Journal of the Meteorological Society of Japan

EARLY ONLINE RELEASE

This is a PDF of a manuscript that has been peer-reviewed and accepted for publication. As the article has not yet been formatted, copy edited or proofread, the final published version may be different from the early online release.

The DOI for this manuscript is
DOI:10.2151/jmsj.2023-005
J-STAGE Advance published date: November 1st, 2022
The final manuscript after publication will replace the preliminary version at the above DOI once it is available.

A Machine Learning Approach to the Observation Operator for Satellite Radiance Data Assimilation

Jianyu LIANG[1]*Data Assimilation Research Team, RIKEN Center for Computational Science, Kobe, Japan*

Prediction Science Laboratory, RIKEN Cluster for Pioneering Research, Kobe, Japan

Koji TERASAKI

Data Assimilation Research Team, RIKEN Center for Computational Science, Kobe, Japan
Prediction Science Laboratory, RIKEN Cluster for Pioneering Research, Kobe, Japan
Now at "Meteorological Research Institute, Japan Meteorological Agency, Ibaraki, Japan"

Takemasa MIYOSHI

Data Assimilation Research Team, RIKEN Center for Computational Science, Kobe, Japan
Prediction Science Laboratory, RIKEN Cluster for Pioneering Research, Kobe, Japan
RIKEN interdisciplinary Theoretical and Mathematical Sciences (iTHEMS), Wako, Japan
Application Laboratory, Japan Agency for Marine-Earth Science and Technology (JAMSTEC), Yokohama, Japan
Department of Atmospheric and Oceanic Science, University of Maryland, College Park, MD, USA"

September 1, 2021

----

1) Corresponding author: Jianyu Liang, RIKEN Center for Computational Science. 7-1-26, Minatojima-minami-machi, Chuo-ku, Kobe, Hyogo 650-0047, Japan

Email: jianyu.liang@riken.jp

Abstract

The observation operator (OO) is essential in data assimilation (DA) to derive the model equivalent of observations from the model variables. In the satellite DA, the OO for satellite microwave brightness temperature (BT) is usually based on the radiative transfer model (RTM) with a bias correction procedure. To explore the possibility to obtain OO without using physically based RTM, this study applied machine learning (ML) as OO (ML-OO) to assimilate BT from Advanced Microwave Sounding Unit-A (AMSU-A) channels 6 and 7 over oceans and channel 8 over both land and oceans under clear-sky conditions. We used a reference system, consisting of the nonhydrostatic icosahedral atmospheric model (NICAM) and the local ensemble transform Kalman filter (LETKF). The radiative transfer for TOVS (RTTOV) was implemented in the system as OO, combined with a separate bias correction procedure (RTTOV-OO). The DA experiment was performed for one month to assimilate conventional observations and BT using the reference system. Model forecasts from the experiment were paired with observations for training the ML models to obtain ML-OO. In addition, three DA experiments were conducted, which revealed that DA of the conventional observations and BT using ML-OO was slightly inferior, compared to that of RTTOV-OO, but it was better than the assimilation based on only conventional observations. Moreover, ML-OO treated bias internally, thereby simplifying the overall system framework. The proposed ML-OO has

limitations due to (1) the inability to treat bias realistically when a significant change is present in the satellite characteristics, (2) inapplicability for many channels, (3) deteriorated performance, compared with that of RTTOV-OO in terms of accuracy and computational speed, and (4) physically based RTM is still used to train the ML-OO. Future studies can alleviate these drawbacks, thereby improving the proposed ML-OO.

Keywords: satellite radiance data assimilation; machine learning; neural network; observation operator; forward operator

1. Introduction

Data assimilation (DA) is a combination of model simulations and observations. DA provides an optimal estimate of the initial condition, thereby improving the forecast. Because satellite observations provide dominant observational information for global numerical weather predictions (NWP) (Eyre et al. 2020), assimilating them is important. In this context, the Advanced Microwave Sounding Unit-A (AMSU-A) is a multichannel microwave radiometer, which is sensitive to the temperature profile of the atmosphere. It has been already used for improving NWP performance (e.g. Miyoshi et al. 2010; Terasaki and Miyoshi 2017 (hereafter simply "TM17")). For atmospheric data assimilation, the model space and the observation space are generally different because the locations of the observations do not ideally coincide with the model grid points, and the observed variables may not be the same as the model variables. Due to this, to compare the model state and observations, an observation operator (OO) (e.g., a forward model) is required. When satellite radiances are used as observations, the OO has two primary purposes. First, it performs a horizontal interpolation of the model variables at the model grids to the observation locations. Second, at each observation location, the simulated (synthetic) radiance is calculated using a vertical profile of the model variables (Kalnay 2002, p. 161). In this study, we elucidated only the second aspect, which requires knowledge about the relationships between the model variables and satellite radiances. To fulfill this, there are two approaches: physically based RTMs (RTM-OO) and machine learning (ML) models (ML-OO).

From the RTM-OO perspective, the line-by-line (LBL) RTM is an accurate and flexible RTM, applicable over a full spectral range. Such ability lays the foundation for numerous radiative transfer applications (Alvarado et al. 2013). As the calculation of the RTM needs to be fast for data assimilation, it is generally not recommended to use LBL RTM for RTM-OO. Therefore, faster RTMs have been developed. For instance, in the radiative transfer for TOVS (RTTOV) (Saunders et al. 2018), the layer optical depth for specific gas and channel is parameterized in terms of layer mean atmospheric variables (Saunders et al. 2018; Hocking 2019). To obtain the regression coefficients of these predictors, layer-to-space transmittances at high spectral resolutions computed from LBL RTMs using a variety of atmospheric profiles, have been used. Notably, fast RTMs such as RTTOV are the most widely used observation operators for satellite radiance data assimilation. The biases between the model equivalent of observations from the RTM and the actual satellite observations can emerge due to various effects including the calibration problem of the instrument, temporal change of instrument characteristics, preprocessing of the data, the RTM inaccuracies, and the bias in the model field (Derber 1998; Dee 2004; Harris and Kelly 2001). A bias correction procedure is required for two common types of biases: scan bias and airmass bias (Dee 2004; Harris and Kelly 2001). The former is related to the satellite scan position and latitude, and the latter is related to the state of the atmosphere (Harris and Kelly 2001). Note that some NWP centers use only the scan positions for scan bias correction. Scan bias can be estimated offline (Harris and Kelly 2001) or online (TM17). The airmass bias can be estimated offline (Harris and Kelly 2001). It can also be estimated online adaptively in the variational data assimilation using the variational bias correction method (VarBC) (Derber 1998; Dee 2004). Furthermore, an equivalent method in ensemble data assimilation has been previously proposed (Miyoshi et al., 2010; TM17).

From the ML-OO perspective, MLs are efficient for identifying complex statistical relationships within the data. The application of artificial intelligence and ML in Earth and atmospheric studies has become increasingly popular in research, whereas a review of such applications can be found in Boukabara et al. (2021). ML can be used to reduce the high computational cost in applications that involve complicated physical processes. For instance, ML can be used to emulate and accelerate parametrization schemes in the NWP (Chantry et al. 2021; Pal et al. 2019; Krasnopolsky et al. 2008). Likewise, the ML method can be applied to build an ML-OO. In general, there are primary two ways to train an ML-OO.

Firstly, one can use the input and output data from the RTM. The development of ML-OO using the RTM requires various input variables, covering the full physical parameter space with sufficient resolution. Such inputs are provided to the RTM to generate the corresponding synthetic radiances. The input variables and output radiances are paired to train an ML model to obtain the ML-OO. If the original RTM is computationally expensive, ML-OO can generally reduce the computational cost, while retaining sufficient accuracy. For instance, a look-up table (LUT) can be generated by the ML emulator to be further used for retrieval purposes (Rivera et al. 2015). In Scheck (2021), slow RTM for visible satellite images was emulated using a neural network (NN).

Secondly, it is possible to develop the ML-OO without the use of RTM. Compared to the method using RTM, such an approach relies on satellite-observed radiance data instead of synthetic radiance data from the RTM. Being combined with the model state, satellite radiance can be used to train the ML model. To represent the actual relationship between the model state and the satellite radiance, the model state used for training the ML-OO must be good enough. To this end, we suggest that analyses or short-term forecasts after analyses from data assimilation and reanalysis data can be used. Kwon et al. (2019) have previously used atmospheric reanalysis data to provide forcing for a land surface model to generate synthetic snow depth. Then synthetic snow depth and observed radiance were utilized to train a support vector machine model. They showed that ML-OO is computationally more efficient than RTM. To the best of our knowledge, the analyses or short-term forecasts from data assimilation were never utilized to combine with the satellite-observed radiance to train an ML model.

Our goal is to build an ML-OO without using the RTM. However, this study is only the first step since we still used RTM to assimilate the satellite radiance to generate better short-term forecasts. As mentioned earlier, the bias between synthetic radiance and satellite-observed radiance should be addressed. Zhou and Grassotti (2020) used ML to address the radiometric bias to improve the satellite retrievals. Rodríguez-Fernández et al. (2019) have previously trained the ML model using the Soil Moisture and Ocean Salinity (SMOS) brightness temperature (BT) from observations as the input and soil moisture (SM) from the model as the output. They found no global bias between the retrieved SM predicted from the ML and modeled SM. Similarly, if ML-OO for satellite radiance is built using the model state and the observed radiance, the bias between the simulated radiance from ML-OO and the observed radiance would be assumingly low. As one of the objectives of the study, we evaluate this surmise by our analysis. Moreover, we compare ML-OO with the RTM-OO. Lastly, we discuss how our preliminary study can be extended to broader applications.

