Journal of the
Meteorological
Society of
Japan

The DOI for this manuscript is
DOI:10.2151/jmsj.2022-027
J-STAGE Advance published date: February 22nd, 2022
The final manuscript after publication will replace the preliminary version at the above DOI once it is available.

Nonlinear Data Assimilation by Deep Learning Embedded in an Ensemble Kalman Filter

Tadashi TSUYUKI[1]

Meteorological Research Institute
Japan Meteorological Agency, Tsukuba, Japan

and

Ryosuke TAMURA

Research Institute for Sustainable Humanosphere
Kyoto University, Kyoto, Japan

Submitted: September 7, 2021
Revised: December 6, 2021
Finalized: February 5, 2022

----

[1] Corresponding author: Tadashi Tsuyuki, Observation and Data Assimilation Research Department, Meteorological Research Institute, 1-1 Nagamine, Tsukuba, 305-0052 JAPAN.
Email: ttuyuki@mri-jma.go.jp
Tel: +81-29-853-8642
Fax: +81-29-853-8649

Abstract

Recent progress in the particle filter has made it possible to use it for nonlinear or non-Gaussian data assimilation in high-dimensional systems, but a relatively large ensemble is still needed to outperform the ensemble Kalman filter (EnKF) in terms of accuracy. An alternative ensemble data assimilation method based on deep learning is presented, in which deep neural networks are locally embedded in the EnKF. This method is named the deep learning-ensemble Kalman filter (DL-EnKF). The DL-EnKF analysis ensemble is generated from the DL-EnKF analysis and the EnKF analysis deviation ensemble. The performance of the DL-EnKF is investigated through data assimilation experiments in both perfect and imperfect model scenarios using three versions of the Lorenz 96 model and a deterministic EnKF with an ensemble size of 10. Nonlinearity in data assimilation is controlled by changing the time interval between observations. Results demonstrate that despite such a small ensemble the DL-EnKF is superior to the EnKF in terms of accuracy in strongly nonlinear regimes and that the DL-EnKF analysis is more accurate than the output of deep learning due to positive feedback in assimilation cycles. Even if the target of training is an EnKF analysis with a large ensemble or a simulation by an imperfect model, the improvement introduced by the DL-EnKF is not very different from the case where the target of training is the true state.

Keywords data assimilation; deep learning; ensemble Kalman filter; particle filter

1. Introduction

Data assimilation in nonlinear or non-Gaussian systems has been a challenge in meteorology and other geosciences (Bocquet et al., 2010). For instance, it is well known that cumulus convection exhibits strong non-Gaussianity in data assimilation (e.g., Kondo and Miyoshi, 2019; Kawabata and Ueno, 2020). The ensemble Kalman filter (EnKF) is formulated under the Gaussian assumption and is close to optimal in weakly nonlinear regimes, but it does not work well if nonlinearity is strong. On the other hand, the particle filter (PF) does not need the Gaussian assumption, but the weight degeneracy had been preventing the use of the PF for high-dimensional data assimilation (Snyder et al., 2008; van Leeuwen 2009). However, this limitation is disappearing due to recent developments in the PF, including the use of localization and hybrids with the EnKF (Farchi and Bosquet, 2018; van Leeuwen et al., 2019). Despite this progress, a relatively large ensemble is still needed for the PF to outperform the EnKF (e.g., Penny and Miyoshi, 2016). This may be plausible since non-Gaussian data assimilation needs some information on higher-order moments of probability density functions (PDFs). As for the 4-dimensional variational method (4D-Var), Tsuyuki (2014) showed that the 4D-Var with a conventional cost function implicitly used a non-Gaussian prior PDF that evolved according to the Liouville equation (Ehrendorfer, 1994) if a certain condition was satisfied, and that the difficulty caused by multiple minima could be alleviated by combining with the EnKF. The iterative ensemble Kalman filter/smoother (IEnKF/IEnKS) have been shown to be the missing link between the PF and the EnKF and

4D-Var, and can work very well with mild nonlinearity and generate a much better analysis than the above data assimilation methods (Sakov et al., 2012; Bocquet and Sakov, 2013, 2014; Bocquet, 2016). However, the IEnKF/IEnKS need much larger computational cost due to the iterative application of the EnKF/EnKS and the use of a long assimilation window.

Recent developments in machine learning, in particular in deep learning (Le cum et al., 2015), have demonstrated impressive skills in various fields. Data-driven modeling, including data-driven parametrizations, based on machine learning has been extensively explored for improving simulations and predictions of nonlinear dynamical systems. Dueben and Bauer (2018) discussed the question of whether models that were based on deep learning and trained on atmospheric data could compete with weather and climate models that were based on physical principles. Reichstein et al. (2019) advocated a hybrid modeling approach in which physical process models were coupled with machine learning to further improve understanding and predictive ability in earth system science. Abarbanel et al. (2018) and Geer (2020) showed an equivalence in formulation between data assimilation and deep learning. Lists of literature of recent studies are available in Reichstein et al. (2019) and Chattopadhyay et al. (2020), for instance. Quite recently the combination of data assimilation and machine learning has been explored to address sparse and noisy observations in data-driven modeling (Brajard et al., 2020a; Bocquet et al., 2020; Tomizawa and Sawada, 2020; Gottwald and Reich, 2021; Wikner et al., 2021), data-driven parametrizations (Brajard et al., 2020b), and model error correction (Farchi et al., 2021).

Research on the application of deep learning to data assimilation itself has also started. Arcucci et al. (2021) proposed a method for integrating variational data assimilation with deep learning, in which a recurrent neural network is trained on the state of a dynamical model and the result of data assimilation. Silva et al. (2021) proposed the use of a generative adversarial network to make prediction and to assimilate observations by using a low-dimensional space for the spatial distribution of the simulated state. However, it is difficult to directly apply those methods to data assimilation in high-dimensional systems such as atmospheric models for numerical weather prediction.

In this study, we present an ensemble data assimilation method combining the EnKF and deep learning as an alternative to the PF for high-dimensional systems. The additional computational cost to assimilate observations is a very small fraction of that of the EnKF. Since a deep neural network (DNN) can learn a data assimilation method for a specific dynamical system and a specific observing system by training, we could expect this method to work with a relatively small ensemble size even in strongly nonlinear regimes. However, data assimilation in meteorology is generally a large-scale problem, and the background error covariance and the distribution of radar and satellite data change with the analysis time. The EnKF, as well as the PF and 4D-Var, can properly deal with this nonstationarity in data assimilation. On the other hand, deep learning is based on the minimization of the sum of errors over many samples. In addition, it would be difficult to provide sufficient information on the forecast error covariance to a DNN, because the feasible size of a DNN is limited, where we define the size of a DNN as the total number of weights including bias parameters to be optimized by training. If the output of a DNN is not well optimized for each analysis time, the analysis accuracy may deteriorate in assimilation cycles. However, since the EnKF does not work very well in strongly nonlinear regimes, we could expect data assimilation by deep learning to outperform the EnKF in such regimes.

The purpose of this study is to propose a nonlinear data assimilation method based on deep learning that is locally embedded in an EnKF and to investigate its performance through data assimilation experiments in both perfect and imperfect model scenarios using toy models. By applying deep learning in combination with an EnKF, we can reduce the size of a DNN and address the nonstationarity in data assimilation. This method is named the deep learning-ensemble Kalman filter (DL-EnKF).

The remainder of this paper is organized as follows. Section 2 introduces the method of DL-EnKF. Section3 describes the design of experiments in both perfect and imperfect model scenarios. Section 4 presents the results of these experiments. Summary and discussion are mentioned in Section 5.

2. Method

Since data assimilation is generally a large-scale problem, it is desirable to keep the size of a DNN as small as possible. For instance, the size of a feedforward neural network with *m* layers with *n* nodes per layer is about $n^2(m\text{-}1)$ and a greater number of samples would be required for training. If we directly apply deep learning to data assimilation, the number of input nodes is at least the sum of the number of observations and the degrees of freedom of a numerical model, and the number of output nodes is the degrees of freedom of the model, while the number of nodes of a hidden layer is usually required to be larger than the number of input or output nodes. For high-dimensional systems such as atmospheric models, the size of a DNN would become too large to be stored in the memory of a computer and to prepare sufficient training samples. To apply deep learning to data assimilation for atmospheric models, we need to introduce a localization procedure and to train a DNN to have some versatility so that it is applicable to each grid point in a certain range of geographical areas.

Fig. 1

Figure 1a shows the workflow of the DL-EnKF, in which deep learning is locally embedded in an EnKF. The “EnKF” box in this figure represents the analysis step of the EnKF, and “Deep Learning” box consists of an ensemble of several DNNs. The inputs of DNNs to create the DL-EnKF analysis at a grid point are the EnKF analysis, forecast, observations, availability of observations in binary, and pseudo-observations that supplement missing observations. The EnKF analysis and forecast are the ensemble means of each ensemble. We do not explicitly use the information contained in the forecast ensemble other than the ensemble mean to reduce the size of DNNs. Since observational data for which the DNN has input nodes may be sometimes missing, it is necessary to provide the information on the availability of observations to DNNs. The pseudo-observations are prepared by using the EnKF analysis and the observation operators of the missing observations. Those input data are extracted from a small domain centered at the analysis grid point. The radius of this domain is hereafter referred to as the input radius, and it is assumed that this value is smaller than the covariance localization radius of the EnKF. According to Hsieh and Tang (1998), the DL-EnKF analysis is the average of outputs from the ensemble of DNNs. The individual outputs from DNNs would be scattered in phase space due to multiple minima of a loss function of deep learning, and we would likely obtain a better DL-EnKF analysis by averaging those individual outputs.

The analysis ensemble $\{\boldsymbol{x}_m^a\}_{m=1}^M$, where M is the ensemble size, is created by modifying the EnKF analysis ensemble $\left\{\boldsymbol{x}_{\mathrm{EnKF},\ m}^a\right\}_{m=1}^M$ such that its ensemble mean is equal to the DL-EnKF analysis $\boldsymbol{x}_{\mathrm{DL-EnKF}}^a$ as follows:

$$\boldsymbol{x}_m^a = \boldsymbol{x}_{\mathrm{DL-EnKF}}^a + \alpha\left(\boldsymbol{x}_{\mathrm{EnKF},\ m}^a - \ \boldsymbol{x}_{\mathrm{EnKF}}^a\right), \quad (1)$$

where $\boldsymbol{x}_{\mathrm{EnKF}}^a$ is the EnKF analysis and α is a parameter for adjusting the spread of the analysis ensemble. If adaptive covariance inflation is used in the EnKF, we can set α to 1 since the effect of tuning α is almost canceled by this procedure. However, if we conduct ensemble forecasts using the analysis ensemble, we may need to adjust the value of α. The members of the analysis ensemble thus generated are evolved by the time integration of a numerical model to prepare the forecast ensemble for the next analysis time.

For the training of a DNN, we use the EnKF analysis and forecast provided by an EnKF run as shown in Fig. 1b. The weights including bias parameters are optimized by reducing a loss function that measures a difference between the output of the DNN and the target of training. We prepare several DNNs by randomly initializing the weights before the training.

One of the reasons for including the EnKF analysis in the inputs of DNNs is that this analysis at a grid point contains some information on the forecast, observations, and forecast error covariance in a domain within the covariance localization radius, so that we can reduce the input radius of DNNs and implicitly utilize some information of the forecast error covariance. In addition, even if DNNs cannot deal with some observational data because the input nodes for these observations are absent, they are assimilated by the EnKF part of DL-EnKF and their information is partly provided to the deep learning part through the EnKF analysis.

We can prepare pseudo-observations by other methods. However, it is easily shown by the sequential assimilation method (e.g., Houtekamer and Mitchell, 2001) that if observation errors are independent of each other the additional assimilation of pseudo-observations does not change the EnKF analysis. Therefore, it can be considered that the pseudo-observations thus created are assimilated in the EnKF part of DL-EnKF along with the real observations, and that the same observations including the pseudo-observations are provided to the deep learning part. In this sense, the method adopted in this study may be a natural choice, although it may not be optimal and spurious correlations and biases will be generated. We could expect that DNNs will learn to properly deal with this problem by training.

Lawson and Hansen (2004) showed that an analysis ensemble generated by a deterministic EnKF tends to retain multi-modality that may appear in a forecast ensemble, while this is not the case for a stochastic EnKF. Therefore, a stochastic EnKF is better than a deterministic EnKF for generating an analysis ensemble of the DL-EnKF. However, it is well known that if the ensemble size is relatively small, a stochastic EnKF is inferior in terms of the accuracy of the ensemble mean due to random perturbations that are added to observations (e.g., Sakov and Oke, 2008; Bowler et al., 2013), so that we adopt a deterministic EnKF for the EnKF part in the present paper.

3. Design of experiments

3.1. Outline

The performance of the DL-EnKF is investigated through both perfect and imperfect model experiments using three versions of the 40-variable Lorenz 96 models (Lorenz, 1996) and the serial ensemble square root filter (EnSRF; Whitaker and Hamill, 2002), which is one of the deterministic EnKFs. The ensemble size of the serial EnSRF is set to 10, because we are interested in the performance of the DL-EnKF with a relatively small ensemble. In this and the next sections, the EnKF means the serial EnSRF unless otherwise stated. The purpose of the perfect model experiments is to clarify the basic performance of the DL-EnKF, while that of the imperfect model experiments is to gain insight into the performance of the DL-EnKF when applied to data assimilation in the real atmosphere.

The experiments consist of two phases: the training phase of DNNs and the test phase using data assimilation experiments. In the training phase, we run the models and the EnKF to prepare training and validation datasets, which are used to train DNNs and to verify the accuracy of the output of DNNs, respectively. The length of period and time interval of these datasets are 1 000 and 1, respectively. This large time interval is taken to ensure that each data is almost independent of each other. In the test phase, we run the models to prepare a test dataset for the data assimilation experiments. The length of period of this dataset is also 1 000. The accuracy of the DL-EnKF analysis is compared with the deep learning and EnKF analyses to evaluate the performance of the DL-EnKF. The workflow to create the deep learning analysis is the same as that of the DL-EnKF analysis except for the absence of feedback from the deep learning part to the EnKF part (Fig. 2). The analysis accuracy is evaluated by the RMSE that is the square root of the squared error averaged over the grid points and the period of the test dataset at a time interval of 1.

