrid ensemble Kalman filter (H-EnKF), which is applied to a two-layer quasi-geostrophic turbulent flow as a test case. This framework utilizes a pre-trained deep learning-based data-driven surrogate that inexpensively generates and evolves a large data-driven ensemble of the states to accurately compute the background error covariance matrix with smaller sampling errors. The H-EnKF framework outperforms EnKF with only dynamical model or only the data-driven surrogate, and estimates a better initial condition without the need for any ad-hoc localization strategies. H-EnKF can be extended to any ensemble-based DA algorithm, e.g., particle filters, which are currently too expensive to use for high-dimensional systems.

1. Introduction

Data assimilation (DA) is an indispensable component in many of the forecasting models used for applications in science and engineering [1–5]. DA allows one to estimate better initial conditions for a system from which noisy and sparse observations are available, along with an imperfect dynamical model of the system. These initial conditions are then used by the dynamical model to predict the future states of the system. The quality of future forecasts strongly depends on the accuracy of the initial conditions. This is especially important in chaotic systems, where even a small error in an initial condition

* Corresponding authors.
E-mail address: pedram@rice.edu (P. Hassanzadeh).

https://doi.org/10.1016/j.jcp.2023.111918
0021-9991/

can result in drastically different forecasts [6]. In such systems, an accurate initial condition is of the utmost importance for predictive dynamical models to provide accurate forecasts. DA is critical in various areas of engineering and sciences, such as weather prediction [7–10], environmental and geophysical flows [11–13], combustion systems [14–16], aeronautics [17], hydrology [18], acoustics [19], and fluid mechanics [20,21].

There are two main categories of DA algorithms: ensemble-based methods and variational methods. Ensemble-based DA algorithms such as ensemble Kalman filter (EnKF) were proposed for estimating better initial conditions from noisy observations assuming Gaussian observation noise [22]. Several other DA algorithms such as variational methods (e.g., 3D-Var and 4D-Var) or a combination of ensemble and variational algorithms also exist [23,24]. 4D-Var requires obtaining the adjoint of the system's dynamical model, which is often a difficult and non-trivial task. Moreover, 4D-Var optimizes a cost function which can be computationally expensive. Ensemble-based algorithms do not require obtaining adjoints but require evolving a large ensemble of forecasts of the dynamical system. In this paper, we will focus on ensemble-based algorithms for DA.

A major challenge in EnKF is generating and evolving a large number of ensemble members of the dynamical model in time. The accuracy of the background error covariance matrix in EnKF (which affects the performance of EnKF) depends on the ensemble size [25]. For a full-rank estimation of the background error covariance matrix, the number of ensemble members should be of the order of the number of states, s, in the system, which can be large (e.g., in the weather system, $s \approx O(10^7) - O(10^8)$). Having said that, an accurate estimation of the covariance matrix that leads to a divergence-free filter, even with rank deficiency, requires one to only generate and evolve ensemble members up to the number of unstable and neutral Lyapunov vectors of the system [26,27]. However, for practical systems, the number of unstable and neutral Lyapunov vectors would still be a very large number and evolving such a large number of ensemble members over multiple time steps becomes computationally intractable. Thus, fewer ensemble members are typically used in practice ($\approx O(50)$ in operational weather models [28]). These covariance matrices, generated from a smaller number of ensemble members, are rank-deficient and suffer from sampling error that degrades the quality of the estimated initial condition (often referred to as the "analysis state"). For this reason, various ad-hoc localization strategies have been proposed to remove spurious long-range spatial correlations in the covariance matrix [3]. However, in this process, one may also remove physically-consistent long-range spatial correlations and this can adversely affect the performance of EnKF and thus, the quality of forecasts [29]. There are other methods to estimate the background error covariance matrix without evolving a large ensemble, e.g., the stochastic Galerkin method [30].

In recent years, we have seen an increase in interest at the intersection of machine learning (ML) and DA for applications in dynamical systems. For example, in Yang et al. [31], a generative model was used to facilitate ensemble generation for analog-based DA. In our previous work [32], we have shown that a data-driven weather forecasting model can be integrated with a sigma-point ensemble Kalman filter to obtain accurate initial conditions for forecasting. Tsuyuki et al. [33] integrated ML with an EnKF to perform state estimation in a nonlinear dynamical system using a small number of ensemble members. Maulik et al. [34] used 4D-Var-based DA with ML for forecasting in high-dimensional dynamical systems. Penny et al. [35] used a recurrent neural network and 4D-Var for scalable state estimation. Chen et al. [36] showed the application of DA in ML-based forecasts in complex turbulent flows with partial observations. There are several other studies that have shown different applications, where ML has been integrated with DA, e.g., predicting subgrid-scale processes in multi-scale chaotic systems [37,38], closed-form equation discovery of model error [39], etc.

In this paper, we propose hybrid ensemble Kalman filter (H-EnKF), a hybrid algorithm for enhancing the performance of EnKF without resorting to localization strategies. H-EnKF leverages deep learning to build a data-driven surrogate of the dynamical model of the system. This surrogate is used to generate and evolve a large number of data-driven ensemble members, $O(s)$, to compute the background error covariance matrix of the dynamical system. The data-driven model is trained on the full state of the system and serves as a computationally cheap surrogate to predict the evolution of the states, and is used just for estimating the background error covariance matrix. While the less-accurate but large data-driven ensembles are used to compute the background error covariance matrix with low sampling error, the numerical model of the system is used to generate and evolve a small number of ensemble members, g, where $g \ll s$ (note that g could be just 1). This small number of ensemble members are used to compute an accurate background forecast state to be used in the H-EnKF algorithm. In order to demonstrate the performance of H-EnKF, we have applied it to a well-known test-bed for fully turbulent geophysical flows: the two-layer quasi-geostrophic (QG) system [40,41].

The remainder of the paper is organized into several sections. In section 2, we describe the QG system, the numerical solver used to simulate the system, the data-driven surrogate model, and the EnKF algorithm. In section 3, we introduce our proposed H-EnKF algorithm. Section 4 describes the metrics to evaluate the performance of EnKF and H-EnKF. In section 5, the performance of the algorithms, in terms of accuracy and cost, is presented, followed by summary and discussion in section 6.

2. Method

2.1. Two-layer QG system

The dimensionless dynamical equations of the two-layer QG flow have been developed following Lutsko et al. [40] and Nabizadeh et al. [41]. The system consists of two constant density layers with a β-plane approximation in which the meridional temperature gradient is relaxed towards an equilibrium profile. The system's equations are:

Table 1
Number of layers and filters in the U-NET architecture used as the data-driven surrogate model.

Number	Layer	Number of Filters
1	5×5 2D Convolution	32
2	5×5 2D Convolution	32
3	2×2 Max Pooling	–
4	5×5 2D Convolution	32
5	5×5 2D Convolution	32
6	2×2 Max Pooling	–
7	5×5 2D Convolution	32
8	5×5 2D Convolution	32
9	Up-sampling	–
10	Concatenation	–
11	5×5 2D Convolution	32
12	5×5 2D Convolution	32
13	Up-sampling	–
14	5×5 2D Convolution	32
15	Concatenation	–
16	5×5 2D Convolution	32
17	5×5 2D Convolution	32
18	5×5 2D Convolution	2

$$\frac{\partial q_k}{\partial t} + J(\psi_k, q_k) = -\frac{1}{\tau_d}(-1)^k(\psi_1 - \psi_2 - \psi_R) - \frac{1}{\tau_f}\delta_{k2}\nabla^2\psi_k - \nu\nabla^8 q_k. \tag{1}$$

Here, q is potential vorticity

$$q_k = \nabla^2\psi_k + (-1)^k(\psi_1 - \psi_2) + \beta y, \tag{2}$$

where ψ_k is the streamfunction of the system. In Eqs. (1) and (2), k denotes the upper ($k = 1$) and lower ($k = 2$) layers. β is the $y-$gradient of the Coriolis parameter. τ_d is the Newtonian relaxation time scale and τ_f is the Rayleigh friction time scale, which only acts on the lower layer. δ_{k2} is the Kronecker $\delta-$ function. J denotes the Jacobian. ν denotes the hyperdiffusion coefficient. We have introduced a baroclinically unstable jet at the center of a zonally periodic channel by setting $\psi_1 - \psi_2$ to be equal to a hyperbolic secant centered at $y = 0$. When eddy fluxes are absent, ψ_2 is identically zero, making zonal velocity in the upper layer, $u_1(y) = -\frac{\partial \psi_1}{\partial y} = -\frac{\partial \psi_R}{\partial y}$, where we set

$$-\frac{\partial \psi_R}{\partial y} = \mathrm{sech}^2\left(\frac{y}{\sigma}\right). \tag{3}$$

σ is the width of the jet. Parameters of the model are set following previous works [40,41]; $\beta = 0.19$, $\sigma = 3.5$, $\tau_f = 15$, and $\tau_d = 100$.

To non-dimensionalize the equations, we have used the maximum strength of the equilibrium velocity profile as the velocity scale (U) and the deformation radius (L) for the length scale. The system's time scale (L/U) is referred to as the "advective time scale" (τ_{adv}) which is approximately 6 h in this system.

2.2. Numerical solver

The spatial discretization is spectral in both x and y, where we have retained 96 and 192 Fourier modes, respectively. The length and width of the domain are equal to 46 and 68, respectively. Sponge layers are applied to the northern and southern boundaries. Note that the domain is wide enough for the sponges to not affect the dynamics. Here $5\tau_{adv} \approx 1$ Earth day $\approx 200\Delta t$, where $\Delta t = 0.025$ is the time step of the leapfrog time integrator used in the numerical scheme.