The remainder of this paper is organized as follows. The materials and methods are described in Section 2. Sections 3 and 4 present the experimental setup and results. Finally, a discussion and summary are presented in Sections 5 and 6, respectively.

2. Materials and Methods

2.1 Data assimilation system

In this study, we used the nonhydrostatic icosahedral atmospheric model (NICAM) (Satoh et al. 2014) and the local ensemble transform Kalman filter (LETKF) (Hunt et al. 2007) to conduct data assimilation experiments. The configuration of the system (NICAM-LETKF) mostly followed that of the TM17. In this study, only a few important aspects of the system related to this study are presented. The horizontal resolution of the NICAM model was 112 km. There were 78 vertical levels (38 levels in TM17) up to the height of ~50 km. The NICAM-LETKF has an observation operator for assimilating the satellite radiance. The observation operator horizontally interpolates model variables in the first guess from the model grids to the observation locations. These variables include pressure, temperature, specific humidity, surface pressure, 2-m temperature, surface (skin) temperature, 2-meter specific humidity, and 10-m zonal/meridional winds. After the interpolation, at each observation location, the interpolated model variables combined with other variables were utilized in RTTOV (version 12.2) to calculate the model equivalent of the brightness temperature. A complete list of the variables required by the RTTOV can be found in Hocking (2019). The NICAM-LETKF system uses an online bias correction method to correct scan bias and airmass bias (TM17). The biases were estimated adaptively during the data assimilation and subtracted from the observed BT before the analysis. The input variables (predictors) for the airmass bias included the integrated weighted lapse rate (IWLR) at two layers: 1000−200 hPa and 200−50 hPa, surface temperature, and the inverse of cosine of satellite zenith angle. For brevity, we use the RTTOV-OO to indicate the observation operator based on RTTOV combined with an online bias correction method.

The NICAM-LETKF system contained 64 ensemble members. The relaxation to prior spread (RTPS) method was applied for covariance inflation (Whitaker and Hamill 2012; Kotsuki et al. 2017). Covariance localization based on the Gaussian function was applied with a standard deviation σ = 250 km in the horizontal and 0.4 in the vertical natural-log-pressure coordinate, but the localization function was replaced by zero beyond $2\sqrt{10/3}\sigma$. Note that vertical localization was not used for AMSU-A BT. Its impact on performance will be investigated in future studies.

2.2 Observation data

Observations were assimilated every 6-h (Fig. 1a). At each analysis time point, the observations within the $\pm 3\,h$ time window were assimilated. Overall, there were seven observation files (time slots) in an analysis time window. One file in each time slot contained observations ± 30 minutes. After finishing an analysis, we forecasted 9 hours so that there were observations within the $\pm 3\,h$ time window at the next analysis time. This process was continued until the end of the experimental period.

The observations included the NCEP ADP Global Upper Air and Surface Weather Observations dataset (NCEP PREPBUFR). This dataset includes records from radiosondes, wind profilers, aircraft, land surface observations, marine observations, atmospheric motion vectors (AMVs), and sea surface winds from satellite scatterometers. The satellite radiance data were represented by BT obtained from the AMSU-A instruments onboard the NOAA-15, NOAA-18, NOAA-19, METOP-A, and METOP-B satellites. Note that in this study, the term ‘conventional observations’ indicates the observations from the NCEP PREPBUFR dataset. As our model was vertically constrained by the 50 km height in this study, we assimilated only the channel numbers 6, 7, and 8. The channels 9 and beyond, which are sensitive to the stratosphere and mesosphere, were not assimilated due to this reason. Moreover, the lower channels were not assimilated due to their sensitivity to the lower troposphere and the Earth's surface, where the quality control in this study was rather simplistic to handle the data (TM17). Not all three channels were assimilated for some satellites (Table 1). The standard deviation of the observation error used for data assimilation was set at 0.3 K for all the used channels.

Before the data assimilation, the observations were preprocessed, where data thinning, quality control, and gross error checks were applied to the observations. The observation errors in data assimilation include measurement and representation errors (Janjić et al. 2018). The observation errors are correlated in terms of space and time. Particularly, for satellite radiances, correlated errors may be present between channels. However, it is challenging to identify and implement a full observation error-covariance matrix for such data. Therefore, spatial thinning was implemented in this study to reduce potential spatial correlations. For thinning of the AMSU-A observations, we selected the nearest observations from every grid point of the uniform virtual horizontal grids with the 250-km resolution by following the JMA's setting from Okamoto et al. (2005). Note that a thorough examination of the thinning distance effects is beyond the scope of this study. Furthermore, the quality control for assimilating AMSU-A was applied after thinning. The observations from channels 6 and 7 over the land were completely filtered out. Over the oceans, observations from these two channels were filtered out at the liquid water path (LWP) >0.12 kg kg^{-1} and 0.15 kg kg^{-1}, respectively. The LWP was calculated using channels 1 and 2 from AMSU-A (Grody et al. 2001). LWP was utilized to remove cloud- and raincontaminated observations by following the method of Bormann et al. (2012). Channel 8 was assimilated without any quality control because the peak height of the weighting function is higher, which implies that this channel is less affected by clouds and rain. Finally, we performed a gross error check to remove the data with a large observation-minus-first-guess departure. Specifically, when the departure was greater than three times the standard deviation of the observation error, the data were filtered out. To train the ML model, the same quality control was applied to the AMSU-A observations. However, data thinning of AMSU-A observations was not applied because more data were required to train the ML model. As shown in Fig. 1b, after the quality control, all the data over land from channels 6 and 7 were excluded, while there were data over land and oceans from channel 8.

2.3 Machine learning method

As discussed in the introduction, the observation operator interpolates and converts the model variables into the model equivalent of the observation. To build a good observation operator, we should ideally use the values of the model variables and observed variables that are close to the true state of the atmosphere. In this context, data assimilation is necessary to obtain such model variable values. On this basis, we opted to use the model forecasts after assimilating conventional observations and AMSU-A (using RTTOV-OO) as the input data for building the ML model. At every analysis time slot (00 UTC, 06 UTC, 12 UTC, 18 UTC), a 9-h forecast was performed, thereby yielding the hourly forecast data from

3 to 9 h after the analysis. Instead of using the forecasts at all 7 h as the input to train the ML model, we used only the forecasts from 3 to 6 h after each time slot of the analysis. Fundamentally, they were closer to the analysis, while the satellites also had global coverage in this time window. Note that the experiments that produced model forecasts from the NICAM-LETKF will be explained below in Section 3.

For each atmospheric column, we would like to use ML-OO to predict BT according to the related model variables in the same column. On this basis, the development of the ML model implied that the input data (model variables) and output data (BT) in the same row of the dataset should originate from the same atmospheric column. As observation locations differed from the model grids, we interpolated the model variables at the model grids to the observation locations. Note that most input variables of the ML model were the same as RTTOV-OO (Table 2). For the three-dimensional variables, such as pressure, temperature, and specific humidity, each layer was considered as a feature in the ML model. However, the specific humidity above the NICAM model level 40 (~200 hPa height) was constant and almost 0. They were excluded as features because constant inputs do not contribute to the input-output relationship (Krasnopolsky et al. 2008).

Moreover, we added two predictors for the biases to the input variables because our initial idea was to ensure that the ML model can capture biases. The sections below briefly describe how biases had been treated by previous studies and explain how the ML treats biases in this study. In the offline bias correction method (Harris and Kelly 2001; Dee 2004), the bias corrections were precomputed using historical data using the following steps: (1) the scan bias coefficient $b^{scan}(\theta, \emptyset)$ is obtained. It is a function of scan angle θ, and latitude $\emptyset$. Then, (2) the scan bias is removed from the departures: $y - h(x_b) - b^{scan}(\theta, \emptyset)$, where y is the observation, x_b is the model background (usually in the vicinity of the radiosonde to ensure accuracy), and $h()$ is the RTM. $y - h(x_b) - b^{scan}(\theta, \emptyset)$ is then fit by a linear regression model. The linear regression model is the airmass bias correction term in this case.

$$y - h(x_b) - b^{scan}(\theta, \emptyset) = \beta_0 + \sum_{i=1}^{N} \beta_i p_i\,(x_b) + \widetilde{e}\,,\ <\widetilde{e}> \ = 0 \quad (1)$$

where N is the number of predictors, p_i (i = 0, … N) are the predictors (state-dependent), β_i (i = 0,…N) are the coefficients of the predictors, and $\widetilde{e}$ is the residual error. As linear regression always passes the center of the data, the expectation of the residual errors is zero. The coefficients of scan bias and airmass bias are stored in the file and were used in the data assimilation. Eq. (1) can be changed to:

$$y' = y - b^{scan}(\theta, \emptyset) - \beta_0 - \sum_{i=1}^{N} \beta_i p_i\,(x_b) = h(x_b) + \widetilde{e}\,,\ <\widetilde{e}> \ = 0 \quad (2)$$

Therefore, before assimilating an observation y, the scan bias and the airmass bias are removed from the observations to obtain a 'bias-corrected' observations y', which are subsequently assimilated. In the end, because $<\widetilde{e}> \ = 0$, there is no bias between the simulated radiance $h(x_b)$ and the 'bias-corrected' radiance y'.

Furthermore, the constant coefficients of the predictors for the airmass bias correction can be updated adaptively during data assimilation. In VarBC, the coefficients of the predictors are added to the model state to form an augmented vector. The original observation operator is modified using the airmass bias term. The minimization of the cost function updates the augmented vector and the coefficients of the predictors, as shown by equations 10 to 14 of Dee (2004). The ensemble-based VarBC in the ensemble data assimilation also updates the coefficients adaptively based on the formulas from VarBC. Note that for VarBC and ensemble-based VarBC, the cost function contains two terms: distance to the background and distance to the observations. Thus, its minimization cannot ensure the minimization of the bias. However, some previous studies, mentioned in the introduction, have demonstrated the efficiency of these methods.