Fig. 2

In the perfect model experiments, we use the original 40-variable Lorenz 96 model and conduct two types of experiments, Exp-PA and Exp-PB, in which the targets of training are different. The target in Exp-PA is the true state generated by the model, while the target in Exp-PB is an analysis by the stochastic EnKF (Evensen, 1994; Burgers et al., 1998) with an ensemble size of 1 000. This analysis is hereafter referred to as the EnKF1000 analysis. Although this ensemble size may be unrealistic for the 40-variable model, the purpose of Exp-PB is to examine the performance of the DL-EnKF when an analysis with a high accuracy is used as a target.

In the imperfect model experiments, the two-scale Lorenz 96 model with 40 large-scale variables and 400 small-scale variables is used as a substitute of the real atmosphere, while a parameterized Lorenz 96 model with a parameterization of large-scale forcing by small-scale variables is used as a substitute of a numerical model of the real atmosphere. We conduct two types of experiments, Exp-IA and Exp-IB. In Exp-IA, we train DNNs using the simulation data generated by the parameterized model, and conduct the data assimilation experiment using observations generated by the two-scale model. The idea behind Exp-IA is that if a dynamical system and a observing system that are used for the taining of a DNN resemble the real-world systems, we could expect that a data assimilation method the DNN has learned by training also works in the real-world applications. In Exp-IB, the two-scale Lorenz 96 model is used for the training and data assimilation experiments in a perfect model scenario for comparison to Exp-IA.

Table 1 summarizes the models used in the training and test phases of the experiments. The following subsections describe further details of the experimental design.

Table 1

3.2. Models

The governing equations of the Lorenz 96 model for the perfect model experiments are

$$\frac{dX_k}{dt} = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F, \qquad (2)$$

for $k = 1, \cdots, K$, satisfying periodic boundary conditions: $X_{-1} = X_{K-1}$, $X_0 = X_K$, and $X_1 =$

X_{K+1}. The number of variables K and the forcing parameter F are set to 40 and 8, respectively. Note that since the number of positive Lyapunov exponents of the model is 13 for those parameter values (Lorenz and Emanuel, 1998), the ensemble size of 10 is not very small. The leading Lyapunov exponent corresponds to a doubling time of 0.42 (Lorenz and Emanuel, 1998). When the nonlinearity in data assimilation is controlled by changing the time interval between observations as in the present study, this value can be used as a reference for estimating the degree of nonlinearity. The fourth-order Runge-Kutta scheme is adopted for the time integration of the model with a time step 0.01.

The governing equations of the two-scale Lorenz 96 model for the imperfect model experiments are

$$\frac{dX_k}{dt} = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F - \frac{hc}{b}\sum_{j=1}^{J} Y_{j,\,k}, \tag{3}$$

$$\frac{dY_{j,\,k}}{dt} = -cbY_{j+1,\,k}\left(Y_{j+2,\,k} - Y_{j-1,\,k}\right) - cY_{j,\,k} + \frac{hc}{b}X_k, \tag{4}$$

for $k = 1, \cdots, K$ and $j = 1, \cdots, J$, where $\{X_k\}$ are large-scale variables and $\{Y_{j,\,k}\}$ are small-scale variables, satisfying periodic boundary conditions: $X_{-1} = X_{K-1}$, $X_0 = X_K$, $X_1 = X_{K+1}$, $Y_{0,1} = Y_{J,K}$, $Y_{J+1,K} = Y_{1,1}$, and $Y_{J+2,K} = Y_{2,1}$. To make Eq. (4) meaningful, we further define $Y_{0,k} = Y_{J,k-1}$, $Y_{J+1,k} = Y_{1,k+1}$, and $Y_{J+2,k} = Y_{2,k+1}$. Large- and small-scale variables interact with each other through the last terms on the right-hand side of Eqs. (3) and (4). We set the parameters as follows: $K = 40$, $J = 10$, $F = 10$, $h = 1$, $c = 10$, and $b = 10$. These values are the same as the ones used by Lorenz (1996) except for K. Note that the forcing parameter F is larger than in the perfect model experiments. The fourth-order Runge-Kutta scheme is adopted for the time integration of the model with a time step 0.005.

We also need the parameterized Lorenz 96 model in the imperfect model experiments. Although advanced parametrization methods such as stochastic parametrization (e.g., Wilks, 2005) and machine learning-based parametrization (e.g., Schneider et al., 2017) are available, a simple function fitting is adopted in the present study; the last term on the right-hand side of Eq. (3) is approximated by a linear function of X_k. The reason we adopt such a simple approach is that we intend to demonstrate that even an unsophisticated imperfect model works well for the training of a DNN. Then the governing equations of the parameterized Lorenz 96 model are

$$\frac{dX_k}{dt} = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F + (a_1 X_k + a_0), \tag{5}$$

for $k = 1, \cdots, K$, satisfying periodic boundary conditions: $X_{-1} = X_{K-1}$, $X_0 = X_K$, and $X_1 = X_{K+1}$. The number of variables K and the forcing parameter F were set to 40 and 10, respectively, to be consistent with the two-scale Lorenz 96 model. The constants a_1 and a_0 are to be determined by the function fitting. The fourth-order Runge-Kutta scheme is adopted for the time integration of the model with a time step 0.01.

The three models are integrated from $t = 0$ to $t = 2\,050$ for preparing the training and validation datasets. The initial condition at each grid point is F plus an independent random number drawn from the normal distribution with the mean 0 and the variance 1, except that the small-scale variables of the two-scale Lorenz 96 model are set to 0 at the initial time. The data from $t = 51$ to $t = 1\,050$ are used for preparing the training dataset, and those from $t = 1\,051$ to $t = 2\,050$ for the validation dataset. The other time integration of the models from $t = 0$ to $t = 1\,050$ with initial conditions generated by using another random number sequence is conducted for preparing the test dataset, and the state variables from $t = 51$ to $t = 1\,050$ are used as the true state (target) for computing the analysis error.

3.3. Observations

Observations are generated by adding random errors to the results of the time integration of the models. The observation errors are independent random draws from the normal distribution with the mean 0 and the variance 1, so that the standard deviation of observation errors is 1. Observations used in the imperfect experiments are of large-scale variables of the two-scale Lorenz 96 model except for the training phase of Exp-IA, in which observations are prepared by the parameterized Lorenz 96 model.

Nonlinearity in data assimilation is controlled by changing the time interval between observations Δt. All experiments are performed for three values of Δt: 0.05, 0.20, and 0.50. The case of $\Delta t = 0.05$ corresponds to a weakly nonlinear case, and that of $\Delta t = 0.50$ corresponds to a strongly nonlinear one. Note that the latter value is close to the doubling time 0.42 mentioned in Subsection 3.2, and that Penny and Miyoshi (2016) used $\Delta t = 0.50$ for their experiments of a local PF. All observations are prepared such that observations at the same analysis time are the same regardless of the time interval between observations.

For the spatial distribution of observations, two cases are examined. In one case, observations are available at all grid points, and the number of observations is always 40. In other words, observations are available at each grid point with a probability of 1. In the other case, observations are available at each grid point with a probability of 1/2. It is assumed that events that an observation exists are independent of each other in space and time so that the spatial distribution of observations randomly changes at every observation time. The average number of observations is 20, and the standard deviation of the number of observations is $\sqrt{40 \cdot (1/2)^2} \approx 3.16$. Hence, the number of pseudo-observations used by deep learning is about the same as that of observations. The probability of observations is hereafter denoted by p.

3.4. Data assimilation by EnKF

Covariance localization and covariance inflation are needed to optimize the performance of the EnKF. The correlation function defined by Eq. (4.10) of Gaspari and Cohn (1999) is taken for covariance localization. The parameter c in this equation is regarded as the localization radius r_L (unit: grid intervals) in the present study, at which radius the correlation coefficient decreases to 5/24. An adaptive inflation method proposed by Li et al. (2009) is used for multiplicative covariance inflation. This method is based on the innovation statistics by Desroziers et al. (2005). Li et al. (2009) imposed lower and upper limits in the “observed” inflation factor $\tilde{\Delta}^o$ before applying a smoothing procedure: $0.9 \leq \tilde{\Delta}^o \leq 1.2$. Since we conduct data assimilation over a much wider range of the time interval between observations Δt, we optimize the upper limit of $\tilde{\Delta}^o$ for each set of parameters (r_L, Δt, p) leaving the lower limit at 0.9. The candidates of the upper limit are 1.2, 1.3, 1.4, 1.5, 2.0, 3.0, 5.0, and no limit. In addition, although Li et al. (2009) set the error growth parameter κ to 1.03, we adopt a larger value κ = 1.1 because this value leads to a better analysis in the present study. A set of values of r_L and the upper limit of $\tilde{\Delta}^o$ with the best analysis accuracy is hereafter referred to as the optimal parameters. We determine the optimal parameters for each pair of (Δt, p) in Exp-PA and Exp-IA by data assimilation experiments using the target and observations in each training dataset. The optimal parameters thus determined are also used in Exp-PB and Exp-IB, respectively, unless otherwise stated.

In Exp-PB, the target of training is the EnKF1000 analysis that is yielded by the stochastic EnKF with an ensemble size of 1 000, as mentioned in Subsection 3.1. The reason we adopt the stochastic EnKF is that when an ensemble size is very large the accuracy of the serial EnSRF tends to deteriorate and to become less accurate than the stochastic EnKF. We can avoid this problem with the serial EnSRF by applying the mean-preserving random rotation of an analysis ensemble (Sakov and Oke, 2008), but an additional computational cost is very large for the random rotation of a 1000-member ensemble. Figure 3 compares the analysis accuracy of the two EnKFs for ensemble sizes of 10 (in cold colors) and 1 000 (in warm colors), plotting the RMSEs averaged over the period from $t = 51$ to $t = 1050$. This result is obtained by using the target and observations in the training datasets of Exp-PA. The localization radius and the upper limit of $\tilde{\Delta}^o$ are

Fig. 3

optimized in the case of the ensemble size 10, while no covariance localization is applied and the upper limits of $\tilde{\Delta}^o$ is set to 1.2 in the case of the ensemble size 1 000. It is found from Fig. 3 that in the latter case the RMSE of the stochastic EnKF is smaller than that of the serial EnSRF for all values of the time interval between observations Δt and probability of observations p. Note that the serial EnSRF with an ensemble size 10 outperforms the serial EnSRF with an ensemble size 1 000 in the three cases: $(\Delta t,\ p) = (0.05, 1)$, $(0.05, 1/2)$, and $(0.20, 1)$.

In Exp-IB, the two-scale Lorenz 96 model is used to assimilate observations of large-scale variables. As noted by Tsuyuki (2019), when the ensemble size is relatively small, forecast correlations between large- and small-scale variables are not reliable. Hence, these forecast correlations are neglected in the EnKF, and the analysis ensemble of small-scale variables is left unchanged from the forecast ensemble at each analysis time.

3.5. Deep learning

A simple feedforward neural network with the same number of nodes for all hidden layers is adopted for the deep learning part of the DL-EnKF. As we assume that the input radius of DNNs is relatively small, a convolutional neural network would not be needed. We could use a recurrent neural network instead of the feedforward neural network to utilize the information obtained by the previous processing in deep learning, but it is not adopted in the present study for simplicity.

The inputs of a DNN are the EnKF analysis using the optimal parameters, forecast, observations, availability of observations, and pseudo-observations in a small domain centered on an analysis grid point within the input radius r_I (unit: grid intervals). The availability of observation at a grid point is set to 1 if the observation is available and set to -1 if not available. The DNN assumes that observations are always available at all grid points, and we supplement missing observations with pseudo-observations. Since the availability of observations is not necessary in the case of $p = 1$, the input layer of the DNN has $3(2r_I + 1)$ nodes for $p = 1$, and $4(2r_I + 1)$ nodes for $p = 1/2$. The number of hidden layers is set to 5 or 10 and the number of nodes per hidden layer is optimized as will be mentioned later. Since the balance of analysis (Kalnay, 2003) is not a serious issue in Lorenz 96 models, we let the output of the DNN be the analysis value at the analysis grid point only. For Exp-IB in which the two-scale Lorenz 96 model is used in the EnKF part of DL-EnKF, the input and output of the DNN are of large-scale variables only.

Table 2

Table 2 summarizes the architecture and training of the DNN. All input and output data except for the availability of observations are normalized by using the mean and standard deviation of the target state in the training dataset of Exp-PA for the perfect model experiments and of Exp-IB for the imperfect model experiments. Since the statistical behavior of the models does not depend on the location of a grid point, the data at all grid points are used to prepare the training and validation datasets. Hence, the number of samples of each dataset is $40 \times 1\,000 = 40\,000$. The training dataset is split into small batches called mini-batches that are used to compute the loss function and update the weights of the DNN. Learning rate decay is adopted in the training to avoid the situation in which the DNN converges towards minima in a noisy manner and ends up oscillating far away from actual minima. The number of epochs is the number of times each element in the training dataset is used by the DNN for optimizing the weights. For most of the cases, iterations almost converge within 10 epochs. We use PyTorch (Paszke et al., 2019) as the deep learning software.