2.3. The data-driven model: U-NET

To build a data-driven surrogate model to approximate the QG dynamics, we adopt a U-NET architecture [42]. The choice of this architecture is inspired by our previous work in data-driven weather forecasting [32]. The details of the architecture are provided in Table 1 and a schematic is shown in Fig. 1. The U-NET allows one to extract small-scale features in the encoder that are directly passed into the decoder as skipped connections. This has been shown to result in improved performance on turbulent flow prediction [43]. The U-NET architecture is trained on N_{tr} noise-free samples (we have performed experiments with $N_{tr} = 10^5$ and $N_{tr} = 10^4$ samples; see section 5.1) of the system's state, i.e., streamfunction, $\psi_k(t)$, $k = \{1, 2\}$. The input to the network is $\psi_k(t)$ and the target is $\psi_k(t + \gamma\Delta t)$. We have shown in a previous study [32] that

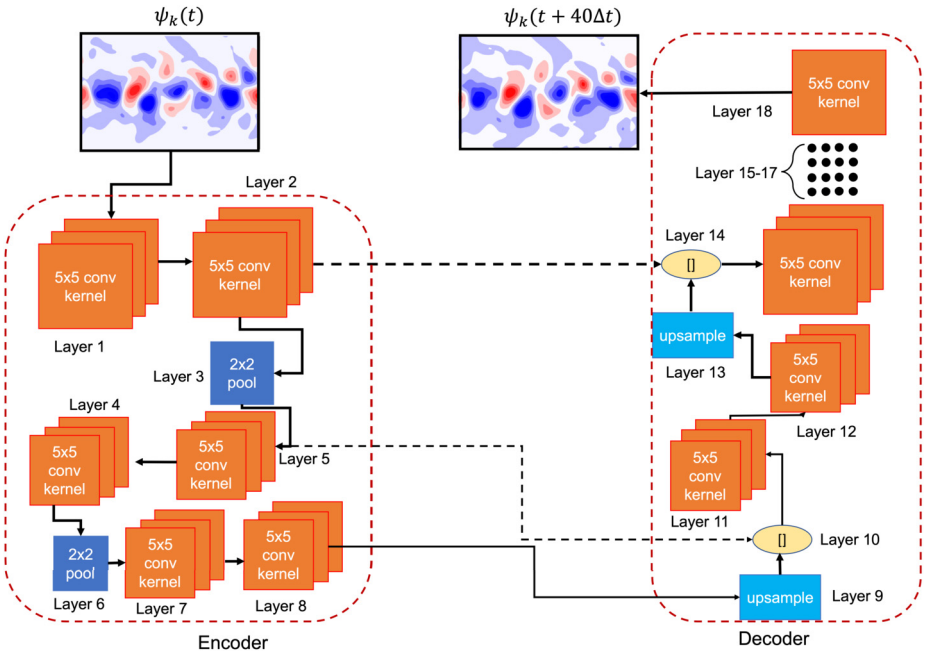

Fig. 1. Schematic of the U-NET model used as a surrogate for data-driven prediction of $\psi_k(t)$ in the two-layer QG system. The input to the model is the system's full state, $\psi_k(t)$ and the output is $\psi_k(t+40\Delta t)$, where $k=\{1,2\}$. The detailed information of the number of layers, their index number, number of filters, and size of convolutional kernels after hyperparameter optimization is given in Table 1. Here, $40\Delta t \approx 6h$.

large values of γ improve the prediction horizon of the data-driven model over small values of γ. Hence, $\gamma > 1$ would result in a longer prediction horizon than $\gamma = 1$. However, in the H-EnKF algorithm (see section 3 for more details) an accurate computation of the background error covariance matrix with data-driven ensembles requires one to evolve the data-driven model for a sufficient number of time steps as well. Since we obtain observations from the system at every $200\Delta t$ (1 DA cycle), γ has to be smaller than 200. Hence, the aim is to choose a γ such that the U-NET has a good prediction horizon while the data-driven ensembles can evolve sufficiently to provide an accurate covariance matrix as well. In this paper, we have chosen $\gamma = 40$, such that $\gamma\Delta t$ matches τ_{adv} (≈ 6 h) of the QG system.

The samples of $\psi_k(t)$ are obtained from numerically solving Eq. (1) and Eq. (2). Each sample corresponds to a different time (t). The training and testing sets are obtained from independent simulations starting from different random initial conditions so that there is no correlation between the training and testing sets. The hyperparameters of the U-NET are determined after extensive trial and error.

As shown by us and others in previous studies, the performance of the U-NET can be improved by incorporating the symmetries in the system inside the architecture [44,32] or by using physics-based regularizers in the loss function of the U-NET [45]. However, in the H-EnKF framework, we only need the U-NET to predict short-term evolution ($200\Delta t \approx$ 1 Earth day into the future, more details in section 3). For such a short-term forecast, the performance of the U-NET remains roughly the same as compared to an architecture with physics-constrained loss functions or imposed symmetries. For more complicated problems (such as weather forecasting) that might benefit from enforcing physics within the data-driven architecture, see the comprehensive review by Kashinath et al. [46]. Note that other types of neural architectures can also be used to build the surrogate model, e.g., neural operator-based models, which have recently been shown to achieve state-of-the-art performance in weather forecasting [47].

2.4. Data assimilation with a stochastic EnKF

In this section, we describe DA with stochastic EnKF. Stochastic EnKF requires a dynamical model of the system, also called "background forecast model", represented by M. We further assume that an ensemble of noisy observations, ψ^j_{obs}, where j is the index of the ensemble member, is obtained by adding Gaussian white noise to the observations. Here, we assume that the observation noise distribution can be represented as a standard Gaussian with zero mean and standard deviation of σ_{obs}. In this paper, we have considered $\sigma_{obs} = 0.1$ (10% of standard deviation of $\psi_k(t)$, 1.0). Throughout the rest of the paper, we will drop the word "stochastic", assuming that an ensemble of noisy observations is available as opposed to a single noisy observation. M evolves an ensemble of state vectors ψ^j from t to $t+\Delta t$ starting from a noisy initial condition, $\psi(t_0)$. Let us assume that $j \in \{1, 2, \cdots n\}$, where n denotes the number of ensemble members. Here, and throughout the rest of the paper, we have dropped the subscript k (the index for the two layers in the system) for clarity unless it is necessary

(e.g., in Eq. (A.1) and Eq. (A.2)) keeping in mind that all computations take place on both layers. Furthermore, we assume that assimilation of observations takes place at every $\alpha\Delta t$ (in this paper, we choose $\alpha = 200$, i.e., DA occurs every day).

The ensemble of state vectors is generated by adding Gaussian white noise with zero mean and standard deviation of σ_b to the noisy state vector, $\psi(t_0)$, at initialization time, t_0:

$$\psi^j(t_0) = \psi(t_0) + \mathcal{N}\left(0, \sigma_b^2\right). \tag{4}$$

The value of σ_b has been chosen after significant trial and error for both EnKF and H-EnKF algorithms separately to obtain the best performance (see section 5). The evolution of the ensemble of state vectors, $\psi^j(t)$, at any time t, can be written in discrete form:

$$\psi^j(t+\alpha\Delta t) = \underbrace{M \circ M \circ \cdots M}_{\alpha}\left[\psi^j(t)\right]. \tag{5}$$

Then, we compute the background error covariance matrix using the ensembles that are obtained at $\alpha\Delta t$:

$$\begin{aligned}\mathbf{P} = \mathbb{E}\Big[&\left(\psi^j(t+\alpha\Delta t) - \overline{\psi}(t+\alpha\Delta t)\right)\\ &\left(\psi^j(t+\alpha\Delta t) - \overline{\psi}(t+\alpha\Delta t)\right)^T\Big],\end{aligned} \tag{6}$$

where $\overline{\psi}(t+\alpha\Delta t)$ is the mean over the n ensemble members and $\mathbb{E}$ is the sample expectation operator. The Kalman gain is computed using $\mathbf{P}$ as:

$$\mathbf{K} = \mathbf{P}\left(\mathbf{P} + \sigma_{obs}\mathbf{I}\right)^{-1}. \tag{7}$$

Here, the observation operator, H, is $\mathbf{I}$. However, we can extend Eq. (7) to nonlinear H as well. Finally, using the ensemble of noisy observations, $\psi^j_{obs}(t+\alpha\Delta t)$, the noise-reduced analysis state is computed as:

$$\begin{aligned}\psi^j_a(t+\alpha\Delta t) &= \psi^j(t+\alpha\Delta t) +\\ \mathbf{K}&\left(\psi^j_{obs}(t+\alpha\Delta t) - \psi^j(t+\alpha\Delta t)\right),\end{aligned} \tag{8}$$

where $\psi^j_a(t+\alpha\Delta t)$ is the j^{th} member of the ensemble of analysis states. $\overline{\psi_a}(t+\alpha\Delta t)$ is a noise-reduced estimate of the state of the system. Hence, $\overline{\psi_a}(t+\alpha\Delta t)$ is used as the initial condition by the dynamical model, M, for free forecasting. Moreover, the ensemble of analysis states, $\psi^j_a(t+\alpha\Delta t)$, can be used for probabilistic forecasting as well, see e.g., Holstein et al. [48].

3. Proposed DA algorithm: H-EnKF

The major challenge with using EnKF is computing the background error covariance matrix, $\mathbf{P}$, using Eq. (6). An accurate computation of $\mathbf{P}$ requires n to be large, typically the same order as that of the dimension of $\psi(t)$. However, this makes the evaluation of Eq. (5) computationally expensive. This is especially true for high-dimensional systems where M is an expensive numerical model. Hence, traditional applications of EnKF use small values of n, which induces sampling error in $\mathbf{P}$, resulting in spurious long-range correlations. In this section, we show how the U-NET-based surrogate model is used in developing H-EnKF. A schematic of H-EnKF is shown in Fig. 2.