Being inspired by the methods above, we argue that ML-OO can also handle the bias. Eq. 1 can be rewritten as:

$$y = h'(x_b, \theta, \emptyset, p) + \tilde{e}\,, \; <\tilde{e}> \; = 0 \tag{3}$$

where $h'(x_b, \theta, \emptyset, p) = h(x_b) + b^{scan}(\theta, \emptyset) + \beta_0 + \sum_{i=1}^{N} \beta_i p_i \,(x_b)$

If the observation y on the left-hand side of Eq. 3 and $x_b, \theta, \emptyset, p$ on the right-hand side are given, ML can be used to find a function to fit the observations.

$$y = h_{ml}(x_b, \theta, \emptyset, p) + e_{ml}, \tag{4}$$

where h_{ml} is ML-OO, and e_{ml} is the residual error of the ML model. $bias = < e_{ml} >$.

The ML algorithm minimizes the mean squared error (MSE). The MSE can be decomposed into the variance of the error and the square of the bias (see Appendix A for the derivation).

$$E([y - h_{ml}(x_b, \theta, \emptyset, p)]^2) = Var[y - h_{ml}(x_b, \theta, \emptyset, p)] + bias^2 \quad (5)$$

Because the variance of the error is positive, the square of the bias is smaller than the MSE. If the MSE is reasonably small after the minimization, the absolute value of the bias may be small enough. However, to ensure the performance of the ML-OO, the MSE and bias should be evaluated using the test data after training and before data assimilation. If both MSE and bias are low enough, the ML model can be used as an observation operator.

In the last step, once the ML-OO is obtained, the data assimilation can be formalized as:

$$x^a = x^b + K[y - h_{ml}(x^b, \theta, \emptyset, p)] \quad (6)$$

where x^a is the analysis, x^b is the model background, and *K* is the optimal weight matrix. Notably, compared with RTTOV-OO, for ML-OO, the original observations can be assimilated directly without subtracting the bias correction terms.

Note that the selection of the predictors in ML-OO was based on the TM17 paper (Table 2). Specifically, in TM17, IWLR, surface temperature, and inverse of the cosine function of the satellite zenith angle were applied as the predictors for air mass bias correction. In this study, IWLR was not explicitly added because the vertical profiles of pressure and temperature in the input can fundamentally reflect IWLR. Surface temperature and satellite zenith angle are required by the RTM (Saunders et al., 2018). Therefore, they are important for both the radiative transfer process and air mass bias correction. Latitude and satellite scan angle were added to the input variables for the scan bias correction. Note that latitude is also used in RTTOV to calculate the effects of Earth's curvature on the atmospheric path (Hocking 2019). Both latitude and satellite scan angles have been previously applied for scan bias correction by Zhou and Grassotti (2020). They have used ML to correct the bias between the simulated radiances and satellite observations. It is important to note that BT estimates from channels 6, 7, and 8 are not sensitive to the radiation from the surface. However, we included the surface variables in the input of ML-OO. To remind, we aim to use similar input variables as RTTOV-OO in TM17 to make the comparison feasible. Moreover, they are also the input variables for RTTOV. Finally, these variables are useful for some other channels that are more sensitive to the lower atmosphere. Therefore, we standardized the same set of input variables for all the channels.

Before feeding the data into the ML model, other preprocessing steps were performed. As the specific humidity was skewed toward lower values, we used the log function to transform it to the normal distribution. For the same reason, the pressure was transformed using the log function. The satellite zenith angle was expressed as $1/\cos(\theta)$, like in TM17. Each satellite dataset was separated into a training set (80%) and a test set (20%). Finally, the input and output data were standardized to zero mean and unit variance to facilitate the fast convergence of the ML during the training.

Fully connected deep neural networks (DNN) were used in this study, as shown in Fig. 2. We built different DNNs for each channel and each satellite because the number of collocated observations from the same channel for different satellites is small. Moreover, given the channel-related quality control methods, different channels from the same satellite may have a small number of collocated locations. For example, there were no data from channels 6 and 7 for land, whereas some data were available from channel 8. There were 205 units in the input layer that matched 205 features in the input data. The output layer had only a single unit that corresponded to one channel. The optimizer we used was a gradient descent algorithm known as 'Adam', which is well suited for solving problems that are large in terms of data and/or parameters (Kingma and Ba 2014). We used the rectified linear unit (ReLU) (Glorot et al. 2011) in the hidden layers and linear regression in the output layer. The batch size is the number of training examples used in one iteration. The number is typically selected to be between one and a few hundred (Bengio 2012). For simplicity, it was fixed at 512 in this study. The following hyperparameters were tuned for each DNN model: number of hidden layers, number of units in one hidden layer, and learning rate. For each combination of the above hyperparameters, a DNN was constructed, and it was trained using 80% of the training set (the training set itself was 80% of all data) and evaluated using 20% of the training set (validation set). The validation set was applied for an early stopping to prevent the overfitting of the model. In other words, if the loss function in the validation set starts to increase, overfitting occurs. In this study, we used the mean squared error (MSE) as the loss function. If the MSE of the validation set did not decrease for five consecutive epochs, we stopped the training. During the training, the 'KerasTuner' software (O'Malley et al. 2019) was utilized to automatically conduct a random search for the best combination of hyperparameters for each channel and each satellite. Before the random search, the search spaces for the hyperparameters were set as follows. The numbers of hidden layers were 2, 3, and 4. The unit numbers for each hidden layer were 250, 300, 350, and 400. The learning rates were 10^{-6} and 10^{-5}.The maximum number of random searches was 25. The combination of hyperparameters that produced the best performance on the validation set was selected for each DNN (Table 3). DNNs were evaluated by comparing the predicted and true values in the test set (Table 3). The coefficient of determination (R^2) between the predicted and true values was ~1. The absolute values of the biases were < 0.02 K, while the root mean square errors (RMSEs) were < 0.4 K. As mentioned in the introduction, minimizing the MSE using the ML optimization algorithm does not guarantee the minimization of bias. However, the test results revealed low bias. Therefore, the performance of the DNNs was reasonably good, and they were hereafter used as ML-OO in our experiment. A linear regression model was applied to the same dataset for further comparison (Table 3). The RMSEs from the linear regression model were all larger than 1, which was higher than those from the DNN models, while the R^2 score was also lower. Overall, ML was better than the linear regression approach for solving this problem.

3. Experiments

Several data assimilation experiments were conducted to produce data for training the ML model and for evaluating its performance (Table 4). The experiments were categorized into two groups: experiments for training in 2015, and experiments for testing in 2016. The initial conditions of the ensemble were drawn at the same local time on different days from a single forecast from January to March in 2015. Since they differed from the true state of the atmosphere, we needed to spin up the model for one month using data assimilation. As discussed in Section 2, a model state close to the true state of the atmosphere is required to build a good ML-OO. Thus, we assimilated the conventional observations as well as AMSU-A BT using RTTOV-OO in Experiments A and B. Note that in this way, we have implicitly used the information from RTM to build the ML-OO. We will discuss how to build ML-OO without using RTM in the discussion section. Note that an online bias correction method was applied during the experiments. Experiment A was designed to spin up the model. At the end of January 2015, the model state would be close to the true atmosphere. After finishing the spin-up, the data assimilation was continued in February 2015 to generate the model forecasts for training the ML model (Experiment B). After finishing the experiment, the model outputs at the model grids in Experiment B were interpolated to the observation locations. The observations were those without the data thinning and with quality control, as described in section 2.2 (Fig. 1b). After the interpolation, the (model) first guess at the observation locations, and the corresponding AMSU-A BT were paired to train the ML model (experiment C).

After the ML-OO was built, we evaluated its performance for the same month the following year in 2016. In general, ML can better generalize to new data if it captures more possible combinations and wider ranges of variable values during training. As only one-month data were used for the training, we also evaluated its performance in the same month of the following year. On 01 January 2016, we used the same initial conditions as in Experiment A. Due to this, we needed to spin up the model using the RTTOV-OO to assimilate the AMSU-A BT and conventional observations (Experiment D). At the end of January 2016, the ensemble members were used for the following data assimilation experiments in February. Experiment E represented the continuation of Experiment D, where we assimilated the conventional observations and AMSU-A BT using RTTOV-OO. In Experiment F, the same observations were assimilated using the ML-OO. Note that no online bias correction was provided for Experiment F because bias correction was included in the ML-OO. The results from Experiments E and F were compared to evaluate the performance of ML-OO compared to RTTOV-OO. Finally, we conducted experiment G, in which we assimilated only the conventional observations. We compared E against G and F against G to estimate the impact of assimilating the AMSU-A BT using either RTTOV or ML as the observation operator.