To determine the optimal number of nodes per hidden layer, we train two DNNs with 5 and 10 hidden layers by changing the number of nodes using the training and validation datasets of Exp-PA. Figure 4 plots the RMSEs of the two DNNs against the input radius r_I for the case of $\Delta t = 0.50$ and $p = 1$. The RMSEs are computed by using the validation datasets. For the DNN with 10 hidden layers and 5 nodes per hidden layer, the training fails for six values of the input radius, so that the RMSE of this case is not plotted in Fig. 4b. Since the RMSE is not very different between 5 and 10 hidden layers, we adopt the DNN with 5 hidden layers. It is found from Fig. 4 that when the input radius and number of nodes are large to some extent, the RMSE tends to increase due to the generalization error of deep learning. Since the DNN with 20 nodes per hidden layer has the smallest RMSE for most of the input radii, we set the optimal number of nodes to 20 for the case of $\Delta t = 0.50$ and $p = 1$.

Fig. 4

The optimal numbers of nodes of the DNN with 5 hidden layers are summarized in Fig.

Fig. 5

5 by blue bars for $p = 1$ and by cyan bars for $p = 1/2$. They tend to increase as the time interval between observations increases because the estimation of state variables becomes more difficult as nonlinearity increases. Although this result is obtained for Exp-PA, those number of nodes are used in all experiments. We also compute the optimal numbers of nodes for the case where the EnKF analysis is not included in the inputs of the DNN. The RMSEs for this case are plotted in Fig. 5 by red bars for $p = 1$ and by orange bars for $p = 1/2$. It is found that the inclusion of the EnKF analysis tends to reduce the optimal number of nodes. This is probably because the EnKF analysis plays the role of a first guess and makes it easier to estimate state variables.

The appendix discusses the impacts of the increase in the ensemble size of EnKF and the sample size for training on the accuracy of output of a DNN. The result of experiments shows that when the ensemble size of EnKF is increased to 40 the improvement by deep learning is considerably reduced, and that we need to increase the sample size much larger to obtain a larger improvement.

3.6 Data assimilation by DL-EnKF

In the data assimilation experiments with the DL-EnKF, the EnKF part of DL-EnKF adopts the optimal parameters, and the DL-EnKF analysis is the average of outputs of 5 or 10 DNNs. Since the adaptive covariance inflation is used in the EnKF part, the parameter α in Eq. (1) is set to 1. In the test phase of Exp-IB, the deep learning part receives only large-scale variables from the EnKF part and generates the DL-EnKF analysis of large-scale variables. The analysis ensemble of large-scale variables is modified by using this analysis, while the analysis ensemble of small-scale variables is left unchanged from the one generated by the EnKF part. The RMSEs of the DL-EnKF, deep learning, and EnKF analysis in Exp-IB are computed by using large-scale variables only.

4. Results

4.1 perfect model experiments

The first issue to be clarified is whether deep learning can outperform the EnKF in terms of analysis accuracy. The EnKF is close to optimal in weakly nonlinear regimes, and the deep learning part of DL-EnKF does not explicitly utilize the forecast error covariance. Figure 6a compares the analysis accuracy between deep learning and the EnKF in Exp-PA for all values of Δt and p, in which the RMSEs are plotted against the RMSE of EnKF for the input radius of 2 grid intervals. The dots indicate the RMSEs of a single DNN, and the short horizontal bars indicate the RMSE of the average of outputs from 5 DNNs. It is found from this figure that all RMSEs of deep learning analysis are the same for each case and that deep learning outperforms the EnKF when $\Delta t = 0.50$. Note that since the EnKF analysis is included in the inputs of DNNs, the accuracy of deep learning analysis does not become worse than that of the EnKF analysis if sufficient training samples are available.

Fig. 6

The second issue is whether the accuracy of the DL-EnKF analysis is better than that of the EnKF and deep learning analyses. As mentioned in the introduction, the analysis by deep learning is based on the minimization of the sum of errors over many samples and not optimized for each analysis time. Hence, the analysis accuracy may deteriorate during assimilation cycles by the DL-EnKF. Figure 6b compares the analysis accuracy between the DL-EnKF and EnKF for Exp-PA. The dots indicate the RMSEs of DL-EnKF when the output of a single DNN is used as the DL-EnKF analysis, and the horizontal bars indicate the ones when the average of outputs from 5 DNNs is used as the DL-EnKF analysis. It is found that when $\Delta t = 0.05$ the RMSEs of DL-EnKF based on a single DNN are scattered and larger than that of EnKF. In other words, the accuracy of the deep learning analysis shown in Fig. 6a is not maintained during assimilation cycles in a weakly nonlinear case. Taking the average over 5 DNNs does not improve the accuracy very well. When $\Delta t = 0.20$ and 0.50, on the other hand, the RMSEs of DL-EnKF based on a single DNN become almost the same for each case and taking the average over 5 DNNs slightly improves the accuracy in the case of $\Delta t = 0.50$. In addition, a comparison of Fig. 6a and 6b shows that the RMSE of DL-EnKF is smaller than that of deep learning when $\Delta t = 0.50$ due to positive feedback in assimilation cycles.

We conduct additional experiments in which the ensemble size of DNNs is increased to 10 in Exp-PA. The initial conditions of weights used for the training are different from the ones used in the case of 5 DNNs. Results are presented in Figs. 6c and 6d, and the former figure looks the same as Fig. 6a. The benefit of taking the average over 10 DNNs for the

DL-EnKF is evident when $\Delta t = 0.05$, although its analysis accuracy is still lower than that of EnKF. When $\Delta t = 0.20$ and $\Delta t = 0.50$, the RMSEs of DL-EnKF are almost the same as in the case of 5 DNNs. This suggests that an ensemble size of 5 is sufficient except for a weakly nonlinear case. Then, all the results shown below are based on the average of outputs from 5 DNNs, because our interest is primarily in the performance of the DL-EnKF in strongly nonlinear regimes.

Fig. 7

Figure 7 compares the time sequences of RMSEs of the EnKF (red line) and the DL-EnKF (green line) in the case of $p = 1$. When $\Delta t = 0.05$ (Fig. 7a), the DL-EnKF is outperformed by the EnKF during the whole period. When $\Delta t = 0.20$ (Fig. 7b), the analysis accuracy of the two methods is close; the correlation coefficient between the two RMSEs computed for the period from $t = 51$ to $t = 1\,050$ is 0.761. When $\Delta t = 0.50$ (Fig. 7c), the EnKF sometimes exhibits a significant deterioration of accuracy, but the DL-EnKF does not show such a tendency. This result demonstrates an excellent performance of the DL-EnKF in strongly nonlinear regimes.

Fig. 8

The third issue is whether the optimal input radius of deep learning is smaller than the optimal localization radius of the EnKF. Figure 8 plots the RMSEs of EnKF (orange lines), deep learning (green lines), and DL-EnKF (blue lines) analysis against the input radius for all cases of Exp-PA (solid lines) and Exp-PB (broken lines). The RMSE of EnKF in Exp-PB is the same as the one in Exp-PA. An orange broken line indicates the RMSE of the EnKF1000 analysis that is used for the training in Exp-PB. The optimal localization radius is plotted by a red arrow, except for the case of $(\Delta t,\ p) = (0.05, 1/2)$ where the optimal localization radius is 11 grid intervals. The RMSEs of EnKF and deep learning overlap in Figs. 8a and 8b, and the RMSEs except for the EnKF1000 analysis almost overlap in Figs. 8c and 8d.

When $\Delta t = 0.05$ (Figs. 8a and 8b), the DL-EnKF is outperformed by the EnKF. Reflecting that an ensemble of 5 DNNs is not sufficient (see Fig. 5b), the graphs of the DL-EnKF are not smooth due to large sampling errors. When $\Delta t = 0.20$ (Fig. 8c and 8d) the two data assimilation methods exhibit almost the same accuracy while when $\Delta t = 0.50$ (Figs. 8e and 8f) the DL-EnKF outperforms the EnKF irrespective of the input radius. The accuracy of the DL-EnKF analysis is higher than that of the deep learning analysis for the latter case due to positive feedback in assimilation cycles. We can conclude that the input radius of 2 grid intervals is sufficient to attain the best accuracy of the DL-EnKF analysis. This value is smaller than the optimal localization radii for both $p = 1$ and $p = 1/2$. Even if the input radius is further increased, the accuracy of the DL-EnKF and deep learning analysis remains almost the same, although slight degradations are seen due to the generalization error of deep learning. This small sensitivity of RMSEs on the input radius indicates that the information at distant grid points contributes little to the DL-EnKF analysis even within the localization radius of the EnKF. The inclusion of the EnKF analysis in the inputs of DNNs also contributes to this insensitivity.

Another point to be noted in Figs. 8e and 8f is that even if DNNs are trained on the

EnKF1000 analysis, the accuracy of the DL-EnKF analysis is not very different from the one trained on the true state. Given the large errors of the EnKF1000 analysis shown in Figs. 8e and 8f, this result may look surprising. That is probably because this analysis well represents the basic dynamics of the Lorenz 96 model despite the large errors. If the difference in the ensemble size between the DL-EnKF and the target analysis is decreased, the accuracy of the DL-EnKF analysis in Exp-PB is more deteriorated. For instance, according to an additional experiment in which the ensemble size of the DL-EnKF is set to 40 for the case of $\Delta t = 0.50$ and $p = 1$ (see the appendix), the RMSEs of the EnKF analysis and DL-EnKF analyses in Exp-PA and Exp-PB are 0.682, 0.617, and 0.638, respectively, for the input radius of 2 grid intervals. The corresponding values for the ensemble size of 10 are 0.798, 0.675, and 0.689 (see Fig. 8e), so that the deterioration of accuracy in Exp-PB is still not very large.

Finally, we examine the impact of including the EnKF analysis in the inputs of DNNs on the accuracy of the deep learning analysis using the test datasets of Exp-PA. Figure 9 plots the RMSE of deep learning in which the EnKF analysis is not included (cyan line) and the one in which the EnKF analysis is included (green line) against the input radius. The green lines are the same as in Fig. 8, and the two RMSEs overlap in Fig. 9f. For comparison, the RMSE of EnKF of which localization radius is not optimized is also plotted by an orange line against the localization radius with the upper limit of $\tilde{\Delta}^o$ optimized for each localization radius. Note that the RMSE of EnKF does not always attain the minimum at the optimal

Fig. 9

localization radius indicated by a red arrow, because the values of the optimal parameters are determined by using the training datasets.

We can see from Fig. 9 that the accuracy of the deep learning analysis is improved by including the EnKF analysis in the inputs of DNNs except for Fig. 9f, in which the EnKF analysis is too inaccurate to be useful. It is also found in Figs. 9c and 9e for $p = 1$ that this procedure reduces the dependence of the analysis accuracy on the input radius. This is because the EnKF analysis contains some information on the forecast ensemble and observations in a domain within the localization radius. Such a reduction in the dependence brought about by including the EnKF analysis is not clearly seen in Figs. 9d and 9f for $p = 1/2$, since deep learning partly utilizes the EnKF analysis through pseudo-observations.

4.2 Imperfect model experiments

The parametrization procedure for the parameterized Lorenz 96 model is described in Subsection 3.2. Figure 10 is the scatter plot between the large-scale variables and the forcing. The initial condition is the same as that used for preparing the training dataset of Exp-IB. The number of samples is 40 000 and the result of linear function fitting is plotted by a straight line. The values of constants in Eq. (5) are $a_1 = -0.320$ and $a_0 = -0.165$. Since the slope of this line is negative, the forcing acts on large-scale variables as negative feedback. Figure 11 compares the Hovmöller diagrams of the Lorenz 96 model, parameterized Lorenz 96 model, and large-scale variables of the two-scale Lorenz 96 model.

Fig. 10

Fig. 11

The initial condition is the same as that used in Fig. 10. Note that the forcing parameter F is larger than in the perfect model experiments. A comparison of the three panels in Fig. 11 shows that the parameterization works well, although the parametrized Lorenz 96 model evolves a little more regularly than the two-scale Lorenz 96 model. Stochastic parameterizations could remedy this defect (Wilks, 2005).

Fig. 12

Figure 12 plots the RMSEs of EnKF (orange lines), deep learning (green lines), and DL-EnKF (blue lines) analysis against the input radius for all cases of Exp-IA (solid lines) and Exp-IB (broken lines). Note that Exp-IA is conducted in an imperfect model scenario, while Exp-IB is conducted in a perfect model scenario for comparison. Unlike Fig. 8, an orange broken line indicates the RMSE of EnKF using the two-scale Lorenz 96 model. The optimal localization radius of the EnKF is indicated by a red arrow. The RMSEs of EnKF and deep learning for each case overlap in Figs. 12a-12d. It is found that the RMSE of EnKF using the two-scale Lorenz 96 model is smaller than the one using the parameterized Lorenz 96 model. We can confirm that the basic performance of the DL-EnKF is the same as in the perfect model experiments; the DL-EnKF is inferior to the EnKF in a weakly nonlinear case (Figs. 12a and 12b), while the opposite is true in a strongly nonlinear case (Figs. 12e and 12f), in which the optimum input radius is smaller than the optimum localization length. A difference from the perfect model experiments is that when $\Delta t = 0.20$ (Figs. 12c and 12d), the accuracy of the DL-EnKF analysis is a little worse than that of the EnKF analysis for $p = 1$ and a little better for small values of the input radius for $p = 1/2$.

An important point to be noted in Figs. 12e and 12f is that even if DNNs are trained on the training dataset prepared by the parameterized Lorenz 96 model, the improvement in analysis accuracy introduced by the DL-EnKF is not very different from the case where the training dataset is prepared by the two-scale Lorenz 96 model. The former model is run in the data assimilation experiments in the test phase of Exp-IA without any trouble, implying that this model well represents the basic dynamics of large-scale variables of the two-scale Lorenz 96 model. When we use the Lorenz 96 model with $F = 10$, of which evolution is shown in Fig.11a, in the above data assimilation experiments, we often experience failures.

5. Summary and discussion

An ensemble data assimilation method based on deep learning was presented, in which an ensemble of DNNs was locally embedded in an EnKF. This method was named the DL-EnKF. The inputs of a DNN were the EnKF analysis, forecast, observations, availability of observations, and pseudo-observations in a small domain centered on an analysis grid point. Missing observations were supplemented with the pseudo-observations created from the EnKF analysis. The DL-EnKF analysis was the average of outputs from an ensemble of DNNs. The DL-EnKF analysis ensemble was generated from the DL-EnKF analysis and the EnKF analysis deviation ensemble. The members of the DL-EnKF analysis ensemble thus generated were evolved by the time integration of a numerical model to prepare the forecast ensemble for the next analysis time.