We denote the U-NET-based surrogate model as M_D, which evolves the state $\psi(t)$ to $\psi(t+\gamma\Delta t)$ (in this paper, $\gamma = 40$). Since M_D is already trained, it is computationally inexpensive during inference. Hence, one can afford to evolve a very large number of ensemble members, $O(1000)$, with M_D. We denote the number of ensemble members that are evolved with M_D as n_D. Similarly, we denote the number of ensemble members evolved with the numerical model (M_N) as n_N. Note that for all practical problems involving high-dimensional systems, one can computationally afford a small number of n_N but a very large number of n_D ($n_N \ll n_D$). Similar to section 2.4, DA occurs at every $\alpha\Delta t$ and we assume that α is an integer multiple of γ.

In H-EnKF, we first evolve a large data-driven ensemble using M_D:

$$\psi^i_D(t+\alpha\Delta t) = \underbrace{M_D \circ M_D \circ \cdots M_D}_{\alpha/\gamma}\left[\psi^i_D(t)\right], \tag{9}$$

where $i \in \{1, 2, 3, \cdots n_D\}$. At the same time, we evolve a small number of numerical ensemble members using M_N:

$$\psi^j_N(t+\alpha\Delta t) = \underbrace{M_N \circ M_N \circ \cdots M_N}_{\alpha}\left[\psi^j_N(t)\right], \tag{10}$$

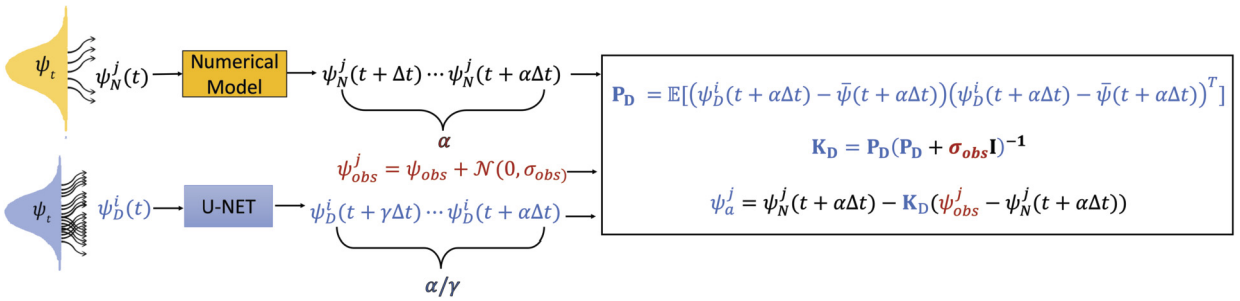

Fig. 2. Schematic of the H-EnKF framework. Noisy initial condition $\psi(t_0)$ is perturbed with $\mathcal{N}(0,\sigma_b^2)$ to generate two sets of ensembles: 1) $\psi_D^i(t)$ with n_D ensembles where $n_D \approx O(1000)$ and 2) $\psi_N^j(t)$ with n_N ensembles where $n_N \approx O(10)$. A pre-trained U-NET predicts the evolution of each of the n_D ensembles autoregressively for $\alpha\Delta t$. Similarly, the numerical solver for the QG system evolves the n_N ensembles of $\psi_N^j(t)$ for $\alpha\Delta t$. At this point, a noisy observation of ψ_{obs} is perturbed with $\mathcal{N}(0,\sigma_{obs}^2)$ to generate n_N ensemble members. Here, an EnKF algorithm computes the background covariance matrix $\mathbf{P_D}$ using the ensembles evolved with the U-NET and finally produces the analysis state ensembles ψ_a^j using the background forecast $\psi_N^j(t)$ from the numerical model. While the analysis ensemble members are carried forward and evolved by the numerical model, the U-NET needs the ensembles to be *restarted* by perturbing the ensemble-averaged analysis state using Gaussian noise (zero mean and σ_b standard deviation) generating n_D ensembles to be evolved over the next $\alpha\Delta t$. In this paper, we have chosen $\alpha = 200$ and $\gamma = 40$. Here, $40\Delta t \approx 6h$ and $200\Delta t \approx 1$ Earth day.

where $j \in \{1, 2, 3, \cdots n_N\}$. In both Eq. (9) and Eq. (10), the ensembles are generated with Gaussian white noise as shown in Eq. (4). At this point, we compute the background error covariance matrix, $\mathbf{P_D}$, using the n_D ensemble members evolved by M_D:

$$\mathbf{P_D} = \mathbb{E}\left[\left(\psi_D^i(t+\alpha\Delta t) - \overline{\psi}_D(t+\alpha\Delta t)\right)\left(\psi_D^i(t+\alpha\Delta t) - \overline{\psi}_D(t+\alpha\Delta t)\right)^T\right]. \tag{11}$$

Similar to Eq. (6), $\overline{\psi}_D(t+\alpha\Delta t)$ denotes the mean of $\psi_D^i(t+\alpha\Delta t)$ over n_D ensemble members. Since $n_N \ll n_D$, $\mathbf{P_D}$ calculated with n_D ensemble members would have much lower sampling error as compared to the one computed with n_N ensemble members. Then, we compute Kalman gain, $\mathbf{K_D}$, as:

$$\mathbf{K_D} = \mathbf{P_D}\left(\mathbf{P_D} + \sigma_{obs}\mathbf{I}\right)^{-1}. \tag{12}$$

Then we compute the ensemble of analysis states as:

$$\psi_a^j(t+\alpha\Delta t) = \psi_N^j(t+\alpha\Delta t) + \mathbf{K_D}\left(\psi_{obs}^j(t+\alpha\Delta t) - \psi_N^j(t+\alpha\Delta t)\right), \tag{13}$$

where $\psi_a^j(t+\alpha\Delta t)$ is the ensemble of analysis states. Note that in Eq. (13), we have used n_N ensemble members of the background forecast state which are more accurate (since M_N is a more accurate numerical model that integrates physical equations as compared to the data-driven model, M_D). However, $\mathbf{P_D}$ is obtained from the large number (n_D) of ensemble members from M_D to alleviate issues related to sampling error and spurious long-range correlations.

In H-EnKF, the n_N ensemble members (obtained as the analysis states) keep evolving for future DA cycles with the numerical model, as is done in a standard EnKF. However, after every DA cycle, we reinitialize the n_D ensemble members using Eq. (4), where $\psi(t_0)$ is replaced with $\overline{\psi_a}(t+\alpha\Delta t)$ for $t > t_0$. This is because the data-driven models' forecasts do not remain stable for long time scales [44,49,50].

It is important to note that, one can also use the data-driven model's prediction to compute the background forecast state, i.e., a fully data-driven model without the need for any numerical integration [32]. However, the quality of such forecasts would depend on how frequently we perform DA. If the frequency of DA is small, i.e., we evolve the data-driven model for a long period of time before performing DA, the quality of the data-driven forecasted state would degrade. That would lead to inaccuracies in the computation of the analysis state.

4. Metrics for measuring performance

We define two metrics for measuring the performance of a model: relative error ($E_k(t)$) and Anomaly Correlation Coefficient (ACC_k) for each layer, k. Details on computing these metrics are given in Appendix A.

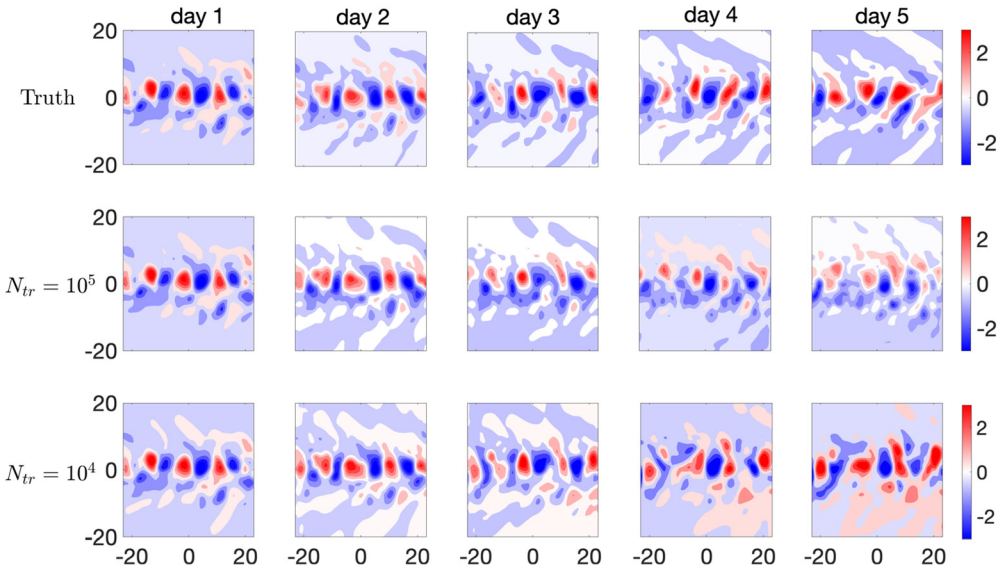

Fig. 3. Predicted patterns of time-mean removed $\psi_1(t)$ anomalies by the U-NET trained on $N_{tr} = 10^5$ and $N_{tr} = 10^4$ training samples as compared to the truth (obtained from numerical simulation). Note that only part of the latitudinal extent of the domain is shown.