4. Results

The ML models were evaluated in the test experiments. For brevity, we used the following annotations: CONV-AMSUA-RTTOV, CONV-AMSUA-ML, and CONV to annotate the experiments E, F, and G (Table 4), respectively. Fig. 3a illustrates the histogram of the observations minus the model background (OMB) from the CONV-AMSUA-ML. The histogram centered at ~0 K. The bias (average of OMB) was estimated to be only 0.002 K, which was the lowest absolute bias among all the channels. Apart from channels 6 and 7 from NOAA-18, all the other channels exhibited similar OMB distributions (figures are omitted) and yielded absolute biases of <0.1 K (Table 5). However, channels 6 and 7 from NOAA-18 experienced large biases (Fig. 3b) (channel 7 is not shown). The biases from channels 6 and 7 were 0.305 and 0.259 K, respectively (Table 5). In contrast, the same channels from CONV-AMSUA-RTTOV exhibited much lower biases. It was 0.0461 K for channel 6 (Fig. 3d). This finding suggests that the bias correction built into the ML-OO might not be effective for these two channels. The horizontal distribution of the OMB demonstrates that the values from METOP-B channel 6 were positive or negative in different regions (Fig. 4a), whereas most of the areas showed positive OMB values from NOAA-18 channel 6 (Fig. 4b). To identify the driver of this pattern, we analyzed the changes of the absolute biases (before the bias correction) from February 2015 to February 2016 from the CONV-AMSUA-RTTOV experiment. The changes were estimated to be 0.17 K and 0.15 K for NOAA-18 channels 6 and 7, respectively. The changes were less than 0.04 K in the other channels. Because the same RTTOV-OO was used for both periods, the changes thereby indicate that the characteristics of observations from NOAA-18 channels 6 and 7 changed significantly from February 2015 to February 2016. As the adaptive bias correction method was applied in CONV-AMSUA-RTTOV, the bias from all channels could be corrected (Fig. 3c, d). However, the training of the ML-OO was based on data from February 2015 and could not treat the bias well in February 2016 with a significant change in satellite characteristics. Previous studies have already shown that the characteristics of satellites can be changed during their operation. For instance, Zou and Wang (2011) have identified bias drifts for some channels of AMSU-A during certain periods. If the change is significant, as shown in our study, the current ML-OO method cannot handle the bias well. Online training by updating pre-trained networks using the latest satellite observations can be useful for correcting new biases. However, the frequency to update the ML-OO should be evaluated to balance accuracy and computational cost.

The root mean square difference (RMSD) and bias of temperature and zonal wind from the three experiments were evaluated using the European Centre for Medium-Range Weather Forecasts (ECMWF) reanalysis data (ERA-interim). At 500 hPa, the temperature RMSDs from CONV-AMSUA-ML were generally higher than that in CONV-AMSUA-RTTOV, but lower than that in CONV (Fig. 5a). This indicates that although the performance of ML-OO was slightly worse than that of RTTOV-OO at this level, the assimilation of additional AMSU-A BT by ML-OO improved the forecast, compared with the assimilation of only the conventional observations. All three experiments exhibited similar trends in the RMSD and bias evolutions (Fig. 5a, b). The ensemble spreads of temperature from CONV-AMSUA-ML and CONV-AMSUA-RTTOV were lower than that in CONV because they assimilate more data (Fig. 5c). For zonal winds, the RMSD in CONV-AMSUA-ML was also generally higher than that in CONV-AMSUA-RTTOV but lower than that in CONV (Fig. 5d). Furthermore, the bias in CONV-AMSUA-ML was similar to that in CONV-AMSUA-RTTOV (Fig. 5e).

The vertical profiles of the global average RMSD and bias for the temperature and zonal wind were further evaluated (Fig. 6). Like in the analyses of the time series above, the RMSDs of temperature and zonal winds from CONV-AMSUA-ML were generally higher than those in CONV-AMSUA-RTTOV but lower than those in CONV (Fig. 6a, c). Above 600 hPa, the reduction of RMSDs in CONV-AMSUA-ML relative to CONV was found to be larger. The T-tests were further conducted to determine the statistical significance of the differences between the RMSDs of the temperature and zonal wind from CONV-AMSUA-ML and CONV. As shown in Fig. 6b, for the p-value profiles above 600 hPa, the RMSDs of the temperature in CONV-AMSUA-ML were significantly different from those in CONV because p-values <0.05. Below 600 hPa, these differences were insignificant. As a result, the reduction in RMSDs by assimilating additional AMSU-A BT using ML-OO mainly reduced the RMSDs above 600 hPa. On the other hand, assimilating additional AMSU-A BT using RTTOV-OO had a greater reduction of RMSDs in a deeper layer (above 850 hPa were statistically significant) (Fig. 6a, b). A similar conclusion can be drawn for the zonal wind (Fig. 6c, d). We also found that the global average RMSD of temperature (zonal wind) in CONV-AMSUA-ML was 2% (3%) higher than that in CONV-AMSUA-RTTOV, but 4% (4%) lower than that in CONV. Fig. *7* shows a similar analysis, applied to the biases. As seen, the biases of temperature in CONV-AMSUA-ML were higher than those in CONV at most levels (Fig. *7*a). The p-value profile proved that these differences were statistically significant (Fig. *7*b). For zonal winds, the biases in CONV-AMSUA-ML were smaller than those in CONV and close to those in CONV-AMSUA-RTTOV at most levels below 450 hPa. As explained above, the biases of the radiance simulated by the ML model to the AMSU-A radiance were high for channels 6 and 7 from NOAA-18 satellite. Since the AMSU-A BT in channels 6,7 and 8 is sensitive to temperature in the mid-to-upper troposphere, the higher biases of BT at two channels might have exacerbated biases in the temperature profile. A higher temperature bias also exacerbates the temperature RMSD. This finding might be among the potential drivers, deteriorating the performance of ML-OO, compared to that of RTTOV-OO.

5. Discussion

The computational cost of training the ML-OO was high. The high training cost was driven by a random search for the best combination of hyperparameters for each channel and each satellite. In practice, it critically hinders the assimilation of numerous channels for other satellites. One can consider designing an NN to treat many channels simultaneously if sufficient collocated data from different channels are present. For instance, the same quality control is applied to many channels. Alternatively, some channels can use a pre-trained NN from other similar channels. Furthermore, the prediction time of the current ML-OO was within ~1 to 5 s range, therefore, slower than that of RTTOV, on ~1 s. The computational complexity of RTTOV is much lower than that of LBL RTM because the optical depth is calculated using a linear regression model with a small number of predictors, and because the radiative transfer equation (see Eq. 4 in Saunders et al. 1999) has only a few hundreds of multiplications and additions. For our ML-OO, the number of multiplications and additions were both ~331,800 (for 300 units with 4 hidden layers) in the forward propagation because it involved several matrix multiplications. Therefore, the computational time of the NN was slower than that of the RTTOV. There are several ways to accelerate the speed of the NN. For example, using a more efficient library to operate on the matrix or reducing the complexity of the NN while maintaining acceptable performance. For other applications, if the NN is not very complicated, its forecast could be faster than that of complicated physically based models. Due to this, some previous studies (Pal et al. 2019; Krasnopolsky et al. 2008) have explored the use of NN to replace the complicated parameterization schemes in NWP. In short, our method can be more advantageous in terms of execution time when other observation types are assimilated where complicated observation operators are used.

The proposed ML-OO does not provide tangent linear or adjoint operators, which are a core part of an observation operator package such as RTTOV, to support mainly variational data assimilation methods. However, it is relatively easy to derive the gradients of an NN because they are differentiable if a differentiable activation function is used (Scheck 2021). Besides, NN has been previously used to emulate the physical parameterization scheme. In this way, it provided its tangent and adjoint models with minimal effort for four-dimensional variational data assimilation (Hatfield et al. 2021).

Our study only elucidated the prospects of using ML-OO to assimilate the BT observations from channels 6 and 7 over the oceans and channel 8 on both land and oceans under clear-sky conditions, where the radiative transfer process was relatively linear. It is also beneficial to understand how to extend this method to assimilate BT for a wider variety of surface conditions and cloudy/rainy regions, where the radiative transfer process is more nonlinear. Moreover, BT from water vapor channels (such as from microwave humidity sounders) and infrared channels tend to be more nonlinear. Therefore, it would be also useful to assimilate the BT from these channels using the ML method.

From a technical standpoint, the NN was trained using Keras, TensorFlow, and Python language. The weights of the NN were saved to binary files, which were read by Fortran code in data assimilation. The prediction by NN during data assimilation was also written in Fortran. We suggest that a standard library can facilitate such integration. For instance, the Fortran–Keras Bridge (FKB) (Ott et al. 2020) can be tested for such purpose in future studies. However, if the NN structures become more complicated, the Fortran code implementation will be challenging. Thus, it might be useful to build standard libraries, thereby facilitating the use of NN for atmospheric research.

The information from the RTTOV-OO was implicitly used to obtain the ML-OO. The training data of the ML model were obtained from the data assimilation experiments, in which the radiance observations were assimilated using the RTTOV-OO. Therefore, the new ML-OO somewhat served as an emulator function for the physically based observation operator. This constraint limited the generalization of the proposed method in this study because some new observations may completely lack physically based observation operators. Ideally, ML-OO should be built without RTTOV or other physically based OO. To achieve this goal, we recommend the following procedure for future studies. First, one can run the data assimilation by assimilating only conventional observation data. Next, only the analysis data at locations that are close to the locations of the conventional observations are selected as the training data because they are expected to be more accurate. This method is, however, limited by the fact that the conventional observations may not have sufficient coverage in space and time. For instance, there are more conventional observations for land than for oceans. Future studies can check whether ML-OO based on such an inhomogeneous dataset will be generalized well or not. In the end, if the ML-OO can be built without a physically based model to assimilate new data, it can greatly extend our freedom to use various types of data, and also accelerate the development process to assimilate new data once a new observing platform is deployed.