The performance of the DL-EnKF was investigated through data assimilation experiments in both perfect and imperfect model scenarios using three versions of the Lorenz 96 model and the serial EnSRF with an ensemble size of 10. The target of training in the perfect model experiments was the true state generated by the Lorenz 96 model or the EnKF1000 analysis generated by the stochastic EnKF with an ensemble size of 1000. In the imperfect model experiments, the true state and observations were provided by the two-scale Lorenz 96 model, while the training dataset was prepared by using the parameterized Lorenz 96 model. Nonlinearity in data assimilation was controlled by changing the time interval between observations.

The DL-EnKF was outperformed by the serial EnSRF in a weakly nonlinear case, but it was superior to the serial EnSRF in terms of analysis accuracy in a strongly nonlinear case despite such a small ensemble size. The DL-EnKF analysis was more accurate than the output of deep learning due to positive feedback in assimilation cycles in the latter case. Even if the target of training was the EnKF1000 analysis or the simulation by the parametrized Lorenz 96 model, the improvement introduced by the DL-EnKF was not very different from the case where the target of training was the true state. The inclusion of EnKF analysis in the inputs of DNNs not only improved the accuracy of the deep learning analysis but also reduced the optimal number of nodes per hidden layer and the dependence of the accuracy on the input radius.

Although the above results were obtained from experiments using toy models, they suggest that the DL-EnKF may be a promising methods for data assimilation in strongly nonlinear regimes. The DL-EnKF works with a relatively small ensemble size compared to the PF, and we can prepare a training dataset for deep learning from simulation data by a numerical model used in data assimilation. Observational data and EnKF analysis data generated with a large ensemble could be used for this purpose, but a huge computational cost may be needed to obtain sufficient samples and a period when observations are available is limited.

The DL-EnKF may be suitable for data assimilation in cloudy or convective regions in the atmosphere to assimilate radar and satellite observations. We need to extend the inputs and output of DNNs in the vertical to assimilate satellite radiance data, since they are nonlocal observations. As for radial wind data by a Doppler radar, the direction and distance of a radar site differ depending on a grid point. However, if the radar site position relative to the grid point is included in the inputs of a DNN, we can train the DNN collectively regardless of the grid point as in the present study.

There are a couple of issues to be addressed before applying the DL-EnKF to data assimilation in the atmosphere. In the data assimilation experiments using Lorenz 96 models, the analysis value at a single grid point is sufficient for the output of a DNN, but we need to take the balance of analysis into account for atmospheric data assimilation. One of the methods for ensuring the balance is to extend the output of a DNN to include analysis values at surrounding grid points. Then the target of training consists of the target state in a small domain centered on an analysis grid point. Since the target state is usually well balanced, the DNN could learn the balance. Adding a penalty term for suppressing imbalance to a loss function of the DNN may help enhance the balance. In addition, taking a weighting average of the outputs in adjacent domains may be effective in improving analysis accuracy.

This study demonstrates that the DL-EnKF is inferior to the EnKF in a weakly nonlinear case. It is found that an increase in the ensemble size of DNNs can mitigate this problem, but it would be difficult to increase the ensemble size sufficiently, given the computational cost needed for the training of DNNs. We may need a criterion for replacing the EnKF analysis with the corresponding DL-EnKF analysis in DL-EnKF assimilation cycles. An advanced DNN such as a recurrent neural network would be useful for improving the performance of DL-EnKF in weakly nonlinear regimes as well as in strongly nonlinear ones.

Data Availability Statement

The Python programs used for Exp-PA in this study are available on J-STAGE Data.

Acknowledgments

The authors thank Dr. Arata Amemiya (RIKEN) for his insightful suggestion on the target of training. The comments of two anonymous reviewers were very helpful to improve the manuscript. This work was supported by the Ministry of Education, Culture, Sports, Science and Technology through the program for Promoting Researches on the Supercomputer

Fugaku (Large Ensemble Atmospheric and Environmental Prediction for Disaster Prevention and Mitigation; hp200128, hp210166), and by the Japan Society for the Promotion of Science through KAKENHI (Study on Uncertainty of Cumulonimbus Initiation and Development Using Particle Filter; JP17H02962).

Appendix

In this study, we perform the experiments using the serial EnSRF with 10 members and the 40 000 training samples. It may be of interest to examine how the accuracy of output of a DNN changes when the ensemble size and the sample size are increased. This appendix presents some results of additional experiments in which the ensemble size of EnKF is set to 10 and 40 and the sample size is set to 40 000, 160 000 and 640 000. These experiments correspond to Exp-PA for $\Delta t = 0.50$. The ensemble size of 40 is the same as the degrees of freedom of the Lorenz 96 model and, according to Fig. 5 of Penny and Miyoshi (2016), an EnKF still outperforms a local PF with this ensemble size. The periods of time integration of the model for preparing the training and validation datasets are 2 050, 8 050, and 32 050 with the first periods of 50 in length are discarded.

Figure A1 shows the optimum numbers of nodes per hidden layer of a DNN, obtained by using the validation datasets. The maximum number of nodes is limited to 100. Although we choose the number of nodes that performs the best for the various input radius, the determination of the optimal number becomes difficult with the increase of the sample size.

Fig. A1

When the number of training samples is increased, the generalization error of deep learning tends to reduce and, therefore, the optimal number of nodes per hidden layer tends to increase.

Figure A2 compares RMSE between the serial EnSRF and the output of a DNN obtained by using the test datasets of Exp-PA. Note that they are not the average over 5 DNNs, so that the RMSEs shown by green lines in Figs. A2a and A2b are different from those in Figs. 8e and 8f, respectively. The optimal localization radius of the serial EnSRF in Fig. A2c is 12 grid intervals. It is found from this figure that the improvement by deep learning is considerably reduced for the ensemble size of 40. When the sample size is increased, the RMSE of the output of a DNN is reduced, but we need much more training samples to obtain a large improvement.

Fig. A2

References

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

Arcucci, R., J. Zhu, S. Hu and Y.-K. Guo, 2021: Deep data assimilation: Integrating deep learning with data assimilation. *Appl. Sci.*, **11**, 1114-1134. doi:10.3390/app11031114.

Bocquet, M., 2016: Localization and the iterative ensemble Kalman smoother. *Quart. J. Roy. Meteor. Soc.*, **142**, 1075–1089. doi:10.1002/qj.2711.

Bocquet, M., J. Brajard, A. Carrassi and L. Bertino, 2020: Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization. *Foundations of Data Science*, **2**, 55–80.

Bocquet, M., C. A. Pires and L. Wu, 2010: Beyond Gaussian statistical modeling in geophysical data assimilation. *Mon. Wea. Rev.*, **138**, 1997-3023. doi:10.1175/2010MWR3164.1.

Bocquet, M. and P. Sakov, 2013: Joint state and parameter estimation with an iterative ensemble Kalman smoother. *Nonlin. Processes Geophys.,* **20**, 803–818. doi:10.5194/npg-20-803-2013.

Bocquet, M. and P. Sakov, 2014: An iterative ensemble Kalman smoother. *Quart. J. Roy. Meteor. Soc.,* **140**, 1521–1535. doi:10.1002/qj.2236.

Bowler, N. E., J. Flowerdew, and S. R. Pring, 2013: Tests of different flavours of EnKF on a simple model. *Quart. J. Roy. Meteor. Soc.*, **139**, 1505-1519. doi:10.1002/qj.2055.

Brajard, J., A. Carrassi, M. Bocquet and L. Bertino, 2020a: 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. Comp. Sci.*, **44**, 101171.

Brajard, J., A. Carrassi, M. Bocquet and L. Bertino, 2020b: Combining data assimilation and machine learning to infer unresolved scale parametrisation. *Philos. Trans. Roy. Soc. London, Ser. A*, **379**, 2020.0086. doi:10.1098/rsta.2020.0086.

Burgers, G., P. J. van Leeuwen, and G. Evensen, 1998: Analysis schemes in the ensemble

Kalman filter. *Mon. Wea. Rev.*, **126**, 1719-1724.

Chattopadhyay, A., P. Hassanzadeh and D. Subramanian, 2020: Data-driven prediction of a multi-scale Lorenz 96 chaotic system using deep learning methods: Reservoir computing, artificial neutral network, and long short-term memory network. *Nonlinear Processes Geophys.*, **27**, 373–389. doi:10.5194/npg-27-373-2020.

Desroziers, G., L. Berre, B. Chapnik, and P. Poli, 2005: Diagnosis of observation, background and analysis-error statistics in observation space. *Quart. J. Roy. Meteor. Soc.*, **131**, 3385–3396.

Dueben, P. D. and P. Bauer, 2018: Challenges and design choices for global weather and climate models based on machine learning, *Geosci. Model Dev.*, **11**, 3999–4009. doi:10.5194/gmd-11-3999-2018.

Ehrendorfer, M., 1994: The Liouville equation and its potential usefulness for the prediction of forecast skill. Part I: Theory. *Mon. Wea. Rev.*, **122**, 703-713.

Evensen, G., 1994: Sequential data assimilation with a nonlinear quasigeostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys., Res.*, **99**, 10143-10162.

Farchi, A. and M. Bocquet, 2018: Review article: Comparison of local particle filters and new implementations, *Nonlinear Processes Geophys.*, **25**, 765-807. doi:10.5194/npg-25-765-2018.

Farchi, A. M. Laloyaux, M. Bonavita and M. Bocquet, 2021: Using machine leaning to correct model error in data assimilation and forecast applications. *Quart. J. Roy. Meteor. Soc.*, **147**, 3067-3084. doi:10.1002/qj.4116.

Gaspari, G. and S. E. Cohn, 1999: Construction of correlation functions in two and three dimensions. *Quart. J. Roy. Meteor. Soc.*, **125**, 723-757.

Geer, A. J., 2020: Learning earth system models from observations: machine learning or data assimilation? *Technical Memorandom 863*, ECMWF, 23pp.

Gottwald, G. A. and Reich, S. (2021). Supervised learning from noisy observations: Combining machine-learning techniques with data assimilation. *Physica D: Nonlinear Phenomena*, **423**, 132911. doi:10.1016/j.physd.2021.132911.

Houtekamer, P. L. and H. L. Mitchell, 2001: A sequential ensemble Kalman filter for atmospheric data assimilation. *Mon. Wea. Rev.*, **129**, 123-137.

Hsieh, W. W. and B. Tang, 1998: Applying neural network models to prediction and data analysis in meteorology and oceanography. *Bull. Amer. Meteor. Soc*., **79**, 1855-1870.

Kalnay, E., 2003: *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, Cambridge, 364 pp.

Kawabata, T. and G. Ueno, 2020: Non-Gaussian probability densities of convection initiation and development investigated using a particle filter with a storm-scale numerical weather prediction model. *Mon. Wea. Rev.*, **148**, 3–20. doi:10.1175/MWR-D-18-0367.1.

Kingma, D. and J. Ba, 2015: Adam: A method for stochastic optimization. arXiv:1412.6980.

Kondo, K. and T. Miyoshi, 2019: Non-Gaussian statistics in global atmospheric dynamics: a study with a 10 240-member ensemble Kalman filter using an intermediate atmospheric general circulation model. *Nonlinear Processes Geophys.*, **26**, 211–225. doi:10.5194/npg-26-211-2019.

Lawson, G. W. and J. A. Hansen, 2004: Implications of stochastic and deterministic filters as ensemble-based data assimilation methods in varying regimes of error growth. *Mon. Wea. Rev.*, **132**, 1966-1981.

Le Cum, Y., Y. Bengio and G. Hinton, 2016: Deep learning. *Nature*, **521**, 436-444. doi:10.1038/nature14539.

Li, H., E. Kalnay and T. Miyoshi, 2009: Simultaneous estimation of covariance inflation and observation errors within an ensemble Kalman filter. *Quart. J. Roy. Meteor. Soc.*, **135**, 523-533. doi:10.1002/qj.371.

Lorenz, E. D., 1996: Predictability – A problem partly solved, *Proceedings of the ECMWF Seminar on Predictability (4-9 September 1995, Reading, UK)*, ECMWF, 1-18.

Lorenz, E. D. and K. A. Emanuel, 1998: Optimal sites for supplementary weather observations: Simulation with a small model. *J. Atmos. Sci.*, **55**, 399-414.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai and S. Chintala, 2019: PyTorch: An imperative style, high-performance deep learning library. In H. Wallach et al., eds. *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 8024–8035. Available at: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Penny, S. G. and T. Miyoshi, 2016: A local particle filter for high-dimensional geophysical systems, *Nonlinear Processes Geophys.*, **23**, 391-405. doi:10.5194/npg-23-391-2016.

Reichstein, M., G. Camps-Vallis, B. Stevens, M. Jung, J. Denzler and N. Carvalhais and Prabhat, 2019: Deep learning and process understanding for data-driven Earth system science, *Nature*, **566**, 195–204. doi:10.1038/s41586-019-0912-1.

Sakov, P. and P. R. Oke, 2008: Implications of the form of the ensemble transformation in the ensemble square root filters. *Mon. Wea. Rev.*, **136**, 1042–1053. doi:10.1175/2007MWR2021.1.

Sakov, P., D. S. Oliver and L. Bertino, 2012: An iterative EnKF for strongly nonlinear systems. *Mon. Wea. Rev.*, **140**, 1988–2004. doi:10.1175/MWR-D-11-00176.1.

Schneider, T., S. Lan, A. Stuart and J. Teixeira, 2017: Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophys. Res. Lett.*, **44**, 12396–12417. doi:10.1002/2017GL076101.

Silva, V. L., C. E. Heaney, Y. Li and C. C. Pain, 2021: Data Assimilation Predictive GAN (DAPredGAN): Applied to determine the spread of COVID-19. arXiv:2105.07729.