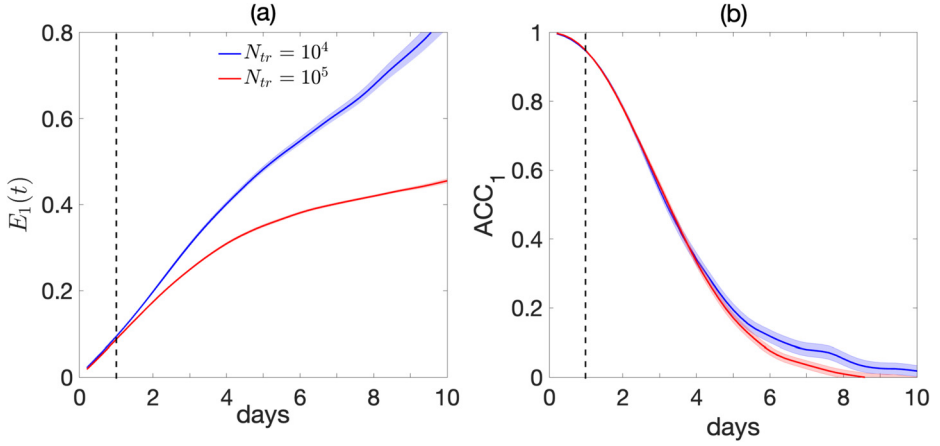

Fig. 4. Performance metrics of the U-NET trained on $N_{tr} = 10^5$ and $N_{tr} = 10^4$ samples. (a) $E_1(t)$, where the subscript "1" refers to the first layer. (b) ACC_1 is the ACC for the first layer. Note that similar results are found for $E_2(t)$ and ACC for 2nd layer, but are not shown for brevity. Shading shows standard deviation over 100 random noise-free initial conditions.

5. Results

5.1. Performance of U-NET for fully data-driven prediction

First, we show the performance of fully data-driven predictions with U-NET trained on N_{tr} samples of $\psi_k(t)$. Fig. 3 shows the predicted patterns of the time-mean removed anomalies of $\psi_1(t)$ with U-NET trained on $N_{tr} = 10^4$ and $N_{tr} = 10^5$ samples. Qualitatively good performance up to day 3 can be seen in Fig. 3 with both $N_{tr} = 10^4$ and $N_{tr} = 10^5$ samples.

We quantify the accuracy of the predicted ψ_1 using the metrics defined in section 4. Fig. 4 shows that $E_1(t)$ is 22.2% lower for the U-NET trained with $N_{tr} = 10^5$ samples as compared to the one trained with $N_{tr} = 10^4$ samples at day 3. ACC_1 in Fig. 4 shows that both U-NETs have roughly the same prediction horizon (time at which $ACC_1 \approx 0.60$).

For the H-EnKF framework, we need the U-NET to predict the states of the system for only 1 day. For 1-day prediction, both the relative error and ACC metrics of the U-NETs are not sensitive to N_{tr}. We have used an U-NET trained on $N_{tr} = 10^5$ samples as our data-driven model in the H-EnKF algorithm. However, using an U-NET trained on $N_{tr} = 10^4$ samples, we would obtain the same performance for the H-EnKF algorithm. We have also conducted experiments where an energy-constrained loss function was used to train the U-NET and found no significant improvement over the baseline performance in short-term forecasts (not shown for brevity).

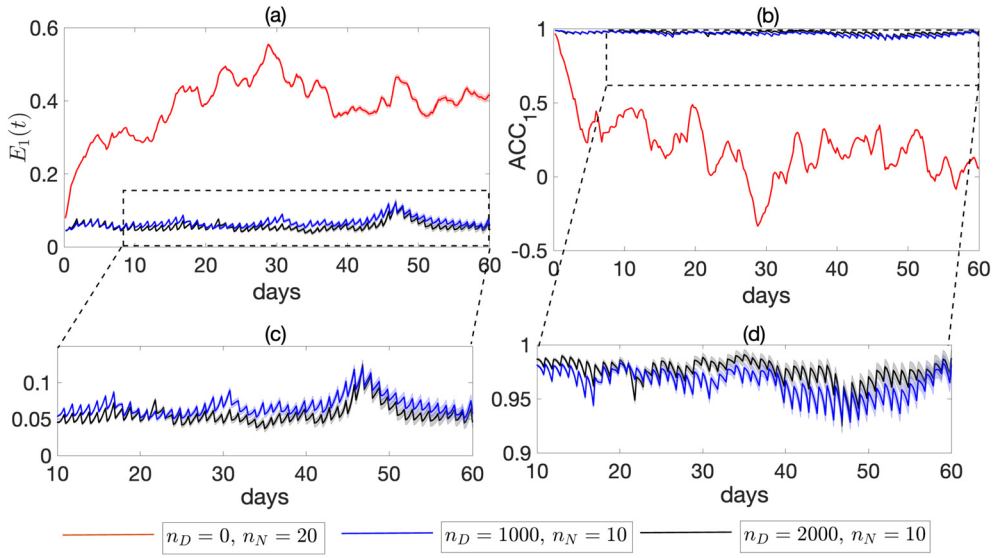

Fig. 5. Performance of H-EnKF and standard EnKF for $\psi_1(t)$ over 60 DA cycles. (a) $E_1(t)$ over 60 DA cycles for H-EnKF ($n_D = 2000$, $n_N = 10$), H-EnKF ($n_D = 1000$, $n_N = 10$), and EnKF ($n_D = 0$, $n_N = 20$). (b) Same as (a) but for ACC_1. (c) Zoomed in view of (a) between $0 \leq E_1(t) \leq 0.15$. (d) Zoomed in view of (b) between $0.90 \leq ACC_1 \leq 1.0$. Shading shows standard deviation over 30 random initial conditions. $\sigma_b = 0.80$ has been used for EnKF while $\sigma_b = 0.10$ has been used for the H-EnKF models. The σ_b value has been chosen to minimize the average $E_1(t)$ over 60 DA cycles based on extensive trial and error.

5.2. Performance of H-EnKF for DA

Next, we show the performance of H-EnKF as compared to standard EnKF for different values of n_D and n_N. For M_D in H-EnKF, we have used a U-NET trained on $N_{tr} = 10^5$ samples. For regular EnKF, we take the dynamical model to be the numerical solver of the QG system. As discussed in section 2.4, we have considered the observation noise, $\sigma_{obs} = 0.1$, to be 10% of the standard deviation (≈ 1.0) of $\psi_k(t)$. Note that the computational cost (based on the wall-clock time of execution, see section 5.5 for details) of evolving 1 numerical ensemble member ($n_N = 1$) of the state vectors $\psi_k(t)$ for one Δt is similar to evolving 200 data-driven ensemble members ($n_D = 200$) using the U-NET.

Fig. 5(a) and (c) show that the best performance is given by H-EnKF ($n_D = 2000$, $n_N = 10$) followed by H-EnKF ($n_D = 1000$, $n_N = 10$) and finally EnKF ($n_D = 0$, $n_N = 20$). Regular EnKF diverges due to the small number of ensemble members, a well known problem with EnKF in the absence of localization. Based on the computational cost analysis, EnKF ($n_D = 0$, $n_N = 20$) is as expensive as H-EnKF ($n_D = 2000$, $n_N = 10$) while the latter has $5\times$ smaller average $E_1(t)$ over 60 DA cycles. Moreover, H-EnKF ($n_D = 1000$, $n_N = 10$) is $0.75\times$ cheaper than EnKF ($n_D = 0$, $n_N = 20$) with $3\times$ smaller average $E_1(t)$ over 60 DA cycles.

A similar conclusion can be made from Fig. 5(b) and (d) where ACC_1 for H-EnKF algorithms remain ≈ 0.95 throughout the 60 DA cycles while EnKF ($n_D = 0$, $n_N = 20$) shows a rapid decrease in ACC_1 from the beginning of the DA cycles (when localization is not performed). For H-EnKF, $\sigma_b = 0.10$ has been used to obtain the best performance. We have conducted several trials with different values of σ_b and chose the value that minimized the average $E_1(t)$ over 60 DA cycles.

These results demonstrate the effectiveness of the H-EnKF framework in terms of estimating a better initial condition from noisy observations of the system. In this framework, one can trade off a small number of computationally expensive numerical ensemble members for a large number of cheap data-driven ensemble members to improve the accuracy of the estimated analysis states without affecting the overall computational cost.

It must be kept in mind that the performance of standard EnKF can be improved by simply increasing the number of numerical ensemble members or with localization. In our experiments with QG, we have seen that stable and divergence-free filters can be obtained with $O(1000)$ numerical ensemble members. Fig. B.9 in Appendix B shows the performance of EnKF over 10 DA cycles with $n_N = 1000$ and $n_N = 5000$. For standard EnKF, $\sigma_b = 0.80$ has been used to obtain the best performance. Similar to H-EnKF, we had conducted several trials to obtain the best σ_b, that minimized the average $E_1(t)$ over 10 DA cycles. However, evolving $O(1000)$ ensemble members with the numerical solver comes at $50\times$ more computational cost compared to H-EnKF ($n_D = 2000$, $n_N = 10$). Moreover, for other practical systems with higher dimensions, evolving such a large size of ensembles may not even be computationally tractable. We further demonstrate results with EnKF ($n_D = 0, n_N = 20$) with localization in Fig. C.10 (Appendix C), which shows that localization can also improve the performance of EnKF in this system without any increase in the computational cost. However, in more complex systems, localization can remove long-range physical correlations and affect the accuracy of the estimated analysis states [29]. Furthermore, for other types of ensemble-based DA algorithms, such as particle filters, one may not be able to perform localization [51,52]. Moreover, in particle filters, the large data-driven ensembles would provide an advantage in terms of sampling non-parametric and non-Gaussian distributions.