6. Summary

In this study, we used machine learning as an observation operator to assimilate brightness temperature from AMSU-A channels 6 and 7 over the oceans and channel 8 over both land and oceans under clear-sky conditions. The ML-OO was built using forecasts from the NICAM-LETKF data assimilation system and the observed satellite radiance. First, we generated the data to train the ML model. We used the NICAM-LETKF system to perform 1-month data assimilation to assimilate the conventional observations and BT using RTTOV-OO. Furthermore, the model forecasts were interpolated from the model grids to the locations of the satellite observations and were combined with the satellite observations to train the DNNs. Second, we evaluated the performance of ML-OO by conducting three experiments under the same initial conditions in the same month of the following year. In the CONV-AMSUA-RTTOV experiment, the conventional observations and BT were assimilated using RTTOV-OO; in the CONV-AMSUA-ML experiment, the same observations were assimilated using ML-OO; in the CONV experiment, only the conventional observations were assimilated.

ERA-interim was utilized to analyze the RMSD and bias of the temperature and zonal wind from these experiments. We concluded that the CONV-AMSUA-ML result was slightly worse than that from CONV-AMSUA-RTTOV, but better than that from CONV. In numerical terms, the global-averaged RMSD of temperature (zonal wind) in CONV-AMSUA-ML was 2% (3%) higher than that in CONV-AMSUA-RTTOV but 4% (4%) lower than that in CONV. This finding indicates ML-OO was effective for the assimilation of BT although it was slightly worse than RTTOV-OO. Moreover, we did not discern any significant bias (< 0.1 K) in the simulated BT by ML-OO in most of the satellite channels without a separate bias correction procedure because the ML model considered bias during training. For two channels, we discerned significant biases (0.305 K and 0.259 K), which may have been associated with the significant changes in the satellite characteristics during the testing period.

Despite these promising results, some limitations of this study should be emphasized. Foremost, (1) the ML-OO could not handle the bias well if there were significant changes in the satellite characteristics. Moreover, (2) the ML-OO training in this study was expensive, which makes it impractical if BT from numerous satellite channels were assimilated. The performance of the ML-OO was (3) slightly worse than RTTOV-OO in terms of accuracy and speed, while only BT from limited channels under clear-sky conditions were assimilated. Lastly, (4) the RTTOV-OO was implicitly used to train the ML-OO. Future studies will try to alleviate these limitations to improve the proposed ML-OO.

Data availability

The conventional observations are obtained from the NCEP PREPBUFR data (https://rda.ucar.edu/datasets/ds337.0/). The AMSU-A radiance data can be obtained from https://rda.ucar.edu/datasets/ds735.0/. The ERA-interim reanalysis data are from https://www.ecmwf.int/en/forecasts/datasets/reanalysis-datasets/era-interim. The research data and code in this study are available from the corresponding author on request.

Acknowledgments

This work is supported by Japan Aerospace Exploration Agency (JAXA), JSPS KAKENHI (Grant number JP19H05605), the RIKEN Pioneering Project "Prediction for Science", JST AIP (Grant number PMJCR19U2), MEXT (Grant number JPMXP1020200305) as “Program for Promoting Researches on the Supercomputer Fugaku” (Large Ensemble Atmospheric and Environmental Prediction for Disaster Prevention and Mitigation), the COE research grant in computational science from Hyogo Prefecture and Kobe City through Foundation for Computational Science, JST SICORP (Grant Number JPMJSC1804), and JST CREST (Grant Number JPMJCR20F2). This research used computational resources of the supercomputer Fugaku provided by the RIKEN R-CCS (Project ID: ra000007) and the supercomputer Flow provided by the Information Technology Center at Nagoya University (Project ID:hp200128) through the High Performance Computing Infrastructure (HPCI) system supported by MEXT.

We are thankful to the members of the data assimilation team at RIKEN for their assistance and suggestions. Particularly, Hideyuki Sakamoto helped us to use the computer resources and Dr. Arata Amemiya gave useful suggestions. We thank Dr. David John Gagne for providing Python code examples at the AI4ESS Summer School Hackathon 2020. We thank Dr. Vladimir M. Krasnopolsky and Dr. Alex Belochitski from NOAA for the suggestion of the neural network design. We appreciate two anonymous reviews for helping us improve

the quality of the paper.

Appendix A: Decompose mean square error (MSE)

X is a random variable. The variance of X can be expressed as

$$\mathrm{Var}(X) = \mathrm{E}[X^2] - (E[X])^2 \qquad \text{(A1)}$$

Therefore,

$$\mathrm{E}[X^2] = \mathrm{Var}(X) + (E[X])^2 \qquad \text{(A2)}$$

Replacing X in equation (A2) by $y - h'(x)$, where x is the input variable and y is the data which the function $h'(x)$ wants to fit, both x and y are random variables.

$$E([y - h'(x)]^2) = Var[y - h'(x)] + (E[y - h'(x)])^2$$

$$= Var(y - h'(x)) + Bias^2 \qquad \text{(A3)}$$

Reference

Alvarado, M. J., V. H. Payne, E. J. Mlawer, G. Uymin, M. W. Shephard, K. E. Cady-Pereira, J. S. Delamere, and J.-L. Moncet, 2013: Performance of the Line-By-Line Radiative Transfer Model (LBLRTM) for temperature, water vapor, and trace gas retrievals: recent updates evaluated with IASI case studies. *Atmos. Chem. Phys.*, **13**, 6687–6711, https://doi.org/10.5194/acp-13-6687-2013.

Bengio, Y., 2012: Practical recommendations for gradient-based training of deep architectures. *arXiv:1206.5533 [cs]*,.

Bormann, N., A. Fouilloux, and W. Bell, 2012: Evaluation and assimilation of ATMS data in the ECMWF system. 15, https://doi.org/10.21957/w8ytxrcq5.

Boukabara, S.-A., and Coauthors, 2021: Outlook for Exploiting Artificial Intelligence in the

Earth and Environmental Sciences. *Bulletin of the American Meteorological Society*, **102**, E1016–E1032, https://doi.org/10.1175/BAMS-D-20-0031.1.

Chantry, M., S. Hatfield, P. Dueben, I. Polichtchouk, and T. Palmer, 2021: Machine Learning Emulation of Gravity Wave Drag in Numerical Weather Forecasting. *Journal of Advances in Modeling Earth Systems*, **13**, e2021MS002477, https://doi.org/10.1029/2021MS002477.

Dee, D. P., 2004: Variational bias correction of radiance data in the ECMWF system. *Proceedings of the ECMWF workshop on assimilation …*, 97–112.

Derber, J. C., 1998: *The Use of TOVS Cloud-Cleared Radiances in the NCEP SSI Analysis System*. American Meteorological Society,.

Eyre, J. R., S. J. English, and M. Forsythe, 2020: Assimilation of satellite data in numerical weather prediction. Part I: The early years. *Quarterly Journal of the Royal Meteorological Society*, **146**, 49–68, https://doi.org/10.1002/QJ.3654.

Glorot, X., A. Bordes, and Y. Bengio, 2011: Deep Sparse Rectifier Neural Networks. Vol. 15 of, 14th International Conference on Artificial Intelligence and Statistics (AISTAT, Fort Lauderdale, FL, USA., 9 http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf.

Grody, N., J. Zhao, R. Ferraro, F. Weng, and R. Boers, 2001: Determination of precipitable water and cloud liquid water over oceans from the NOAA 15 advanced microwave sounding unit. *Journal of Geophysical Research: Atmospheres*, **106**, 2943–2953, https://doi.org/10.1029/2000JD900616.

Harris, B. A., and G. Kelly, 2001: A satellite radiance-bias correction scheme for data assimilation. *Quarterly Journal of the Royal Meteorological Society*, **127**, 1453–1468, https://doi.org/10.1002/qj.49712757418.

Hatfield, S., M. Chantry, P. Dueben, P. Lopez, A. Geer, and T. Palmer, 2021: Building Tangent-Linear and Adjoint Models for Data Assimilation With Neural Networks. *Journal of Advances in Modeling Earth Systems*, **13**, e2021MS002521, https://doi.org/10.1029/2021MS002521.

Hocking, J., 2019: RTTOV v12 Quick Start Guide.

Hunt, B. R., E. J. Kostelich, and I. Szunyogh, 2007: Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenomena*, **230**, 112–126, https://doi.org/10.1016/j.physd.2006.11.008.

Janjić, T., and Coauthors, 2018: On the representation error in data assimilation. *Quarterly*

Journal of the Royal Meteorological Society, **144**, 1257–1278, https://doi.org/10.1002/qj.3130.

Kalnay, E., 2002: *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press.

Kingma, D. P., and J. Ba, 2014: Adam: A Method for Stochastic Optimization.

Kotsuki, S., Y. Ota, and T. Miyoshi, 2017: Adaptive covariance relaxation methods for ensemble data assimilation: experiments in the real atmosphere. *Quarterly Journal of the Royal Meteorological Society*, **143**, 2001–2015, https://doi.org/10.1002/qj.3060.

Krasnopolsky, V. M., M. S. Fox-Rabinovitz, and A. A. Belochitski, 2008: Decadal Climate Simulations Using Accurate and Fast Neural Network Emulation of Full, Longwave and Shortwave, Radiation. *Monthly Weather Review*, **136**, 3683–3695, https://doi.org/10.1175/2008MWR2385.1.

Kwon, Y., B. A. Forman, J. A. Ahmad, S. V. Kumar, and Y. Yoon, 2019: Exploring the Utility of Machine Learning-Based Passive Microwave Brightness Temperature Data Assimilation over Terrestrial Snow in High Mountain Asia. *Remote Sensing*, **11**, 2265, https://doi.org/10.3390/rs11192265.

Miyoshi, T., Y. Sato, and T. Kadowaki, 2010: Ensemble Kalman filter and 4D-Var intercomparison with the Japanese operational global analysis and prediction system. *Monthly Weather Review*, **138**, 2846–2866, https://doi.org/10.1175/2010MWR3209.1.