Snyder, C, T. Bengtsson, P. Bickel and J. Anderson, 2008: Obstacles to high dimensional particle filtering. *Mon. Wea. Rev.* **136**, 4629–4640. doi:10.1175/2008MWR2529.1

Tomizawa, F. and Y. Sawada, 2020: Combining ensemble Kalman filter and reservoir computing to predict spatio-temporal chaotic systems from imperfect observations and models. *Geosci. Model Dev. Discuss.* doi:10.5194/gmd-2020-211.

Tsuyuki, T., 2014: Deterministic predictability of the most probable state and reformulation of variational data assimilation. *J. Meteor. Soc. Japan*, **92**, 599-622. doi:10.2151/jmsj.2014-606.

Tsuyuki, T., 2019: Ensemble Kalman filtering based on potential vorticity for atmospheric multi-scale data assimilation. *J. Meteor. Soc. Japan*, **97**, 1191-1210. doi:10.2151/jmsj.2019-067.

Whitaker, J. S. and T. M. Hamill, 2002: Ensemble data assimilation without perturbed observations. *Mon. Wea. Rev.*, **130**, 1913-1924.

Wikner, A., J. Pathak, B. R. Hunt, I. Szunyogh, M. Girvan and E. Ott, 2021: Using data assimilation to train a hybrid forecast system that combines machine-learning and knowledge-based components. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, ***31***, 053114.

Wilks, D. J., 2005: Effects of stochastic parametrizations in the Lorenz ’96 system. *Quart. J. Roy. Meteor. Soc.*, **131**, 389-407. doi:10.1256/qj.04.03.

van Leeuwen, P. J., 2009: Particle filtering in geophysical systems. *Mon. Wea. Rev.*, **137**, 4089‐4114. doi:10.1175/2009MWR2835.1.

van Leeuwen, P. J., H. R. Kunsch, L. Nerger, R. Potthast and S. Reich, 2019: Particle filters for high-dimensional geoscience applications: A review. *Quart. J. Roy. Meteor. Soc.*, **145**, 2335-2365. doi:10.1002/qj.3551.

List of Figures

Fig. 5 The optimal number of nodes per hidden layer of a DNN with 5 hidden layers. The abscissa is the observation time interval. Blue and cyan bars are for the case of including EnKF analysis in input for the probability of observations 1 and 1/2, respectively. Red and orange bars are for the case of not including EnKF analysis in input for the probability of observations 1 and 1/2, respectively.

Fig. 6 Comparison of RMSEs between (a) EnKF (abscissa) and deep learning with 5 DNNs (ordinate), (b) EnKF and DL-EnKF with 5 DNNs, (c) EnKF and deep learning with 10 DNNs, and (d) EnKF and deep learning with 10 DNNs for Exp-PA. The probability of observations 1 is in blue and 1/2 in red, and the input radius is 2 grid intervals. Dots indicate RMSEs based on a single DNN, and short horizontal bars indicate RMSEs based on an ensemble of DNNs. The three groups of dots and a horizontal bar in the same color correspond to the observation time intervals 0.05, 0.20, and 0.50 from left to right.

Fig. 7 Time sequences of RMSEs of EnKF (red lines) and DL-EnKF (blue lines) for observation time interval (a) 0.05, (b) 0.20, and (c) 0.50 for Exp-PA. The probability of observations is 1 and the input radius is 2 grid intervals.

Fig. 8 Comparison of RMSEs of EnKF (orange lines), deep learning (green lines), and DL-EnKF (blue lines) for Exp-PA (solid lines) and Exp-PB (broken lines). An orange

broken line indicates the RMSE of EnKF1000 analysis used for training in Exp-PB. The RMSEs are plotted against the input radius, and a red arrow indicates the optimal localization radius of EnKF. The observation time interval and the probability of observations are (a) 0.05 and 1, (b) 0.05 and 1/2, (c) 0.20 and 1, (d) 0.20 and 1/2, (e) 0.50 and 1, and (f) 0.50 and 1/2, respectively.

Fig. 9 Comparison of RMSEs of EnKF (orange line), deep learning not including EnKF analysis in input (cyan line), and deep learning including the EnKF analysis in input (green line) for Exp-PA. The RMSE of EnKF is computed for each localization radius. The abscissa is the input radius for deep learning and the localization radius for the EnKF. A red arrow indicates the optimal localization radius. The observation time interval and the probability of observations are (a) 0.05 and 1, (b) 0.05 and 1/2, (c) 0.20 and 1, (d) 0.20 and 1/2, (e) 0.50 and 1, (f) 0.50 and 1/2, respectively.

Fig. 10 Scatter plot between large-scale variables (abscissa) and large-scale forcing by small-scale variables (ordinate) of the two-scale Lorenz 96 model. A solid line is the result of linear function fitting.

Fig. 11 Hovmöller diagrams of (a) the Lorenz 96 model, (b) the parametrized Lorenz 96 model, and (c) large-scale variables of the two-scale Lorenz 96 model.

Fig. 12 Same as Fig. 8 except for Exp-IA (solid lines) and Exp-IB (broken lines) and that an orange broken line indicates the RMSE of EnKF using the two-scale Lorenz 96 model.

Fig. A1 The optimal number of nodes per hidden layer of a DNN with 5 hidden layers for the time interval between observations of 0.50. The abscissa is the number of samples. Blue and cyan bars are for the EnKF ensemble size of 10 for the probability of observations 1 and 1/2, respectively. Red and orange bars are for the EnKF ensemble size of 40 for the probability of observations 1 and 1/2, respectively.

Fig. A2 Comparison of RMSE between EnKF (orange lines) and the output of a DNN with the number of samples of 40 000 (green lines), 160 000 (blue), and 640 000 (cyan) for the observation time interval of 0.50. The ensemble size of EnKF and the probability of observations are (a) 10 and 1, (b) 10 and 1/2, (c) 40 and 1, (d) 40 and 1/2, respectively. The RMSEs are plotted against the input radius, and a red arrow indicates the optimal localization radius of EnKF.

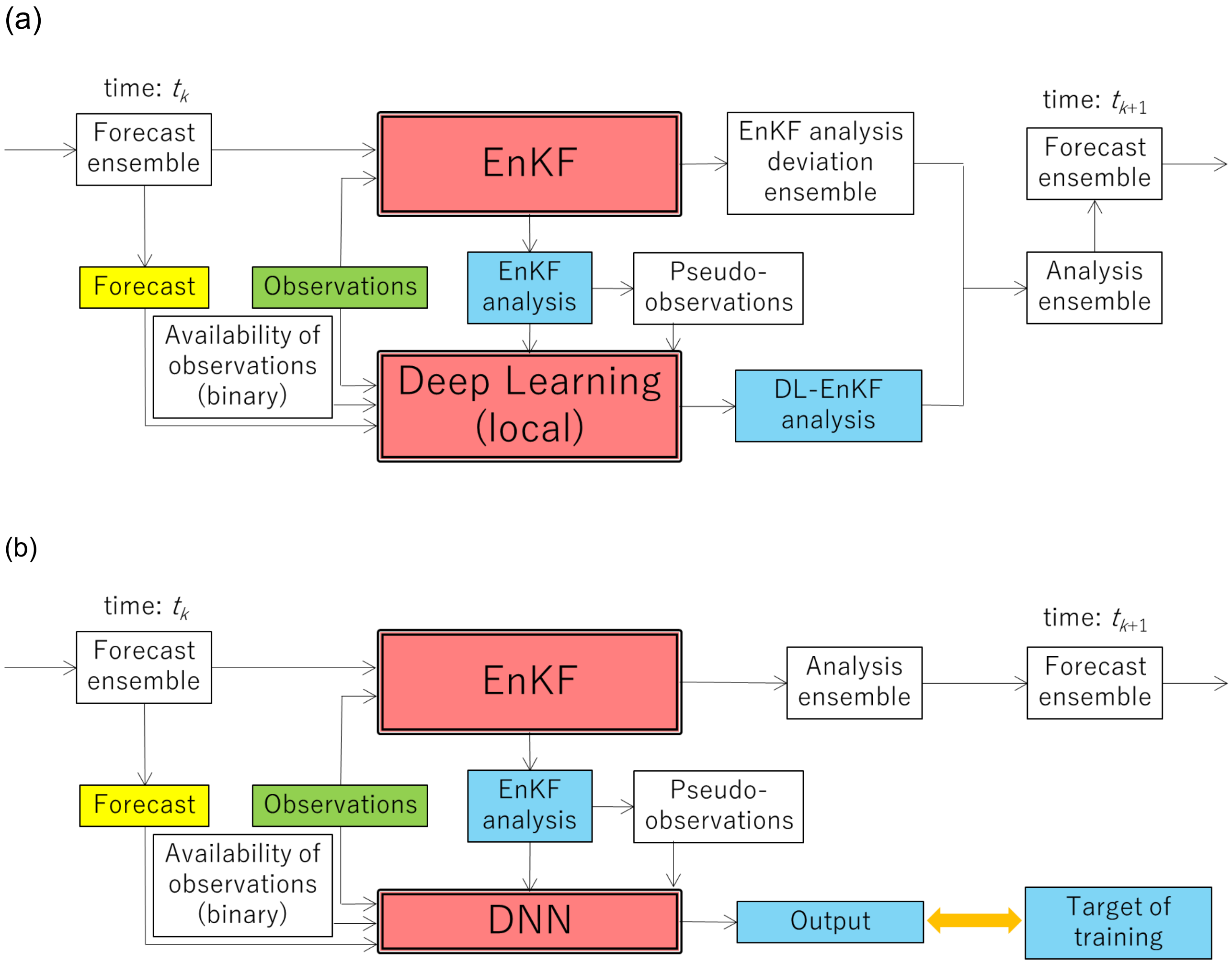


Fig. 1 (a) Workflow of DL-EnKF and (b) workflow to train a DNN for DL-EnKF. See text for details.

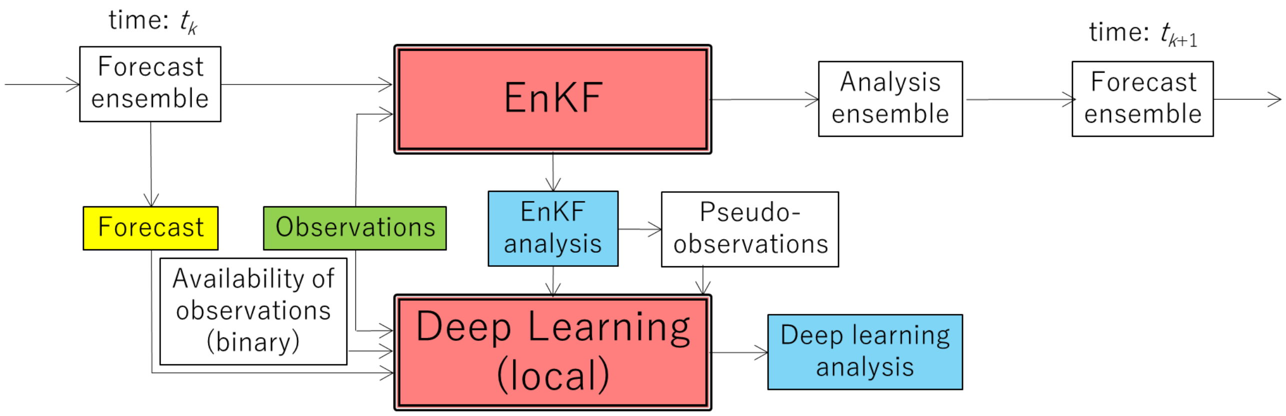


Fig. 2 Workflow to generate deep learning analysis.

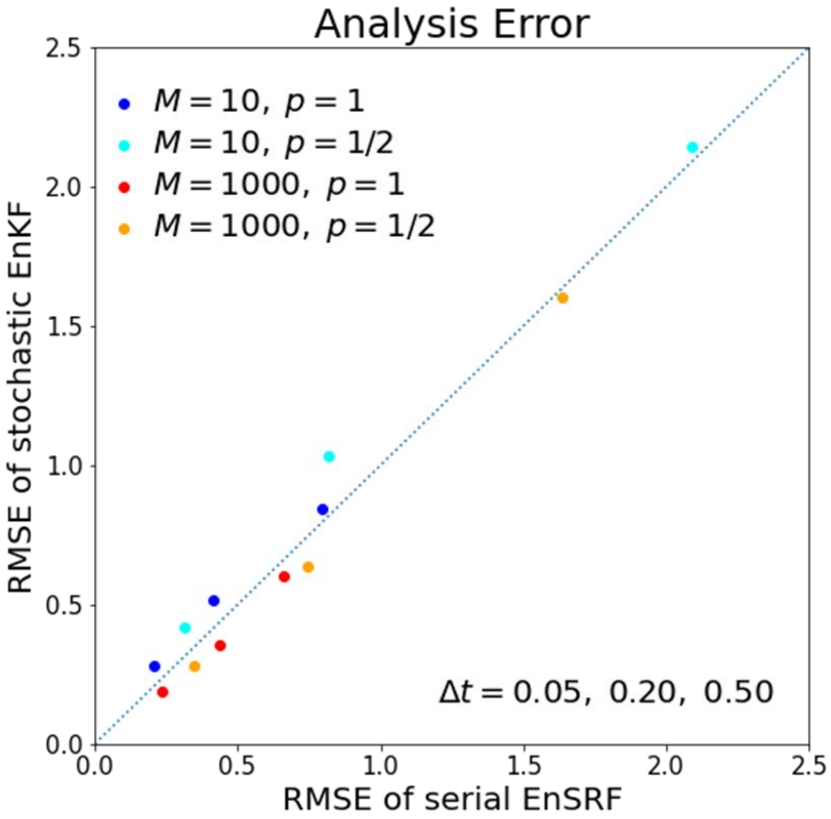


Fig. 3 Comparison of RMSEs between the serial EnSRF (abscissa) and stochastic EnKF (ordinate) for the training dataset of Exp-PA. Dots in cold colors are for an ensemble size 10 and dots in warm colors are for an ensemble size 1 000. Dots in dark colors are for the probability of observations 1 and dots in light colors are for the probability of observations 1/2. The three dots in the same color correspond to the observation time intervals 0.05, 0.20, and 0.50 from left to right.