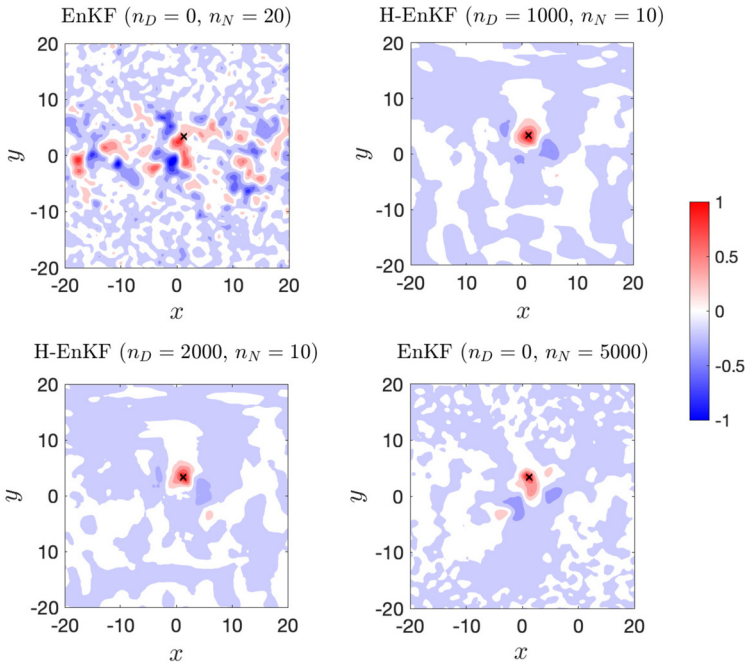

Fig. 6. The normalized background covariance matrix averaged over 60 DA cycles, except EnKF ($n_D = 0$, $n_N = 5000$) which is averaged over 10 DA cycles (due to limitations on computational cost). H-EnKF algorithms show more localized covariance structure and less spurious long-range correlation as compared to EnKF ($n_D = 0$, $n_N = 20$). The non-local structure of covariance of EnKF is due to low ensemble size leading to sampling error.

5.3. Analysis of covariance matrices

The superior performance of H-EnKF ($n_D = 1000$, $n_N = 10$) and H-EnKF ($n_D = 2000$, $n_N = 10$) over EnKF ($n_D = 0$, $n_N = 20$) is due to the improved representation of the background error covariance matrix, $\mathbf{P_D}$, which is calculated using the n_D data-driven ensembles. Fig. 6 shows that the 60 DA cycles-averaged covariance structure of grid point $x = 3.38$ and $y = 1.21$ is localized around that grid point for H-EnKF ($n_D = 1000$, $n_N = 10$) and H-EnKF ($n_D = 2000$, $n_N = 10$). However, for EnKF ($n_D = 0$, $n_N = 20$), the covariance structure is non-local with spurious long-range correlations across the domain. This is due to the reduction of sampling error during the computation of $\mathbf{P_D}$ in Eq. (11) using the n_D ensembles. We have also shown the 10 DA cycles-averaged covariance structure of EnKF ($n_D = 0$, $n_N = 5000$). Due to high computational cost, we have only performed 10 DA cycles with $n_N = 5000$ ensembles. For this problem (≈ 36000 states), one can consider this covariance matrix to be close to the true covariance. It is clear that the H-EnKF covariance matrices are similar to the true covariance but at significantly lower computational cost.

5.4. Performance of free prediction with the numerical model of QG

Next, we compare the free prediction performance of the numerical model for the QG system with initial conditions that are obtained as the mean of the analysis states after 60 DA cycles (see Fig. 5). In Fig. 7(a) and (b), we demonstrate that the initial condition obtained from H-EnKF ($n_D = 1000$, $n_N = 10$) has prediction skill up to 3.8 days, while H-EnKF ($n_D = 2000$, $n_N = 10$) shows prediction skill up to 4.5 days.

Fig. 7(c) and (d) show the free prediction of the numerical model for QG with initial conditions that are obtained as the mean of the analysis states after 10 DA cycles. Here, the initial condition obtained from H-EnKF ($n_D = 1000$, $n_N = 10$) has prediction skill up to 4.1 days and H-EnKF ($n_D = 2000$, $n_N = 10$) has prediction skill up to 5.8 days.

We have not shown the free prediction performance of the initial condition from the standard EnKF because the analysis states obtained from the algorithm have too large of an error, as can be seen in Fig. 5, leading to no prediction skill at all.

An important point to notice in Fig. 7 is the difference between the prediction skill of the initial condition obtained from H-EnKF ($n_D = 1000$, $n_N = 10$) and H-EnKF ($n_D = 2000$, $n_N = 10$). Although the initial conditions from both the models qualitatively seem to be very close, as is evident from inspecting $E_1(t)$ in Fig. 5, they result in significant difference in prediction skill (≈ 0.7 days for the initial condition at 60 DA cycles and 1.7 days for the initial condition at 10 DA cycles). This shows that increasing n_D (which is $200\times$ cheaper than n_N) to compute $\mathbf{P_D}$ leads to significant improvement in the quality of the analysis states which in turn leads to an improvement in the prediction skill of the numerical model.

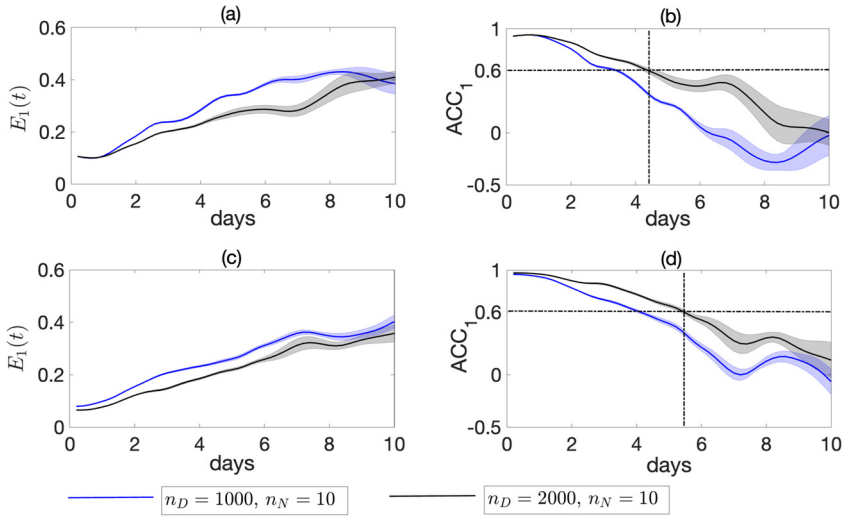

Fig. 7. Performance of free prediction of the QG numerical solver with initial condition (as mean analysis state) from H-EnKF. (a) $E_1(t)$ with initial condition at the end of 60 DA cycles. (b) ACC_1 with the same initial condition as (a). (c) $E_1(t)$ with initial condition at the end of 10 DA cycles. (d) ACC_1 with the same initial condition as (c). Shading shows standard deviation over 30 random initial conditions. Performance of initial condition from EnKF ($n_D = 0$, $n_N = 20$) is not shown because it has no prediction skill at all.

5.5. Computational cost analysis and scaling

In this section, we discuss how the mean error over 10 DA cycles, $\langle E_1(t)\rangle$, where $\langle . \rangle$ denotes mean over 10 DA cycles, scales with the computational cost associated with different EnKF and H-EnKF algorithms. To have a fair cost comparison, we have executed both the EnKF and H-EnKF algorithms on the same hardware. The computations of background error covariance matrix, Kalman gain, and analysis states have been performed on an AMD EPYC 7742 CPU with 64 cores for both EnKF and H-EnKF. The numerical model has been executed on the same CPU. However, the U-NET is run on a NVIDIA Tesla V100 GPU. We have not neglected the overhead cost of transferring data from GPU to CPU for H-EnKF. There are two things to note about the performance of the numerical simulation. First, one could have used a GPU-enabled numerical model as well, which would significantly improve its runtime performance. However, the overarching goal of this work is to facilitate DA in practical large-scale problems. In many practical problems, state-of-the-art simulation codes for fluid dynamics, combustion, weather prediction, etc., are CPU-based and would require enormous amounts of resources to refactor on GPUs. Second, one could have used a distributed-parallel numerical solver, which would also improve its runtime performance. However, one can also execute the U-NET on distributed GPUs as well, wherein the runtime performance gain due to distributed parallelism would be equivalent between the U-NET and the numerical solver. Owing to these careful observations and considerations, our computational cost analysis is fair.

Here, we define 1 computational cost unit as the "wall-clock" runtime of evolving $n_N = 1$ ensemble member with the dynamical model over one Δt. Experimentally, we find that the runtime of $n_N = 1$ ensemble member is the same as the runtime of $n_D = 200$ ensemble members with the U-NET model over one Δt. To obtain a robust measure of runtime for the U-NET for one Δt, we had run the U-NET for $200\Delta t$ and recorded the wall-clock time averaged over 100 independent runs as $T^D_{200\Delta t}$. From here, the runtime for the U-NET over $1\Delta t$ is obtained as $T^D_{200\Delta t}/200$. For the numerical solver, we had followed the same procedure to obtain the runtime for evolving one numerical ensemble member over one Δt as $T^N_{200\Delta t}/200$.

