JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2022MS003309

Correcting Systematic and State-Dependent Errors in the NOAA FV3-GFS Using Neural Networks

Tse-Chun Chen[1,2], Stephen G. Penny[1,3], Jeffrey S. Whitaker[2], Sergey Frolov[2], Robert Pincus[4], and Stefan Tulich[1,2]

[1]Cooperative Institute for Research in Environmental Sciences (CIRES), University of Colorado Boulder, Boulder, CO, USA, [2]National Oceanic and Atmospheric Administration (NOAA), Physical Sciences Laboratory (PSL), Boulder, CO, USA, [3]Sofar Ocean Technologies, San Francisco, CA, USA, [4]Lamont Doherty Earth Observatory, Columbia University, New York, NY, USA

Key Points:

- A neural network (NN) trained to infer analysis increments from model forecasts learns to correct systematic errors in the FV3-GFS model
- Sensitivity analysis of the NN reveals physically consistent error characteristics that may be used to improve the NN architecture
- Applying online corrections from NN improves the accuracy of sequential data assimilation and extended free forecasts

Correspondence to:

T.-C. Chen,
tse-chun.chen@noaa.gov

Citation:

Chen, T.-C., Penny, S. G., Whitaker, J. S., Frolov, S., Pincus, R., & Tulich, S. (2022). Correcting systematic and state-dependent errors in the NOAA FV3-GFS using neural networks. *Journal of Advances in Modeling Earth Systems*, *14*, e2022MS003309. https://doi.org/10.1029/2022MS003309

Received 19 JUL 2022
Accepted 16 OCT 2022

Abstract Weather forecasts made with imperfect models contain state-dependent errors. Data assimilation (DA) partially corrects these errors with new information from observations. As such, the corrections, or "analysis increments," produced by the DA process embed information about model errors. An attempt is made here to extract that information to improve numerical weather prediction. Neural networks (NNs) are trained to predict corrections to the systematic error in the National Oceanic and Atmospheric Administration's FV3-GFS model based on a large set of analysis increments. A simple NN focusing on an atmospheric column significantly improves the estimated model error correction relative to a linear baseline. Leveraging large-scale horizontal flow conditions using a convolutional NN, when compared to the simple column-oriented NN, does not improve skill in correcting model error. The sensitivity of model error correction to forecast inputs is highly localized by vertical level and by meteorological variable, and the error characteristics vary across vertical levels. Once trained, the NNs are used to apply an online correction to the forecast during model integration. Improvements are evaluated both within a cycled DA system and across a collection of 10-day forecasts. It is found that applying state-dependent NN-predicted corrections to the model forecast improves the overall quality of DA and improves the 10-day forecast skill at all lead times.

Plain Language Summary Computer models used for operational weather prediction are not perfect—they are naturally only simplifications of the true atmosphere. Such imperfections result in reduced forecast quality. Weather forecast systems routinely correct the forecasts by pulling them closer to observations, thus providing some information about the errors present in the forecast model. Here, a neural network (NN) is trained to correct National Oceanic and Atmospheric Administration's operational weather forecast model, FV3-GFS, by "learning" the relation between the forecasts and the estimated model errors. The learned NN correction is then fed back into the weather model to improve the quality of the best guess state of the atmosphere and the subsequent 10-day forecasts. By analyzing how the NN output depends on its input forecast, we gain some insight about the model errors, which may be helpful for future atmospheric model development and improvements to future error-correcting NNs.

1. Introduction

Operational numerical weather prediction (NWP) models are inherently imperfect. Systematic errors result from approximations in deriving the governing equations, from their numerical implementation, and from conceptual and numerical errors in the parameterizations that represent subgrid scale physical and dynamical processes. Even small errors in any component of the NWP model can compound over time to produce errors that significantly degrade the forecasting skill.

Systematic errors can be addressed with a wide range of approaches. One approach is to improve the model components—the dynamical core and subgrid scale physics parameterizations. The forecast system as a whole can be improved, say by adopting stochastic parameterizations that account for uncertainty, or by increasing spatial resolution. Model forecasts can also be further improved by an "offline" post-processing using statistical methods (e.g., Model Output Statistics) or machine learning (ML) methods applied to the model output after the completion of model forecast. However, the model errors may be convoluted over time and become more nonlinear as forecast progresses, leading to errors that are more difficult to represent.