Okamoto, K., M. Kazumori, and H. OWADA, 2005: The Assimilation of ATOVS Radiances in the JMA Global Analysis System. *Journal of the Meteorological Society of Japan. Ser. II*, **83**, 201–217, https://doi.org/10.2151/jmsj.83.201.

O'Malley, T., E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi, and others, 2019: KerasTuner.

Ott, J., M. Pritchard, N. Best, E. Linstead, M. Curcic, and P. Baldi, 2020: A Fortran-Keras Deep Learning Bridge for Scientific Computing. *arXiv:2004.10652 [cs]*,.

Pal, A., S. Mahajan, and M. R. Norman, 2019: Using Deep Neural Networks as Cost-Effective Surrogate Models for Super-Parameterized E3SM Radiative Transfer. *Geophysical Research Letters*, **46**, 6069–6079, https://doi.org/10.1029/2018GL081646.

Rivera, J. P., J. Verrelst, J. Gómez-Dans, J. Muñoz-Marí, J. Moreno, and G. Camps-Valls, 2015: An Emulator Toolbox to Approximate Radiative Transfer Models with Statistical

Learning. *Remote Sensing*, **7**, 9347–9370, https://doi.org/10.3390/rs70709347.

Rodríguez-Fernández, N., P. de Rosnay, C. Albergel, P. Richaume, F. Aires, C. Prigent, and Y. Kerr, 2019: SMOS Neural Network Soil Moisture Data Assimilation in a Land Surface Model and Atmospheric Impact. *Remote Sensing*, **11**, 1334, https://doi.org/10.3390/rs11111334.

Satoh, M., and Coauthors, 2014: The Non-hydrostatic Icosahedral Atmospheric Model: description and development. *Progress in Earth and Planetary Science*, **1**, 18, https://doi.org/10.1186/s40645-014-0018-1.

Saunders, R., M. Matricardi, and P. Brunel, 1999: An improved fast radiative transfer model for assimilation of satellite radiance observations. *Quarterly Journal of the Royal Meteorological Society*, **125**, 1407–1425, https://doi.org/10.1002/qj.1999.49712555615.

——, and Coauthors, 2018: An update on the RTTOV fast radiative transfer model (currently at version 12). *Geoscientific Model Development*, **11**, 2717–2737, https://doi.org/10.5194/gmd-11-2717-2018.

Scheck, L., 2021: A neural network based forward operator for visible satellite images and its adjoint. *Journal of Quantitative Spectroscopy and Radiative Transfer*, **274**, 107841, https://doi.org/10.1016/j.jqsrt.2021.107841.

Terasaki, K., and T. Miyoshi, 2017: Assimilating AMSU-A Radiances with the NICAM-LETKF. *Journal of the Meteorological Society of Japan*, **95**, 433–446, https://doi.org/10.2151/jmsj.2017-028.

Whitaker, J. S., and T. M. Hamill, 2012: Evaluating Methods to Account for System Errors in Ensemble Data Assimilation. *Monthly Weather Review*, **140**, 3078–3089, https://doi.org/10.1175/MWR-D-11-00276.1.

Zhou, Y., and C. Grassotti, 2020: Development of a Machine Learning-Based Radiometric Bias Correction for NOAA's Microwave Integrated Retrieval System (MiRS). *Remote Sensing*, **12**, https://doi.org/10.3390/rs12193160.

Zou, C.-Z., and W. Wang, 2011: Intersatellite calibration of AMSU-A observations for weather and climate applications: AMSU-A INTERSATELLITE CALIBRATION. *J. Geophys. Res.*, **116**, https://doi.org/10.1029/2011JD016205.

List of Figures

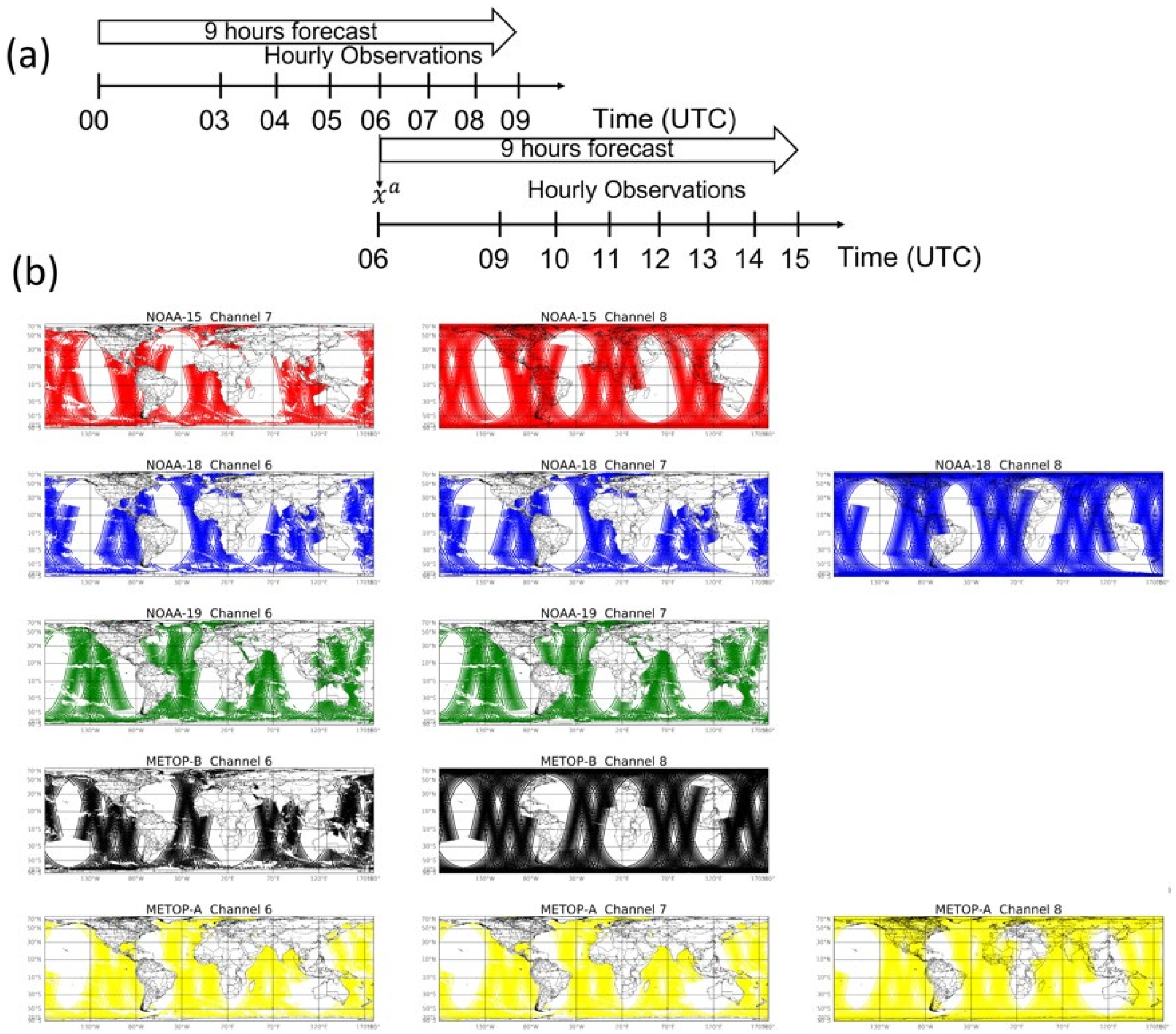

Fig. 1 (a) Schematic of 4D-LETKF, adapted from Fig. 3 of Terasaki and Miyoshi (2017). (b) Locations of AMSU-A BT data from different channels and satellites on 1st February

2015 after applying the quality control. Data thinning is not applied.

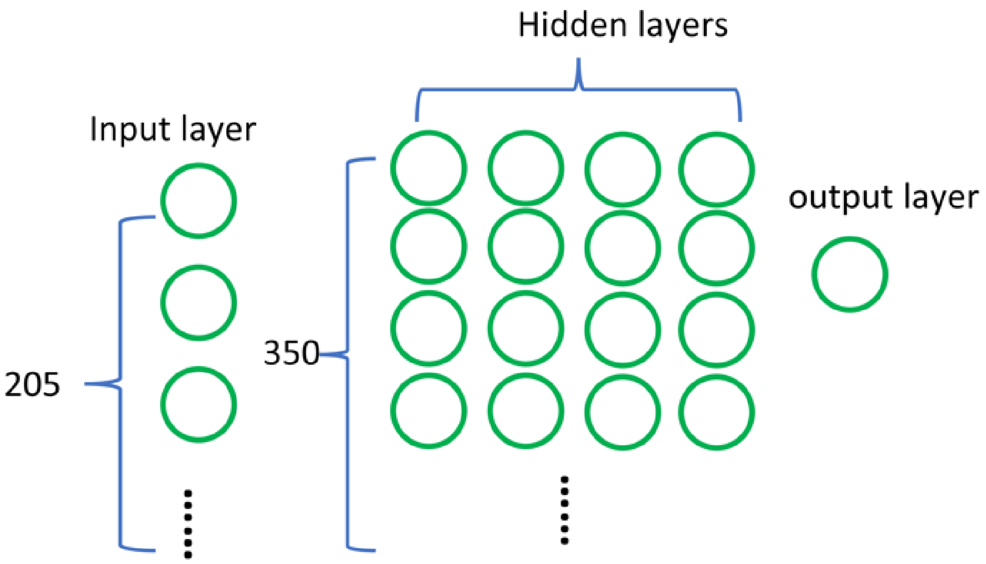

Fig. 2 The architecture of one of the DNNs. The number of units in the input layer is 205. There are 4 hidden layers and each of the layers contains 350 neurons. The number of units in the output layer is 1. Table 3 summarizes the number of hidden layers and the number of units per layer for all DNNs.