In Fig. 8(a), EnKF ($n_D = 0$, $n_N = 20$) shown with red circle has $\langle E_1(t)\rangle$ of 52.6% with computational cost of 20 units. H-EnKF ($n_D = 2000$, $n_N = 10$) shown with black circle has a factor of 5 smaller error at the same computational cost. By increasing the number of numerical ensemble members to $n_N = 90$ and keeping the same number of data-driven ensemble members ($n_D = 2000$), $\langle E_1(t)\rangle$ goes down by a factor of 1.07 but at 5 times higher computational cost. With a further increase in cost to 2000 units, $\langle E_1(t)\rangle$ of EnKF ($n_D = 0$, $n_N = 2000$) improves by a factor 1.5 in comparison to H-EnKF ($n_D = 2000$, $n_N = 10$). At the same cost, $\langle E_1(t)\rangle$ of H-EnKF ($n_D = 2000$, $n_N = 1990$) improves by a factor of 4.0. From this analysis, it is quite clear that trading a small number of numerical ensemble members for a large number of data-driven ones leads to a significant improvement in performance without an increase in computational cost.

Fig. 8(b) shows the scaling of $\langle E_1(t)\rangle$ for H-EnKF and EnKF having different levels of fixed computational cost. We see that the effect of the background forecast derived from n_N is more pronounced when n_N is large ($n_N = 2000$) (given by the black squares). Keeping the cost fixed at 2000 units, adding more n_D ($n_D = 0$ to $n_D = 2000$), decreases the error only very slightly (from 6.5% to 2.5%). For low n_N ($n_N = 100$ or $n_N = 20$), adding more n_D significantly improves the performance of H-EnKF, shown by the blue circles (from 29.5% to 9.46%) or red asterisk (from 48% to 9.40%). It must be noted that

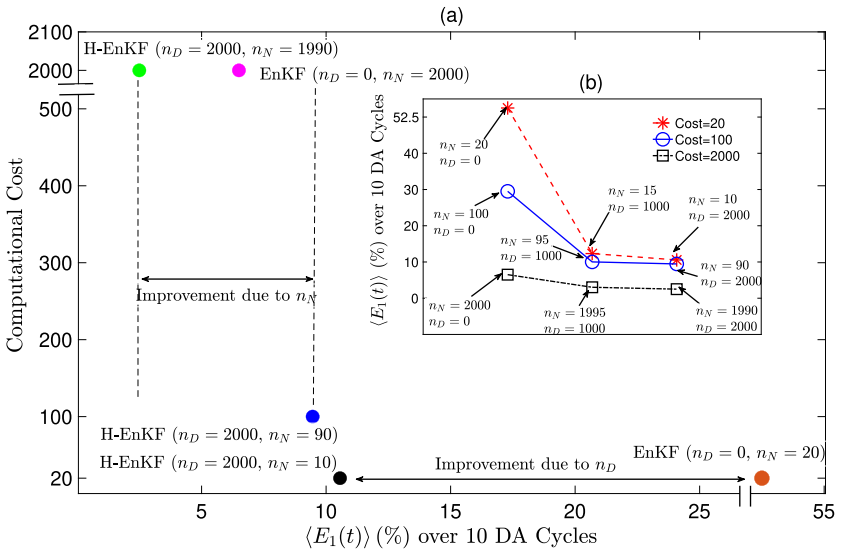

Fig. 8. Computational cost and accuracy scaling for H-EnKF and EnKF algorithms. Here, we have shown $\langle E_1(t)\rangle$, the mean of $E_1(t)$, over 10 DA cycles (instead of 60) because of the high computational cost associated with computing 60 DA cycles on EnKF models with large numerical ensembles ($O(1000)$). (a) $\langle E_1(t)\rangle$ of different H-EnKF and EnKF algorithms along with their computational cost. (b) Scaling of $\langle E_1(t)\rangle$ at different levels of fixed computational cost for different H-EnKF and EnKF models. The unit for computational cost is the runtime for the numerical QG solver to evolve one numerical ensemble for one Δt, $T^N_{200\Delta t}/200$. See section 5.5 for more details.

H-EnKF ($n_D = 2000$, $n_N = 10$) at 5× smaller cost performs almost equally well as H-EnKF ($n_D = 2000$, $n_N = 90$). A further improvement in the error (about 4×) is seen only with a 100× increase in computational cost.

6. Discussion and summary

In this paper, we have proposed H-EnKF: a hybrid ensemble Kalman filter algorithm that leverages a deep learning-based data-driven model to efficiently evolve a *large* ensemble of states of a dynamical model to better estimate the background error covariance with low sampling error. The ensemble of background forecast states used in the H-EnKF algorithm is obtained from an accurate, high-resolution, numerical solver evolving a *small* ensemble of states (small owing to high computational cost). This combination allows one to obtain an accurate DA algorithm at low computational cost alleviating one of the major sources of error –sampling error in covariance due to low ensemble size– without the need for localization.

It should be clarified that there is a major difference between how the numerical ensembles are evolved throughout the DA cycles as compared to the data-driven ensembles in H-EnKF. In H-EnKF, after every DA cycle, the data-driven ensembles are regenerated with Gaussian noise added to the mean of the analysis states. This is due to a limitation of current data-driven models for high-dimensional systems, which become unstable when evolved for a long period of time [44,50,49]. We acknowledge that in more complex systems, this approach can cause issues, as frequently perturbing the state can violate conservation laws, e.g., of mass (this is not a problem here as we are perturbing streamfunctions, ψ). In such problems, the perturbation should be added such that they do not violate the conservation laws, e.g., see Zeng et al. [53].

We have shown that the H-EnKF algorithm achieves a stable filter without localization and has a factor of 5 smaller error at the same computational cost as that of standard EnKF for the two-layer QG system (Fig. 5). We have further shown that H-EnKF is most useful for situations where one can only afford small numerical ensembles (Fig. 8). In such situations, one can trade off a small number of numerical ensemble members for a larger number of data-driven ones and obtain a more accurate covariance estimation. In H-EnKF applied to the two-layer QG system, the runtime for the evolution of the data-driven ensembles is a factor 200 smaller than the numerical ones. However, in more practical problems, such as weather prediction, the difference in runtime between evolving data-driven and numerical ensembles can be much larger, e.g., in FourCastNet [47], the data-driven model is 45000× faster than the state-of-the-art operational weather forecasting models. We expect the H-EnKF algorithm to be most effective for high-dimensional problems such as these as well as for problems in constrained DA. Another advantage of the H-EnKF is that, by eliminating or reducing the need for localization, it could more easily assimilate non-local observations such as satellite radiances. Such non-local observations pose a challenge for localized EnKFs, especially when using domain localization [54]. Moreover, it must also be kept in mind that while we may need the number of ensemble members generated to be the same as the number of states of the system to remove rank deficiency of the covariance matrix, practically, that is not required. In fact, as shown in multiple studies [55,26], if the generated ensembles are aligned with the unstable and neutral Lyapunov vectors of the linearized dynamical system, then one can achieve a stable filter with an ensemble size that equals the number of unstable and neutral Lyapunov vectors. The number of unstable and neutral Lyapunov vectors, in practical systems, while less than the number of states in the

system, can still be too high, such that evolving that many ensemble members with a numerical model remains intractable. However, they can be generated and evolved with data-driven models at a very low computational cost.

As a test case, the proposed hybrid framework has been applied to EnKF, a specific ensemble-based DA algorithm that is used with Gaussian observation noise. However, one can readily extend this hybrid framework to other types of novel extensions of regular EnKF, e.g., multi-model EnKF [56] or other ensemble-based DA algorithms, e.g., particle filters. Particle filters are especially useful for systems with strong non-Gaussian observation noise. However, due to large computational cost, it has been difficult to use in high-dimensional systems such as geophysical flows [57,51]. For future work, we aim to explore how the H-EnKF algorithm can be used in particle filter-based DA.

The EnKF algorithms with small ensemble members are currently used in weather prediction by leveraging ad-hoc techniques such as localization that removes spurious long-range correlations. Localization can also remove physical correlations in the flow fields as well, which would result in inaccuracies in the covariance [29]. Due to low computational cost during inference, deep learning-based surrogates can generate a large ensemble of states that can better approximate the true background covariance as compared to artificially obtained covariance with localization. However, for applications where localization works well, one can use localization on the covariance matrix obtained from the data-driven ensembles in our hybrid framework. In such scenarios, we can bring down the computational cost even further by evolving only a small number of data-driven ensembles (which are already quite inexpensive to evolve). Moreover, localization may be difficult to perform on other ensemble-based DA algorithms, e.g., particle filters. For such algorithms, our hybrid approach, which does not require localization, would be useful as well.

One of the key aspects of the H-EnKF framework is the accurate data-driven surrogate which needs to be built for every application. However, building a fully data-driven surrogate for high-dimensional realistic systems, e.g., weather and climate, might appear to be challenging. However, recent successes in data-driven short-term weather modeling [58,32,47] show that short-term accurate surrogates can be as accurate as operational numerical models if trained on observations [47]. Further improvement in data-driven surrogates can be achieved by conserving key physics as can be seen in many different applications in both fluids [59] and weather and climate [46] communities.

CRediT authorship contribution statement

AC and PH formulated the problem. AC and EN wrote the codes and performed the analysis. All the authors discussed and analyzed the results. All authors contributed to the writing of the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We have already shared all our codes and data in the manuscript.

Acknowledgements

We thank Matti Morzfeld and Yonquiang Sun for insightful comments and discussions. This work was supported by an award from the ONR Young Investigator Program (N00014-20-1-2722), a grant from the NSF CSSI program (OAC-2005123), and NASA grant 80NSSC17K0266 to P.H. Computational resources were provided by NSF XSEDE (allocation ATM170020) to use Bridges GPU and the Rice University Center for Research Computing. The codes for H-EnKF are publicly available at https://github.com/ashesh6810/Hybrid-Ensemble-Kalman-Filter.