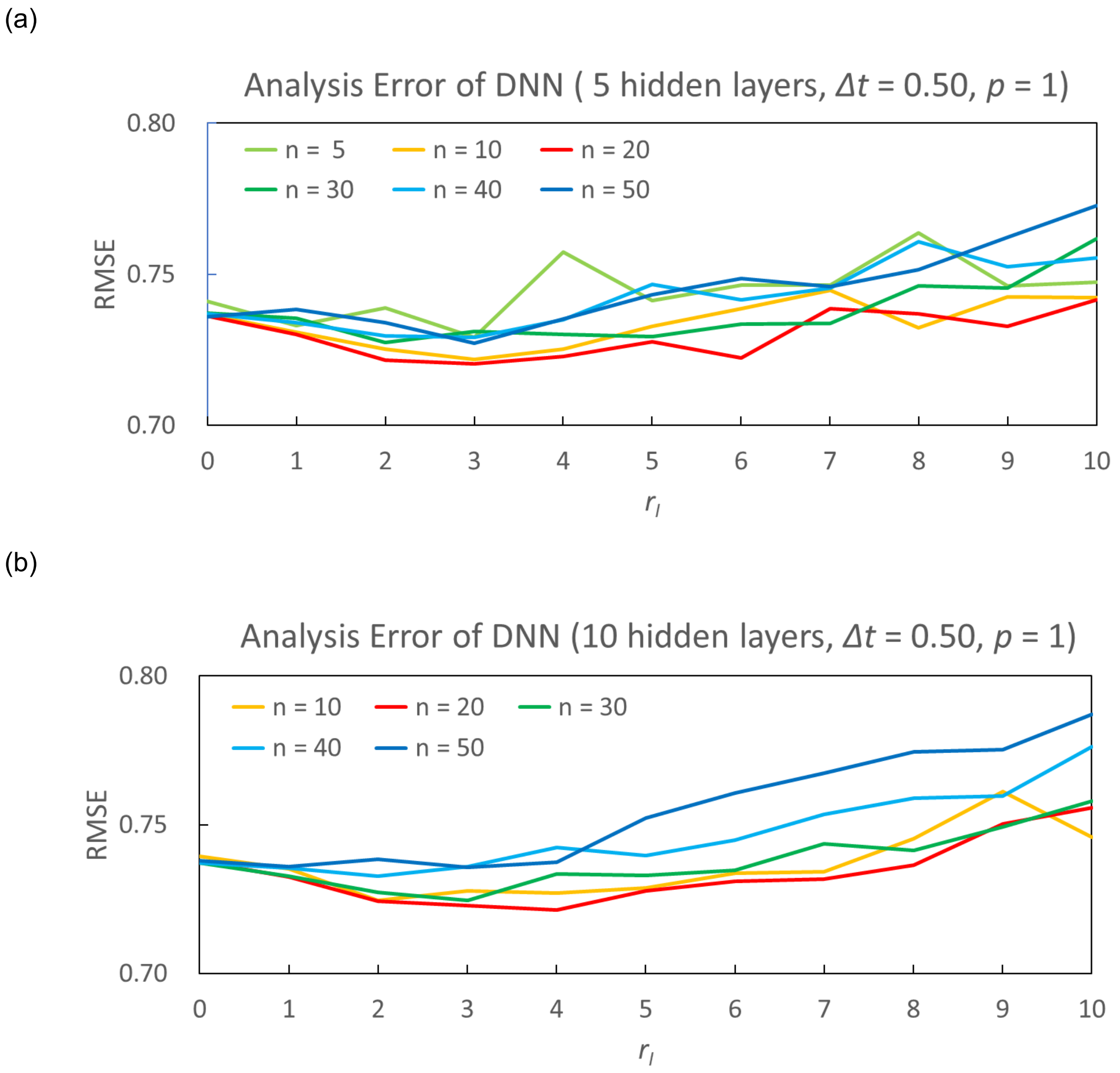


Fig. 4 Comparison of RMSEs of a single DNN with 5 (light green), 10 (orange), 20 (red), 30 (green), 40 (cyan), and 50 (blue) nodes per hidden layer for the validation dataset of Exp-PA. The RMSEs are plotted against the input radius. The number of hidden layers is (a) 5 and (b) 10. The observation time interval is 0.50 and the probability of observations is 1.

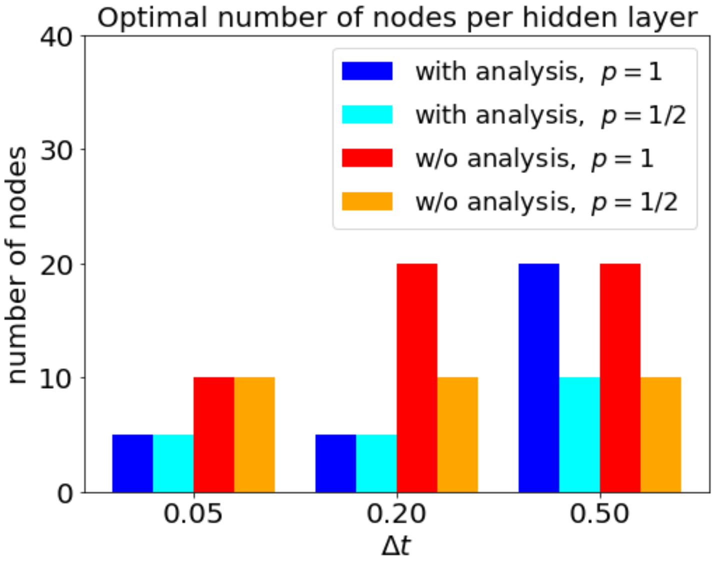


Fig. 5 The optimal number of nodes per hidden layer of a DNN with 5 hidden layers. The abscissa is the observation time interval. Blue and cyan bars are for the case of including EnKF analysis in input for the probability of observations 1 and 1/2, respectively. Red and orange bars are for the case of not including EnKF analysis in input for the probability of observations 1 and 1/2, respectively.

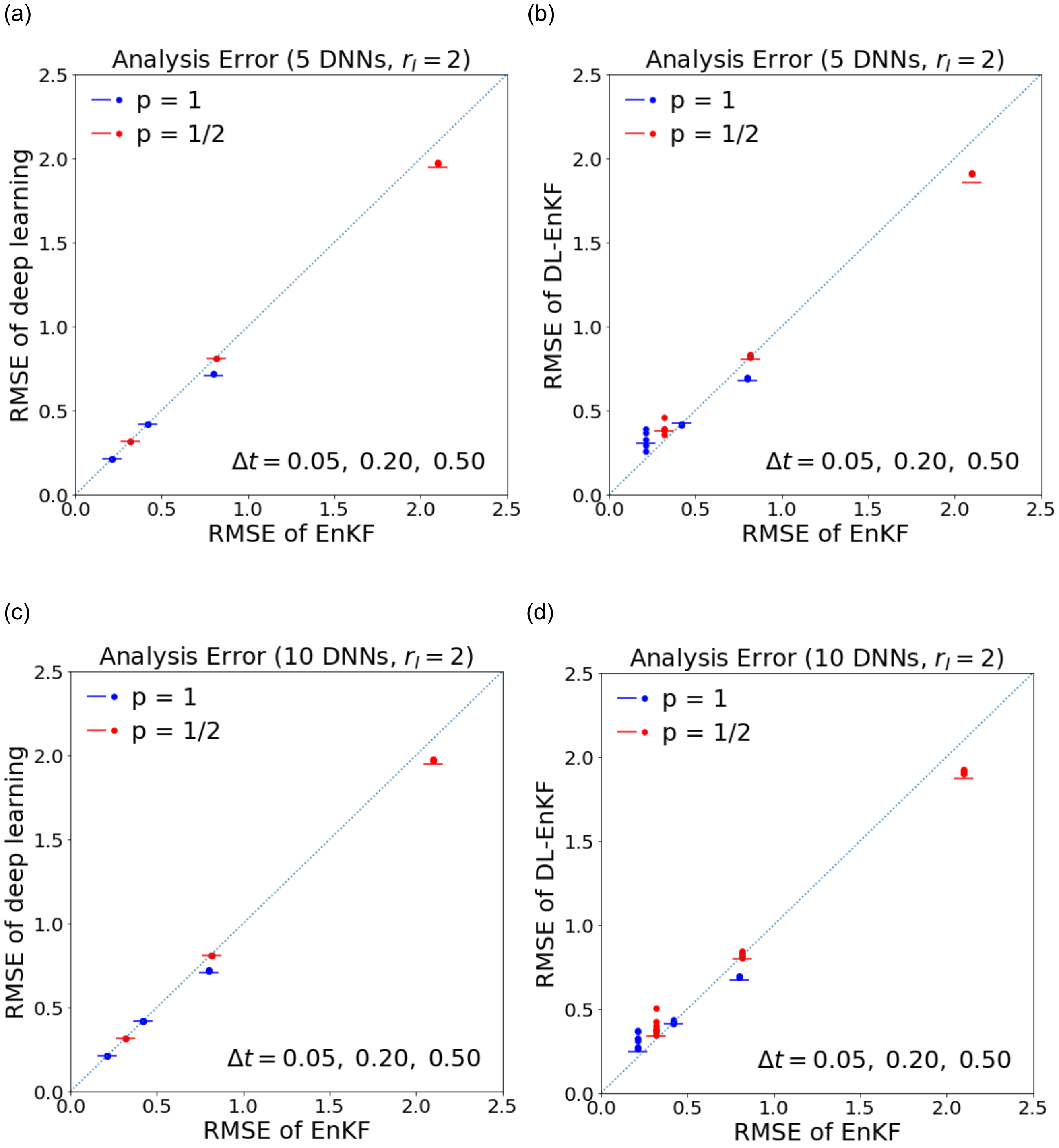
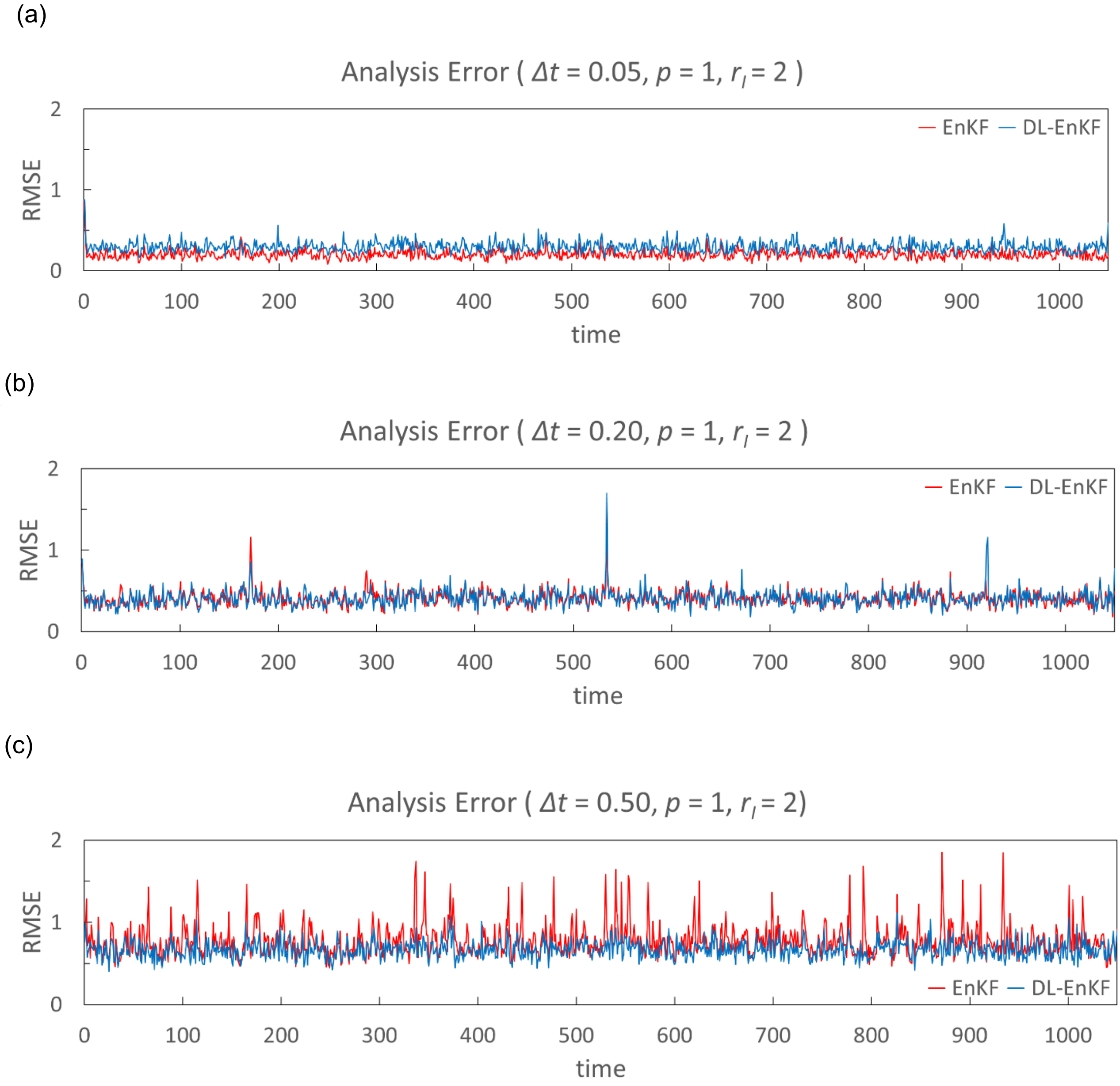


Fig. 6 Comparison of RMSEs between (a) EnKF (abscissa) and deep learning with 5 DNNs (ordinate), (b) EnKF and DL-EnKF with 5 DNNs, (c) EnKF and deep learning with 10 DNNs, and (d) EnKF and deep learning with 10 DNNs for Exp-PA. The probability of observations 1 is in blue and 1/2 in red, and the input radius is 2 grid intervals. Dots indicate RMSEs based on a single DNN, and short horizontal bars indicate RMSEs based on an ensemble of DNNs. The three groups of dots and a horizontal bar in the same color correspond to the observation time intervals 0.05, 0.20, and 0.50 from left to right.