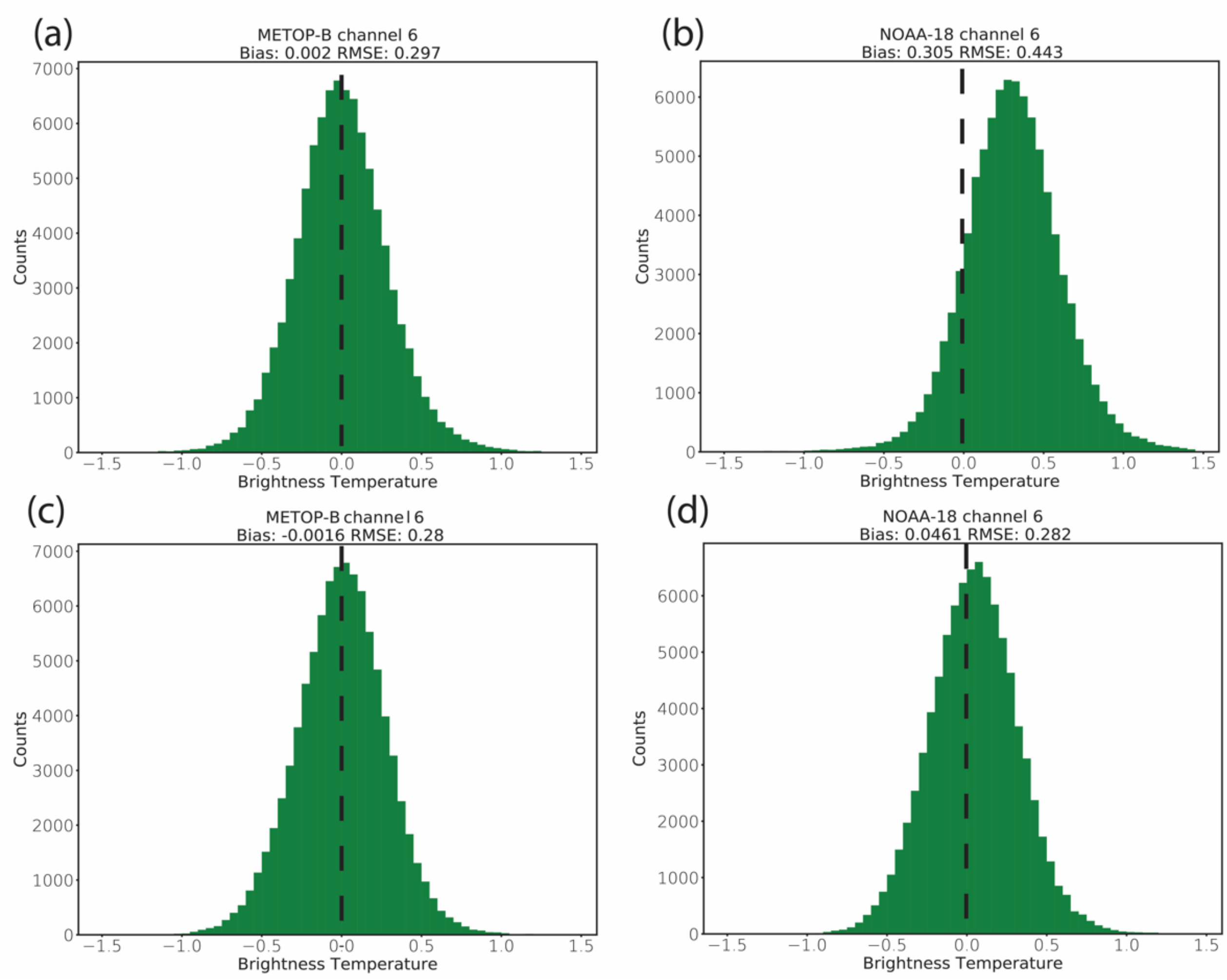

Fig. 3 Histograms of the observations minus model background (OMB) from experiment CONV-AMSUA-ML in February 2016 for (a) METOP-B channel 6 and (b) NOAA-18 channel 6, and experiment CONV-AMSUA-RTTOV for (c) METOP-B channel 6 and (d) NOAA-18 channel 6.

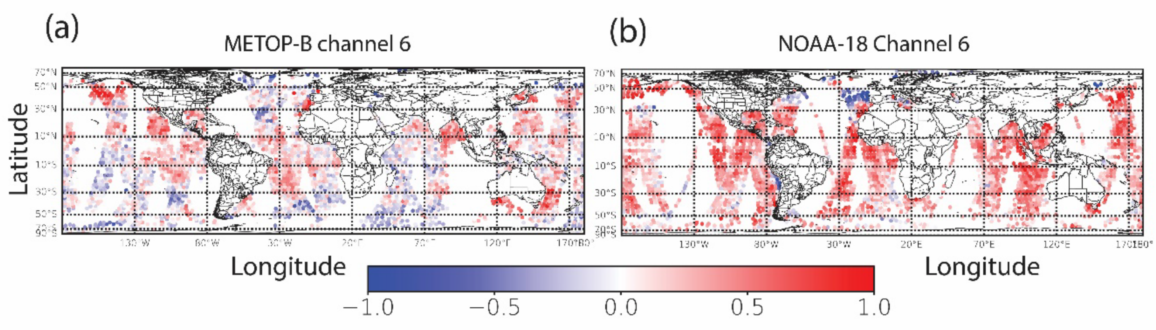

Fig. 4 Horizontal distributions of observation minus model background (OMB) for AMSU-A brightness temperature (K) from experiment CONV-AMSUA-ML from 2100 UTC 31 January to 0000 UTC 02 February 2016 at (a) METOP-B channel 6 and (b) NOAA-18 channel 6.

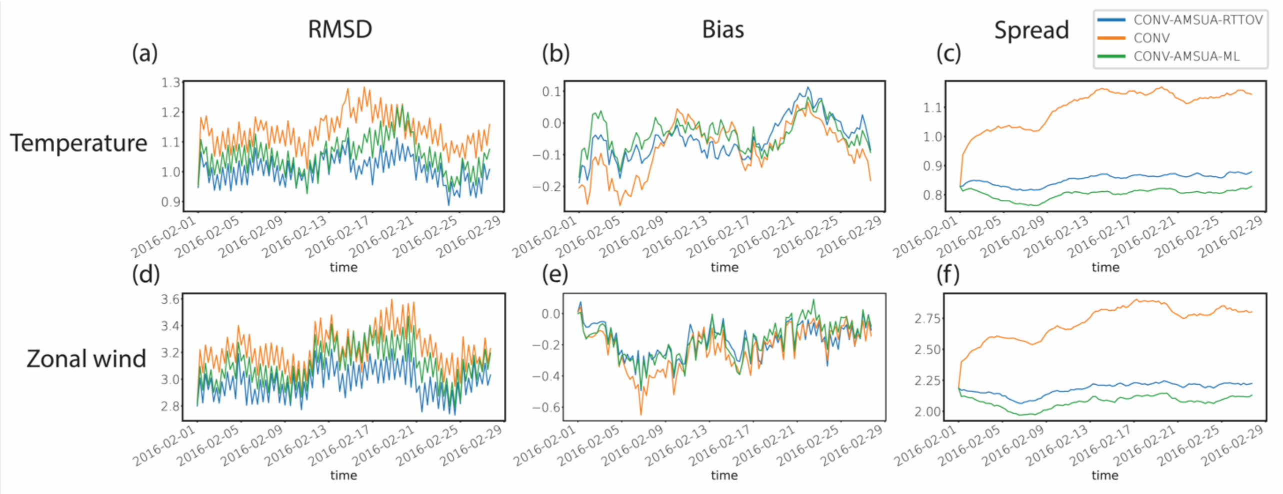

Fig. 5 RMSDs between the analysis and the ERA-interim reanalysis for (a) temperature (K) and (d) zonal wind (m s^{-1}) from three experiments in February 2016. The details of the experiments can be found in Table 4. The biases between the analysis and the ERA-interim reanalysis for (b) temperature and (e) zonal wind. The ensemble spreads for (c) temperature and (f) zonal wind. All plots are for 500 hPa.