Appendix A. Equations for error metrics

We define relative error as:

$$E_k(t) = \frac{||\psi_k^{pred}(t) - \psi_k^{true}(t)||_2}{max\left(\psi_k^{true}(t)\right)}. \tag{A.1}$$

ψ_k^{pred} is the predicted streamfunction and ψ_k^{true} is the true streamfunction obtained from numerical simulations.

We define ACC_k as:

$$ACC_k = \frac{\Sigma_m \Sigma_p \left(\left(\psi_{k,m,p}^{pred}(t) - \left\langle \psi_{k,m,p}^{true} \right\rangle\right) \times \left(\psi_{k,m,p}^{true}(t) - \left\langle \psi_{k,m,p}^{true} \right\rangle\right)\right)}{\sqrt{\left(\Sigma_m \Sigma_p \left(\psi_{k,m,p}^{pred}(t) - \left\langle \psi_{k,m,p}^{true} \right\rangle\right)^2 \times \Sigma_m \Sigma_p \left(\psi_{k,m,p}^{true}(t) - \left\langle \psi_{k,m,p}^{true} \right\rangle\right)^2\right)}}, \tag{A.2}$$

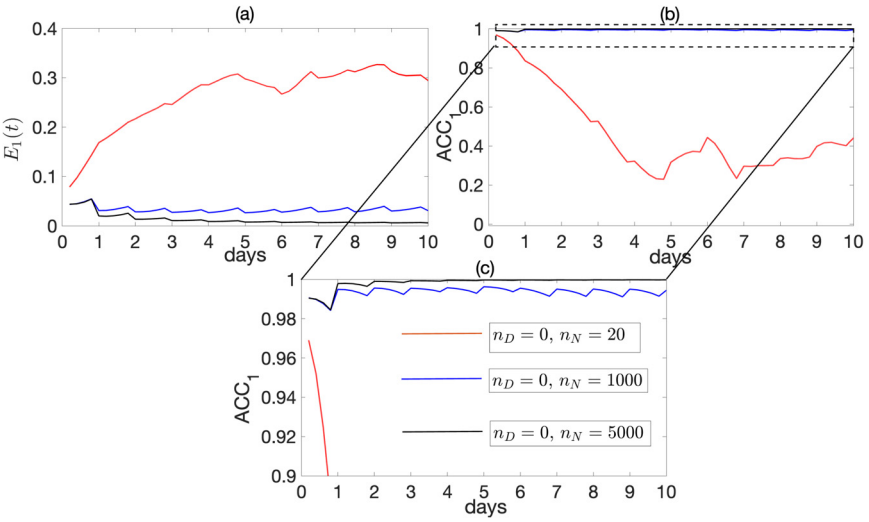

Fig. B.9. Performance of EnKF for $\psi_1(t)$ over 10 DA cycles with large n_N. Due to the computational cost of evolving large ensembles, we only report $E_1(t)$ and ACC_1. (a) $E_1(t)$ for EnKF ($n_D = 0$, $n_N = 20$), EnKF ($n_D = 0$, $n_N = 1000$), and EnKF ($n_D = 0$, $n_N = 5000$). (b) Same as (a) but for the ACC_1. (c) Zoomed in view of (b) between $0.90 \leq ACC_1 \leq 1.0$. Shading shows standard deviation over 30 initial conditions. $\sigma_b = 0.80$ has been used for all the EnKF algorithms. The σ_b value has been chosen to minimize the average $E_1(t)$ over 10 DA cycles based on extensive trial and error.

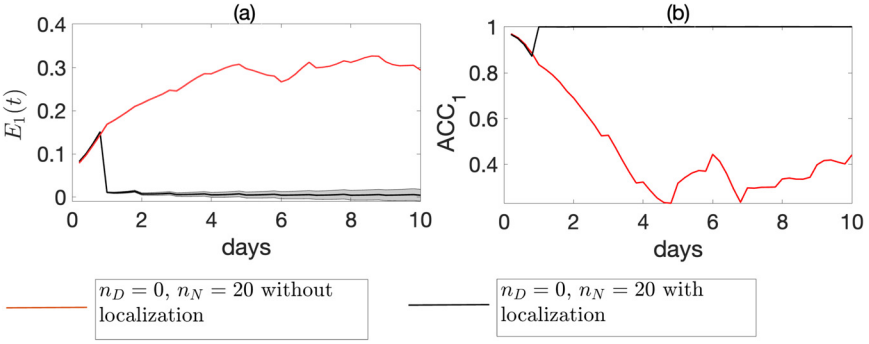

Fig. C.10. Performance of EnKF for $\psi_1(t)$ over 10 DA cycles with localization. (a) $E_1(t)$ over 10 DA cycles for EnKF ($n_D = 0$, $n_N = 20$) and EnKF ($n_D = 0$, $n_N = 20$) with localization. (b) Same as (a) but for ACC_1. Shading shows standard deviation over 30 initial conditions. $\sigma_b = 0.80$ has been used for both the EnKF algorithms. The σ_b value has been chosen to minimize the average $E_1(t)$ over 10 DA cycles based on extensive trial and error.

where $\left\langle \psi^{true}_{k,m,p} \right\rangle$ is the time-averaged value of $\psi^{true}_{k,m,p}(t)$ and the indices m and p refer to the latitudinal and longitudinal grid points on which $\psi_k(t)$ is represented. Similar to $E_k(t)$, ACC is also computed for each layer.

Appendix B. EnKF with large numerical ensembles

Here, we show that EnKF with a large number of numerical ensemble members (n_N) can have stable and divergence-free filters with low $E_1(t)$. However, in order to have performance comparable to H-EnKF, one needs to have $O(1000)$ n_N which comes at a high computational cost. Fig. B.9 shows that as n_N increases, the performance of EnKF improves as well. This is expected since with large ensembles, the matrix, **P**, would not suffer from sampling error and spurious correlations, as shown in Fig. 6 as well.

Appendix C. EnKF with small numerical ensembles and localization

In Fig. C.10, we report $E_1(t)$ and ACC_1 for EnKF ($n_D = 0, n_N = 20$) with and without localization. Here, we have used covariance localization [3]. The Gaspari-Cohn function is used to generate the regularizing correlation function. A radius of $5l$, where l is the L_2 distance between two consecutive grid points, has been used as the radius in the Gaspari-Cohn function. In this system, localization is an effective method to obtain divergence-free filters with similar performance as H-EnKF ($n_D = 2000$, $n_N = 20$). However, it must be noted that localization often removes long-range physical correlations as well and is required to be tuned for each application in more complex systems [29].