(a)

Analysis Error (Δt = 0.05, p = 1, r_l = 2)

(b)

Analysis Error (Δt = 0.20, p = 1, r_l = 2)

(c)

Analysis Error (Δt = 0.50, p = 1, r_l = 2)

Fig. 7 Time sequences of RMSEs of EnKF (red lines) and DL-EnKF (blue lines) for observation time interval (a) 0.05, (b) 0.20, and (c) 0.50 for Exp-PA. The probability of observations is 1 and the input radius is 2 grid intervals.

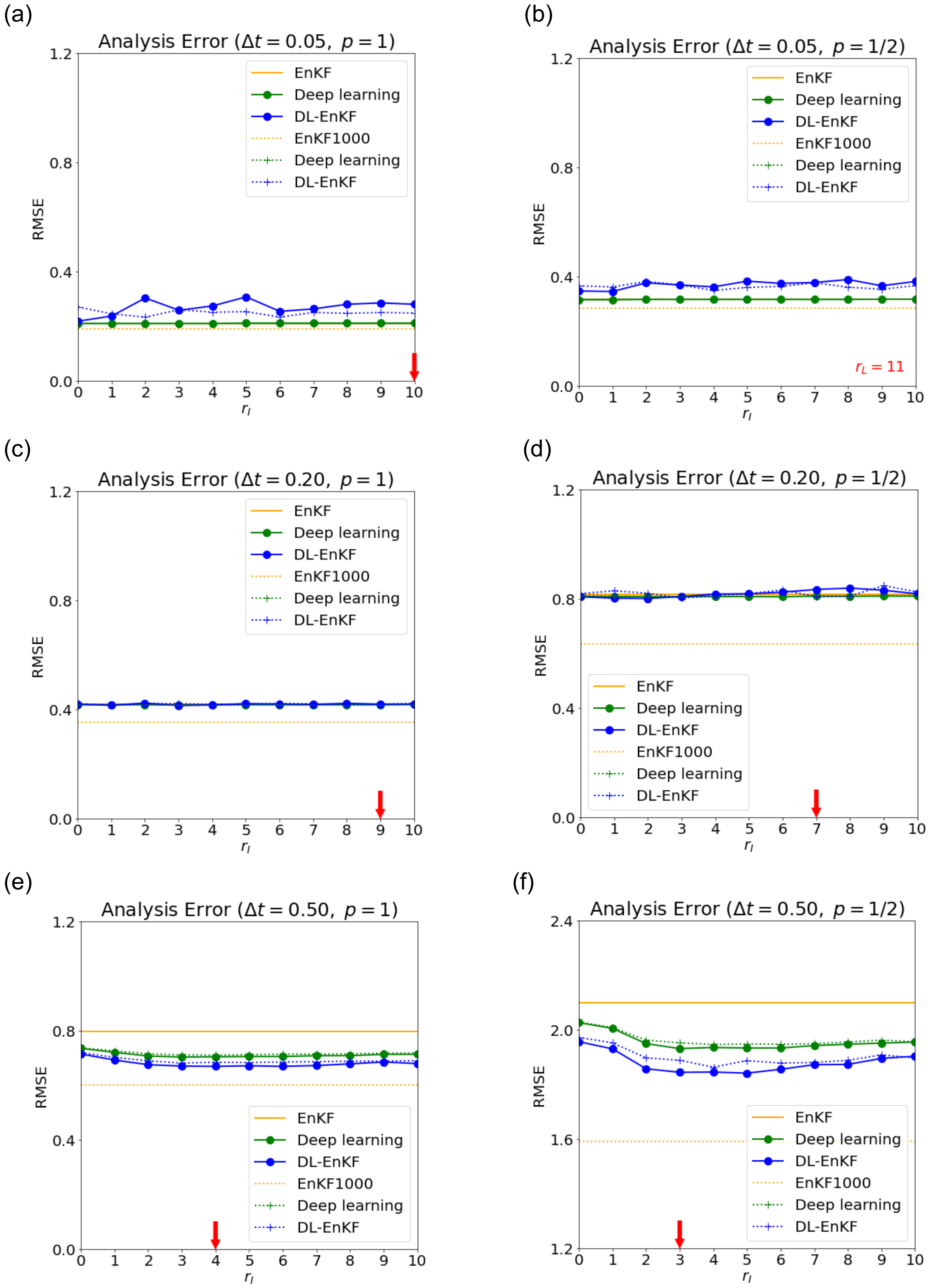


Fig. 8 Comparison of RMSEs of EnKF (orange lines), deep learning (green lines), and DL-EnKF (blue lines) for Exp-PA (solid lines) and Exp-PB (broken lines). An orange broken line indicates the RMSE of EnKF1000 analysis used for training in Exp-PB. The RMSEs are plotted against the input radius, and a red arrow indicates the optimal localization radius of EnKF. The observation time interval and the probability of observations are (a) 0.05 and 1, (b) 0.05 and 1/2, (c) 0.20 and 1, (d) 0.20 and 1/2, (e) 0.50 and 1, and (f) 0.50 and 1/2, respectively.

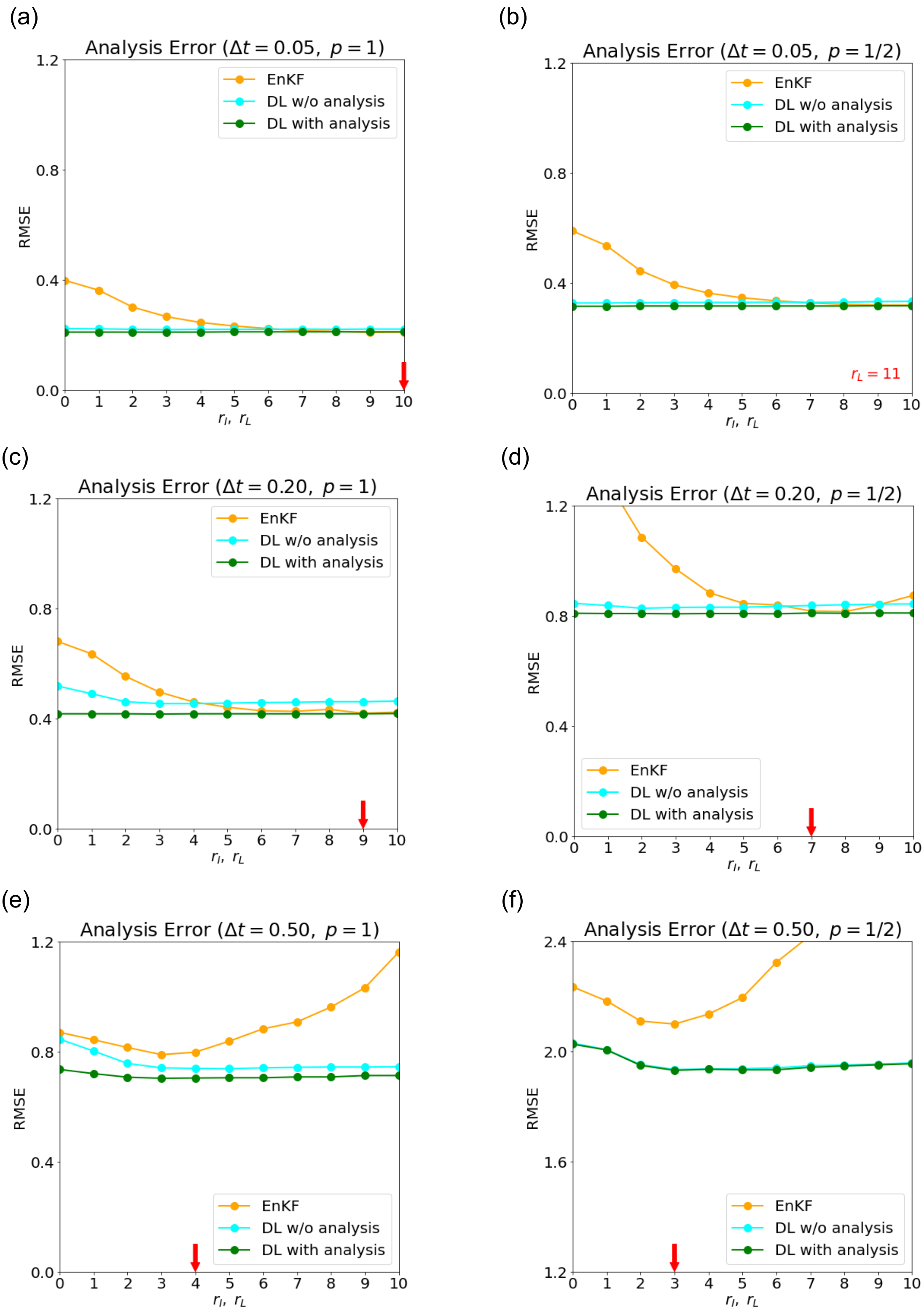


Fig. 9 Comparison of RMSEs of EnKF (orange line), deep learning not including EnKF analysis in input (cyan line), and deep learning including the EnKF analysis in input (green line) for Exp-PA. The RMSE of EnKF is computed for each localization radius. The abscissa is the input radius for deep learning and the localization radius for the EnKF. A red arrow indicates the optimal localization radius. The observation time interval and the probability of observations are (a) 0.05 and 1, (b) 0.05 and 1/2, (c) 0.20 and 1, (d) 0.20 and 1/2, (e) 0.50 and 1, (f) 0.50 and 1/2, respectively.

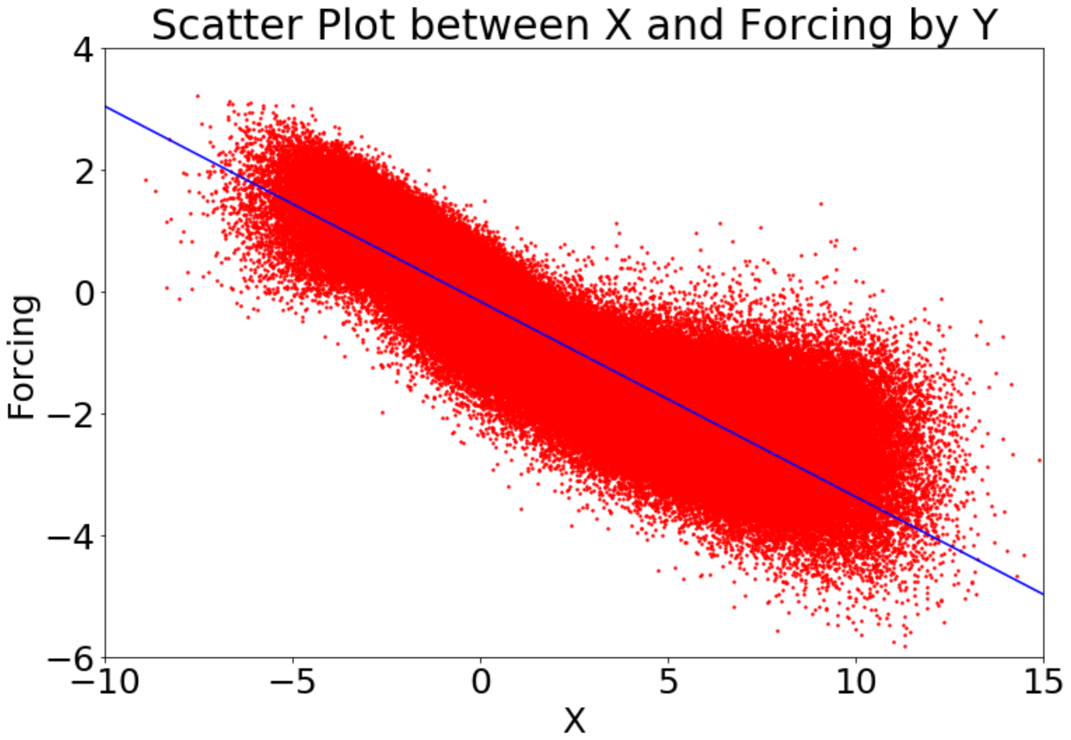


Fig. 10 Scatter plot between large-scale variables (abscissa) and large-scale forcing by small-scale variables (ordinate) of the two-scale Lorenz 96 model. A solid line is the result of linear function fitting.