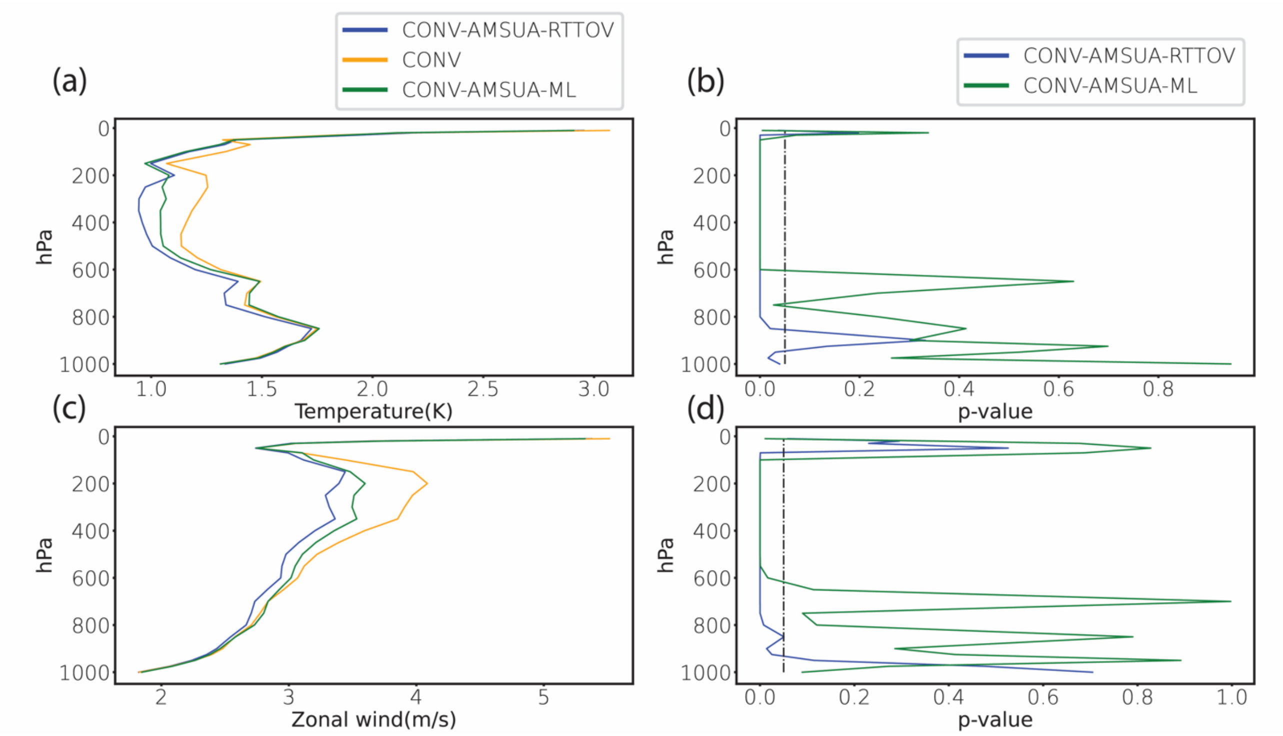

Fig. 6 RMSDs between the analysis and the ERA-interim reanalysis for (a) temperature (K) and (c) zonal wind (m s^{-1}) from three experiments in February 2016. The p-values from two T-tests between CONV-AMSUA-RTTOV and CONV, and between CONV-AMSUA-ML and CONV are shown in (b) for temperature and (d) for zonal wind. The vertical line indicates the p-value threshold of 0.05.

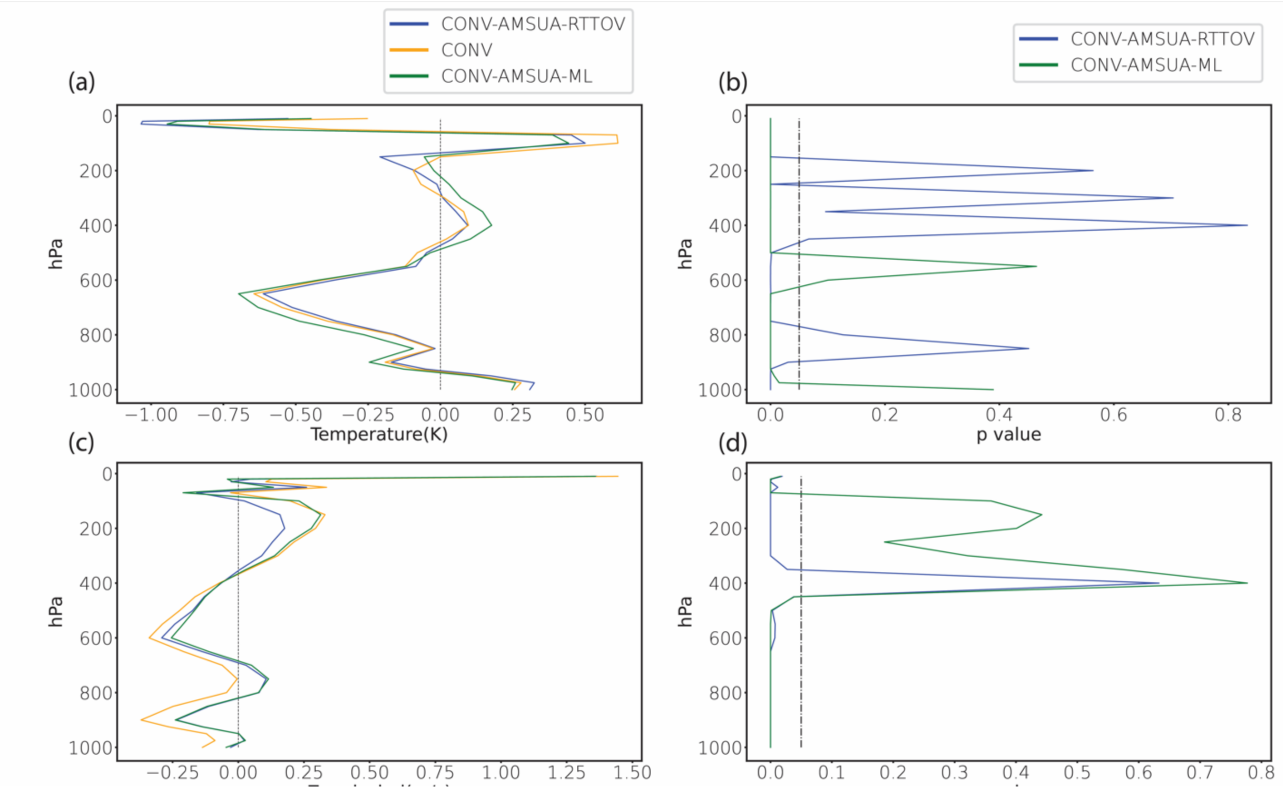

Fig. 7 Similar to Fig. 6 but for the biases.

List of Tables

Table 1 Standard deviation of the observation error (K) in the selected channels of AMSU-A from different satellites. The empty cells imply that the corresponding channels are not assimilated.

Channel	NOAA-15	NOAA-18	NOAA-19	METOP-A	METOP-B
6		0.3	0.3	0.3	0.3
7	0.3	0.3	0.3	0.3	
8	0.3	0.3		0.3	0.3

Table 2 Input features of the ML models.

feature order	1-78	79-156	157-196	197	198	199
variables	pressure (log of Pa)	temperature (K)	specific humidity (kg kg^{-1})	2-m surface pressure (log of hPa)	Surface (skin) temperature (K)	10-m zonal wind (m s^{-1})
features order	200	201	202	203	204	205
variable	10-m meridional wind (m s^{-1})	2-m temperature (K)	2-m specific humidity (kg kg^{-1})	inverse of the cosine function of satellite zenith angle	scan angle (degree)	latitude(degree)

Table 3 Statistics from the comparison between the simulated brightness temperature (K) by the neural network models (linear regression models) and the observed brightness temperature using the test data during the training. The metrics are the root mean square error (RMSE), the bias (observation - prediction), and the coefficient of determination (R^2). The numbers of hidden layers and the number of units per hidden layer of the DNNs are the results of the hyperparameter tuning.

		Deep Neural Networks					Linear Regression		
Satellites	**channels**	**RMSE**	**Bias**	R^2	**Hidden layers**	**units**	**RMSE**	**Bias**	R^2
NOAA-15	**7**	0.181	0.005	0.998	4	300	1.249	-0.001	0.903
	8	0.239	-0.002	0.999	4	350	1.182	0.002	0.971
NOAA-18	**6**	0.204	0.002	0.998	3	350	1.187	0.003	0.949
	7	0.227	-0.007	0.997	3	350	1.318	-0.001	0.910
	8	0.288	0.007	0.998	3	350	1.182	0.001	0.971
NOAA-19	**6**	0.182	-0.004	0.999	3	250	1.191	0.002	0.947
	7	0.366	-0.009	0.993	4	300	1.316	0.003	0.908
METOP-B	**6**	0.199	0.007	0.999	4	350	1.150	0.002	0.951
METOP-A	**6**	0.180	0.002	0.999	4	350	1.142	0.000	0.952
	7	0.183	-0.015	0.998	4	350	1.266	-0.002	0.915
	8	0.251	0.019	0.999	4	300	1.134	0.000	0.973

Table 4 Experiments for training and testing the ML models. The experiment CONV means only assimilating conventional observations. CONV-AMSUA-RTTOV indicates assimilating conventional observations and AMSU-A BT using the RTTOV as the observation operator together with an online bias correction method, while CONV-AMSUA-ML reflects assimilating the same observations using ML as the observation operator. Letters A to G indicate the corresponding experiments described in the main text.

	Training experiments		**test experiments**	
	Jan. 2015 (DA spin-up)	Feb. 2015 (DA cycle)	Jan. 2016 (DA spin-up)	Feb. 2016 (DA cycle)
CONV-AMSUA-RTTOV	A	B	D	E
CONV-AMSUA-ML		build the ML models		F

		(C)		
CONV				G

Table 5 Bias (the average of the observation minus the first guess) of the brightness temperature (K) from test experiments CONV-AMSUA-ML and CONV-AMSUA-RTTOV in February 2016. *NOAA-18 channels 6 and 7 have larger biases in experiment CONV-AMSUA-ML.

	CONV-AMSUA-ML			CONV-AMSUA-RTTOV		
	channel 6	channel 7	channel 8	channel 6	channel 7	channel 8
NOAA-15		-0.010	-0.029		0.0002	0.043
NOAA-18	0.305*	0.259*	0.021	0.046	0.019	-0.058
NOAA-19	-0.026	-0.085		-0.048	-0.055	
METOP-B	0.002			-0.002		
METOP-A	0.026	-0.011	-0.003	0.003	-0.017	-0.052