References

[1] E. Kalnay, Atmospheric Modeling, Data Assimilation and Predictability, Cambridge University Press, 2003.
[2] K. Law, A. Stuart, K. Zygalakis, Data Assimilation, Springer, 2015.
[3] M. Asch, M. Bocquet, M. Nodet, Data Assimilation: Methods, Algorithms, and Applications, SIAM, 2016.
[4] M. Morzfeld, J. Adams, S. Lunderman, R. Orozco, Feature-based data assimilation in geophysics, Nonlinear Process. Geophys. 25 (2) (2018) 355–374.
[5] M. Morzfeld, D. Hodyss, Gaussian approximations in filters and smoothers for data assimilation, Tellus, Ser. A Dyn. Meteorol. Oceanogr. 71 (1) (2019) 1600344.
[6] E.N. Lorenz, Predictability: a problem partly solved, in: Proc. Seminar on Predictability, vol. 1, 1996, pp. 1–18.
[7] A. Carrassi, M. Bocquet, L. Bertino, G. Evensen, Data assimilation in the geosciences: an overview of methods, issues, and perspectives, Wiley Interdiscip. Rev.: Clim. Change 9 (5) (2018) e535.
[8] N. Chen, S. Fu, G.E. Manucharyan, An efficient and statistically accurate Lagrangian data assimilation algorithm with applications to discrete element sea ice models, J. Comput. Phys. 455 (2022) 111000.
[9] J. Eliashiv, A.C. Subramanian, A.J. Miller, Tropical climate variability in the community earth system model: data assimilation research testbed, Clim. Dyn. 54 (1) (2020) 793–806.
[10] T. Gleiter, T. Janjić, N. Chen, Ensemble Kalman filter based data assimilation for tropical waves in the MJO skeleton model, Q. J. R. Meteorol. Soc. 148 (743) (2022) 1035–1056.
[11] K. Belyaev, A. Kuleshov, N. Tuchkova, C.A. Tanajura, An optimal data assimilation method and its application to the numerical simulation of the ocean dynamics, Math. Comput. Model. Dyn. Syst. 24 (1) (2018) 12–25.
[12] L. D'Amore, R. Arcucci, L. Marcellino, A. Murli, HPC computation issues of the incremental 3D variational data assimilation scheme in OceanVar software, J. Numer. Anal. Ind. Appl. Math. 7 (3–4) (2012) 91–105.
[13] R. Arcucci, L. Mottet, C. Pain, Y.-K. Guo, Optimal reduced space for variational data assimilation, J. Comput. Phys. 379 (2019) 51–69.
[14] T. Yi, E.J. Gutmark, Online prediction of the onset of combustion instability based on the computation of damping ratios, J. Sound Vib. 310 (1–2) (2008) 442–447.
[15] J. Bell, M. Day, J. Goodman, R. Grout, M. Morzfeld, A Bayesian approach to calibrating hydrogen flame kinetics using many experiments and parameters, Combust. Flame 205 (2019) 305–315.
[16] M.L. Croci, U. Sengupta, M.P. Juniper, Data assimilation using heteroscedastic Bayesian neural network ensembles for reduced-order flame models, in: International Conference on Computational Science, Springer, 2021, pp. 408–419.
[17] S.L. Brunton, J. Nathan Kutz, K. Manohar, A.Y. Aravkin, K. Morgansen, J. Klemisch, N. Goebel, J. Buttrick, J. Poskin, A.W. Blom-Schieber, T. Hogan, D. McDonald, Data-driven aerospace engineering: reframing the industry with machine learning, AIAA J. 59 (8) (2021) 2820–2847.
[18] Y. Liu, H.V. Gupta, Uncertainty in hydrologic modeling: Toward an integrated data assimilation framework, Water Resour. Res. 43 (7) (2007), https://doi.org/10.1029/2006WR005756.
[19] Y. Maday, A.T. Patera, J.D. Penn, M. Yano, A parameterized-background data-weak approach to variational data assimilation: formulation, analysis, and application to acoustics, Int. J. Numer. Methods Eng. 102 (5) (2015) 933–965.
[20] Y. Gu, D.S. Oliver, An iterative ensemble Kalman filter for multiphase fluid flow data assimilation, SPE J. 12 (04) (2007) 438–446.
[21] M. Habibi, R.M. D'Souza, S.T. Dawson, A. Arzani, Integrating multi-fidelity blood flow data with reduced-order data assimilation, Comput. Biol. Med. (2021) 104566.
[22] G. Evensen, Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys. Res., Oceans 99 (C5) (1994) 10143–10162.
[23] S.J. Fletcher, Data Assimilation for the Geosciences: From Theory to Application, Elsevier, 2017.
[24] R. Bannister, A review of operational methods of variational and ensemble-variational data assimilation, Q. J. R. Meteorol. Soc. 143 (703) (2017) 607–633.
[25] K. Kondo, T. Miyoshi, Impact of removing covariance localization in an ensemble Kalman filter: experiments with 10 240 members using an intermediate AGCM, Mon. Weather Rev. 144 (12) (2016) 4849–4865.
[26] A. Carrassi, M. Bocquet, J. Demaeyer, C. Grudzien, P. Raanes, S. Vannitsem, Data assimilation for chaotic dynamics, in: Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications (vol. IV), 2022, pp. 1–42.
[27] L. De Cruz, S. Schubert, J. Demaeyer, V. Lucarini, S. Vannitsem, Exploring the Lyapunov instability properties of high-dimensional atmospheric and climate models, Nonlinear Process. Geophys. 25 (2) (2018) 387–412.
[28] M. Leutbecher, Ensemble size: how suboptimal is less than infinity?, Q. J. R. Meteorol. Soc. 145 (2019) 107–128.
[29] T. Miyoshi, K. Kondo, T. Imamura, The 10,240-member ensemble Kalman filtering with an intermediate AGCM, Geophys. Res. Lett. 41 (14) (2014) 5264–5271.
[30] T. Janjic, M. Lukacova, Y. Ruckstuhl, P. Spichtinger, B. Wiebe, A test of an alternative approach for uncertainty representation in weather forecasting, in: EGU General Assembly Conference Abstracts, 2021, pp. EGU21–13077.
[31] L.M. Yang, I. Grooms, Machine learning techniques to construct patched analog ensembles for data assimilation, J. Comput. Phys. 443 (2021) 110532.
[32] A. Chattopadhyay, M. Mustafa, P. Hassanzadeh, E. Bach, K. Kashinath, Towards physics-inspired data-driven weather forecasting: integrating data assimilation with a deep spatial-transformer-based U-NET in a case study with ERA5, Geosci. Model Dev. 15 (5) (2022) 2221–2237.
[33] T. Tsuyuki, R. Tamura, Nonlinear data assimilation by deep learning embedded in an ensemble Kalman filter, J. Meteorol. Soc. Jpn. II, https://doi.org/10.2151/jmsj.2022-027.
[34] R. Maulik, V. Rao, J. Wang, G. Mengaldo, E. Constantinescu, B. Lusch, P. Balaprakash, I. Foster, R. Kotamarthi, Aieada 1.0: efficient high-dimensional variational data assimilation with machine-learned reduced-order models, Geosci. Model Dev. Discuss. (2022) 1–20.
[35] S.G. Penny, T.A. Smith, T.-C. Chen, J.A. Platt, H.-Y. Lin, M. Goodliff, H.D. Abarbanel, Integrating recurrent neural networks with data assimilation for scalable data-driven state estimation, J. Adv. Model. Earth Syst. 14 (3) (2022) e2021MS002843.
[36] N. Chen, Y. Li, BAMCAFE: a Bayesian machine learning advanced forecast ensemble method for complex turbulent systems with partial observations, Chaos, Interdiscip. J. Nonlinear Sci. 31 (11) (2021) 113114.
[37] J. Brajard, A. Carrassi, M. Bocquet, L. Bertino, Combining data assimilation and machine learning to infer unresolved scale parametrization, Philos. Trans. R. Soc. A 379 (2194) (2021) 20200086.
[38] S. Pawar, S.E. Ahmed, O. San, A. Rasheed, I.M. Navon, Long short-term memory embedded nudging schemes for nonlinear data assimilation of geophysical flows, Phys. Fluids 32 (7) (2020) 076606.
[39] R. Mojgani, A. Chattopadhyay, P. Hassanzadeh, Discovery of interpretable structural model errors by combining Bayesian sparse regression and data assimilation: a chaotic Kuramoto-Sivashinsky test case, arXiv preprint, arXiv:2110.00546.
[40] N.J. Lutsko, I.M. Held, P. Zurita-Gotor, Applying the fluctuation–dissipation theorem to a two-layer model of quasigeostrophic turbulence, J. Atmos. Sci. 72 (8) (2015) 3161–3177.
[41] E. Nabizadeh, P. Hassanzadeh, D. Yang, E.A. Barnes, Size of the atmospheric blocking events: scaling law and response to climate change, Geophys. Res. Lett. 46 (22) (2019) 13488–13499.

[42] O. Ronneberger, P. Fischer, T. Brox, U-NET: convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.
[43] R. Wang, K. Kashinath, M. Mustafa, A. Albert, R. Yu, Towards physics-informed deep learning for turbulent flow prediction, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 1457–1466.
[44] A. Chattopadhyay, M. Mustafa, P. Hassanzadeh, K. Kashinath, Deep spatial transformers for autoregressive data-driven forecasting of geophysical turbulence, in: Proceedings of the 10th International Conference on Climate Informatics, 2020, pp. 106–112.
[45] R. Wang, R. Walters, R. Yu, Incorporating symmetry into deep dynamics models for improved generalization, arXiv preprint, arXiv:2002.03061.
[46] K. Kashinath, M. Mustafa, A. Albert, J. Wu, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, A. Manepalli, D. Chirila, R. Yu, R. Walters, B. White, H. Xiao, A. Tchelepi, P. Marcus, A. Anandkumar, P. Hassanzadeh, Prabhat, Physics-informed machine learning: case studies for weather and climate modelling, Philos. Trans. R. Soc. A 379 (2194) (2021) 20200093.
[47] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, Thorsten Kurth, D. Hall, Z. Li, K. Azizzadenesheli, P. Hassanzadeh, K. Kashinath, A. Anandkumar, FourCastNet: a global data-driven high-resolution weather model using adaptive Fourier neural operators, arXiv preprint, arXiv:2202.11214.
[48] C.-A.S. Stael von Holstein, An experiment in probabilistic weather forecasting, J. Appl. Meteorol. Climatol. 10 (4) (1971) 635–645.
[49] R. Keisler, Forecasting global weather with graph neural networks, arXiv preprint, arXiv:2202.07575.
[50] A. Chattopadhyay, J. Pathak, E. Nabizadeh, W. Bhimji, P. Hassanzadeh, Long-term stability and generalization of observationally-constrained stochastic data-driven models for geophysical turbulence, arXiv preprint, arXiv:2205.04601.
[51] P. Fearnhead, H.R. Künsch, Particle filters and data assimilation, Annu. Rev. Stat. Appl. 5 (2018) 421–449.
[52] P.J. Van Leeuwen, H.R. Künsch, L. Nerger, R. Potthast, S. Reich, Particle filters for high-dimensional geoscience applications: a review, Q. J. R. Meteorol. Soc. 145 (723) (2019) 2335–2365.
[53] Y. Zeng, T. Janjić, Study of conservation laws with the local ensemble transform Kalman filter, Q. J. R. Meteorol. Soc. 142 (699) (2016) 2359–2372.
[54] A. Farchi, M. Bocquet, On the efficiency of covariance localisation of the ensemble Kalman filter using augmented ensembles, Front. Appl. Math. Stat. 5 (2019) 3.
[55] G.-H. Crystalng, D. Mclaughlin, D. Entekhabi, A. Ahanin, The role of model dynamics in ensemble Kalman filter performance for chaotic systems, Tellus, Ser. A Dyn. Meteorol. Oceanogr. 63 (5) (2011) 958–977.
[56] E. Bach, M. Ghil, A multi-model ensemble Kalman filter for data assimilation and forecasting, arXiv preprint, arXiv:2202.02272.
[57] E.A. Wan, R. Van Der Merwe, S. Haykin, The unscented Kalman filter, Kalman filtering and neural networks 5 (2007) (2001) 221–280.
[58] J.A. Weyn, D.R. Durran, R. Caruana, Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere, J. Adv. Model. Earth Syst. 12 (9) (2020) e2020MS002109.
[59] Y. Guan, A. Subel, A. Chattopadhyay, P. Hassanzadeh, Learning physics-constrained subgrid-scale closures in the small-data regime for stable and accurate LES, Phys. D: Nonlinear Phenom. (2022) 133568.