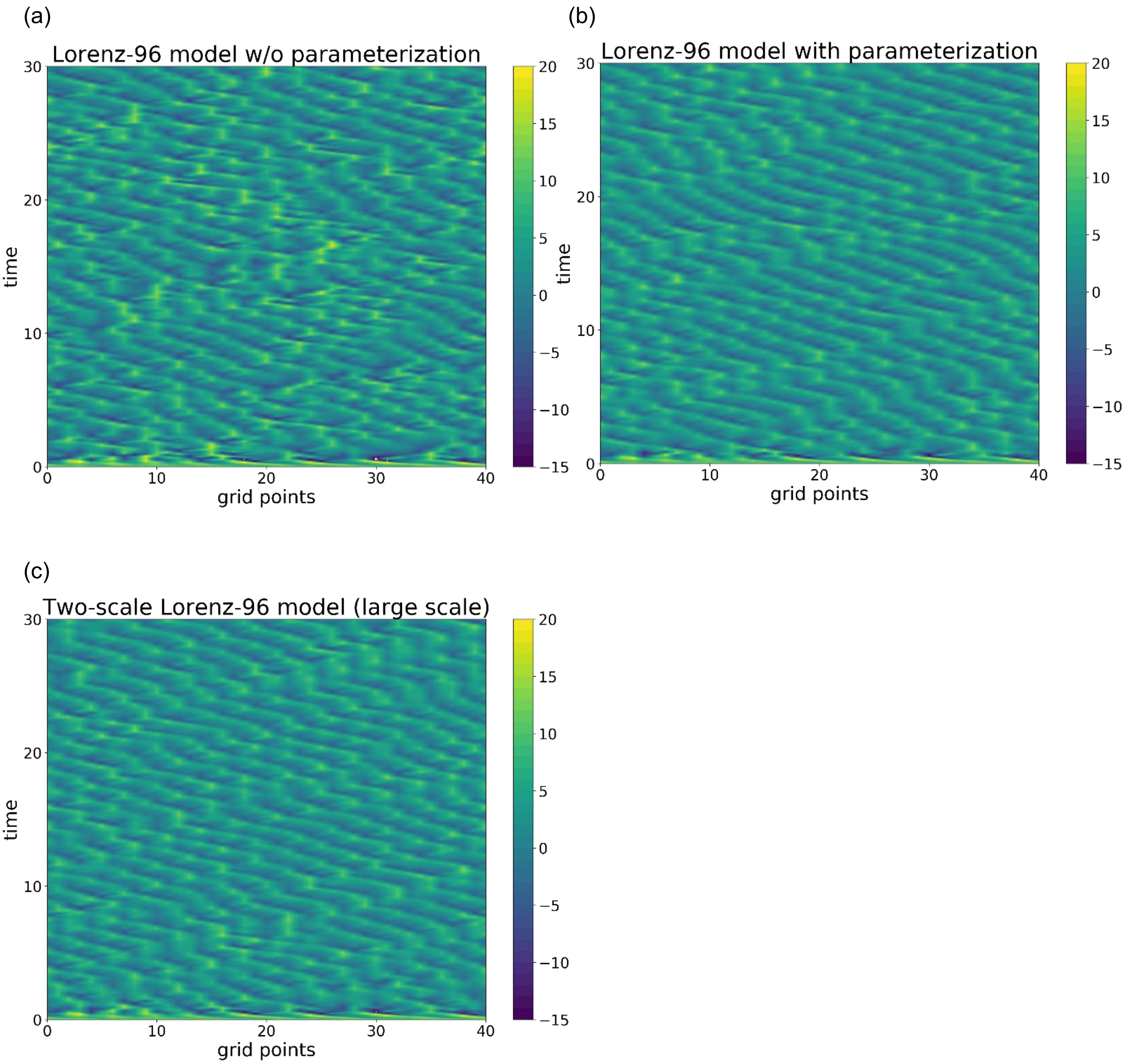


Fig. 11 Hovmöller diagrams of (a) the Lorenz 96 model, (b) the parametrized Lorenz 96 model, and (c) large-scale variables of the two-scale Lorenz 96 model.

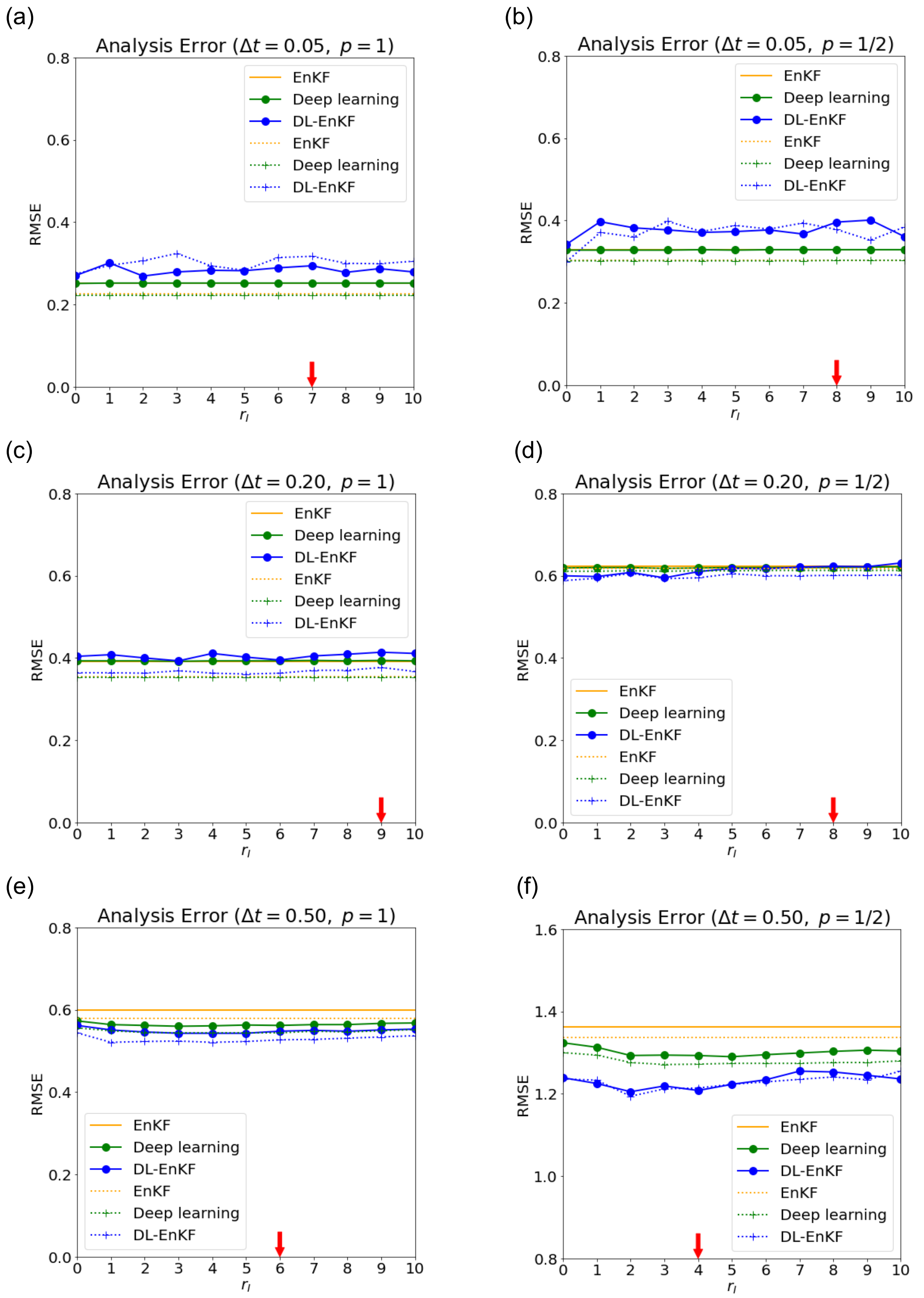


Fig. 12 Same as Fig. 8 except for Exp-IA (solid lines) and Exp-IB (broken lines) and that an orange broken line indicates the RMSE of EnKF using the two-scale Lorenz 96 model.

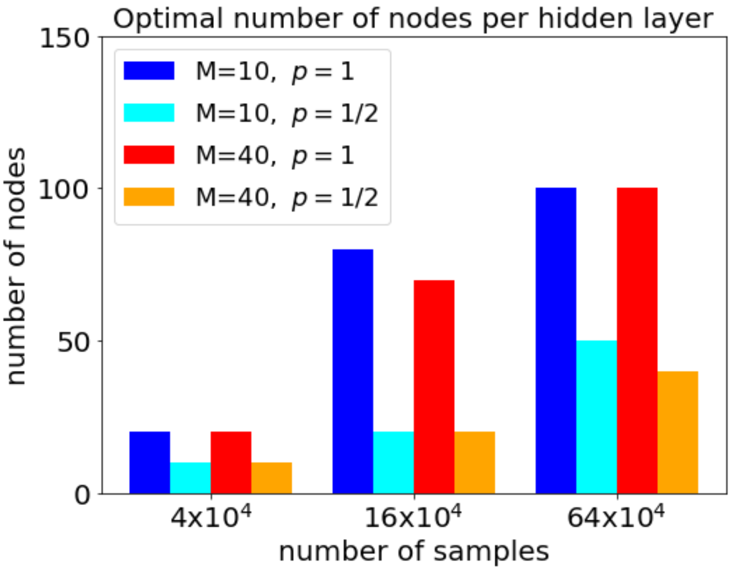


Fig. A1 The optimal number of nodes per hidden layer of a DNN with 5 hidden layers for the time interval between observations of 0.50. The abscissa is the number of samples. Blue and cyan bars are for the EnKF ensemble size of 10 for the probability of observations 1 and 1/2, respectively. Red and orange bars are for the EnKF ensemble size of 40 for the probability of observations 1 and 1/2, respectively.

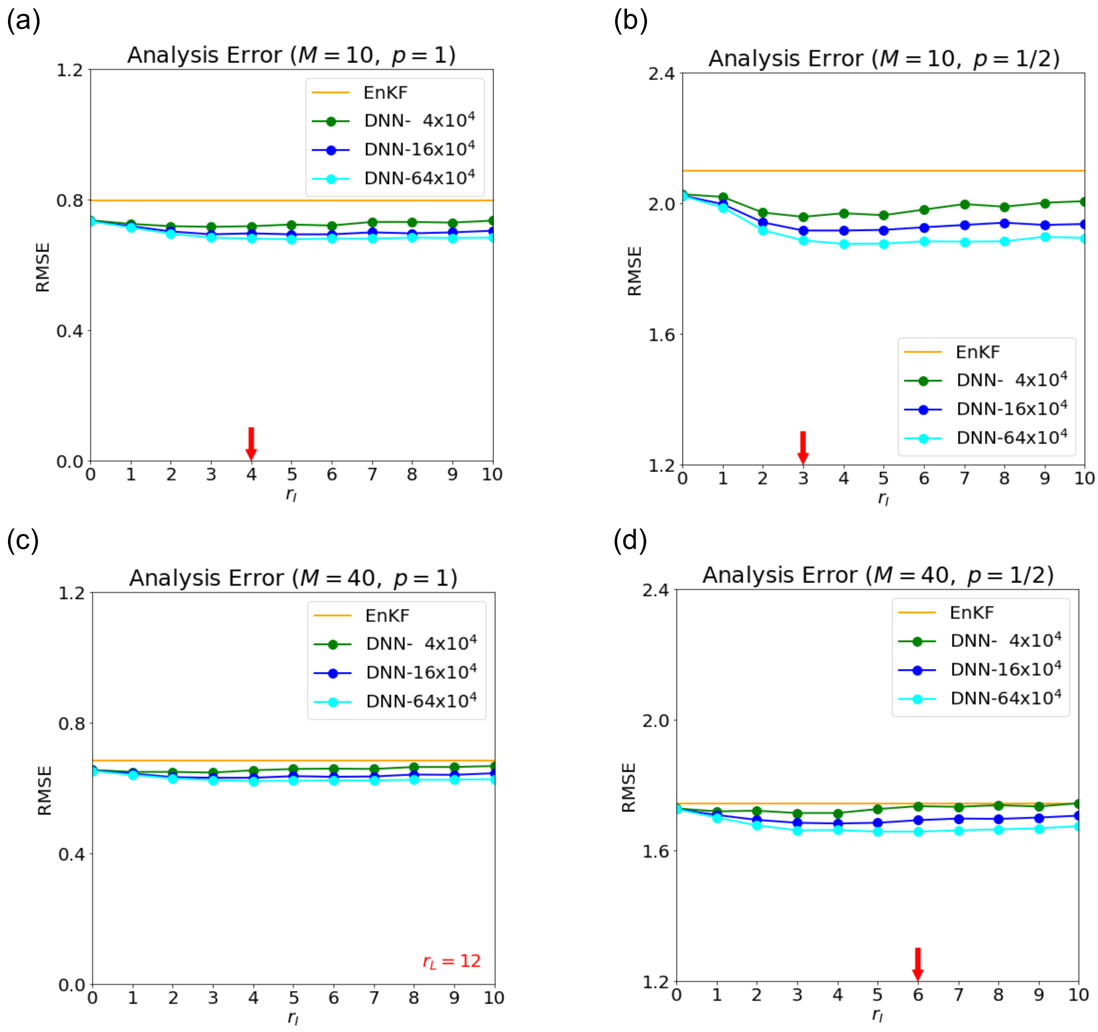


Fig. A2 Comparison of RMSE between EnKF (orange lines) and the output of a DNN with the number of samples of 40 000 (green lines), 160 000 (blue), and 640 000 (cyan) for the observation time interval of 0.50. The ensemble size of EnKF and the probability of observations are (a) 10 and 1, (b) 10 and 1/2, (c) 40 and 1, (d) 40 and 1/2, respectively. The RMSEs are plotted against the input radius, and a red arrow indicates the optimal localization radius of EnKF.

List of Tables

Table 1 Models used in the training and test phases in the experiments.

(a) Perfect model experiments

	Exp-PA		Exp-PB	
	Training	Test	Training	Test
Target	L	L	Analysis	L
Observations	L	L	L	L
Forecast ensemble	L	L	L	L

L: Lorenz 96 model

(b) Imperfect model experiments

	Exp-IA		Exp-IB	
	Training	Test	Training	Test
Target	P	T	T	T
Observations	P	T	T	T
Forecast ensemble	P	P	T	T

P: parameterized Lorenz 96 model, T: two-scale Lorenz 96 model

Table 2 Architecture and training of the feedforward neural network.

No. of nodes of input layer	$3(2r_I + 1)$ for $p = 1$ $4(2r_I + 1)$ for $p = 1/2$
No. of hidden layers	5
No. of nodes per hidden layer	5, 10, or 20 (optimized)
No. of nodes of output layer	1
Activation function	ReLU
Loss function	Sum of squared error
Gradient descent method	Adam*
Learning rate	0.01 to 0.0001 (linear decay)
No. of samples	40 000
Mini-batch size	100
No. of epochs	100

*: Kingma and Ba (2014)