To avoid such a complication, there is increasing interest in applying ML methods for "online" correction of the model forecast within the operational forecast-analysis cycle itself. Here an online correction, as opposed to an offline correction, is referred to as the methods that are integrated into the model forecast cycle such that the subsequent cycles benefit from previous improvements. The attraction of online correction is that, by reducing systematic errors, corrections can improve the forecasts (background state) provided to the data assimilation (DA) analysis algorithm, allowing the full-cycled DA system to make better use of the observations. As one example, Crawford et al. (2020) improved the 10-day forecast skill of the US Navy's NAVGEM model by applying a seasonal moving average of the analysis increment in their 1-year training data as a correction. The correction is fixed throughout different forecast lead-times and is independent of the meteorological conditions of the day. Fixed corrections limit the generalization of the method, as the correction may become invalid for longer forecast lead times or when applied during a year that has a different climate background environment due to interannual or decadal variability (e.g., ENSO). The storage required to maintain at least a full year of the seasonal moving averaged analysis increment data for the full 3D atmosphere is also a burden.

Bonavita and Laloyaux (2020), hereafter BL20, addressed some of these limitations by training a neural network (NN) to predict the analysis increments from the corresponding forecasts. Corrections were computed at low spatial resolution (smoothing to T21 by truncating higher wave number in spectral space) to accelerate training, and a column-based NN predicted analysis increments within the atmospheric column given the corresponding forecast and climatological variables including the time of the day, the month of the year, and the geo-location of the column. The NN correction was applied in conjunction with weak constraint 4D variational DA (4D-Var), as well as extending the original stratosphere-only correction to the troposphere. The validation period of the online correction together with 4D-Var was short due to resource limitations. A question that remains is whether it is possible to apply the NN correction online for medium-range operational forecasts.

Watt-Meyer et al. (2021) built on earlier work (e.g., Brenowitz & Bretherton, 2018, 2019) that used ML to reproduce a high-resolution reference data set from a lower-resolution input data set. They trained a random forest to correct a coarse C48 (~200 km) resolution FV3-GFS model with 79 vertical levels. They generated the training data set by nudging the model toward the higher-resolution operational Global Forecasting System (GFS) analyses. The random forest was trained to predict the nudging tendencies of the prognostic variables of a column from the corresponding column states. The random forest correction improved both the 10-day weather prediction skill and the climatological variables (e.g., annually averaged precipitation) that were not directly updated by the correction. Recently, Bretherton et al. (2022) expanded the work on correcting the coarse C48 model by learning from a high-resolution reference simulation using a modified version of FV3-GFS with a 3 km grid. Both random forest and NN methods were examined in the study. This line of work focused on better representing the subgrid-scale processes of a coarse-resolution model, while we explore a similar approach in the context of operational NWP using a much higher resolution model.

Here, we apply ML methods to learn and correct systematic state-dependent model errors in National Oceanic and Atmospheric Administration's (NOAA's) FV3-GFS by comparison to an observationally informed atmospheric analysis. This work aims to correct model errors online while generating a forecast and improve common weather prediction tasks. Corrections to model error are determined from increments generated by "replaying" (see Section 2.2) NOAA's FV3-GFS model to ECMWF IFS analysis. We generate three progressively more complex predictors for the systematic error: (a) a linear baseline similar to Crawford et al. (2020), (b) a 1D atmospheric column-oriented ML predictor similar to BL20, and (c) an extension of the 1D ML predictor of the BL20 that also includes horizontal information using convolutional neural networks (CNNs). We conduct a comprehensive evaluation of the trained error predictors against each other using an offline set of analysis increments, in a cycling DA system, and in a set of 10-day forecasts.

2. Methods and Setup

We seek to learn state-dependent systematic error from analysis increments and apply corrections to improve the quality of the medium-range forecast and DA of the FV3-GFS using a resolution close to what is operationally used in the national weather service. To achieve this goal, we train two NN architectures to predict the analysis increments conditioned on the corresponding forecasts. The trained NNs are compared with several linear baselines in offline evaluation. Predicted corrections are then applied to forecasts in an online evaluation for both DA

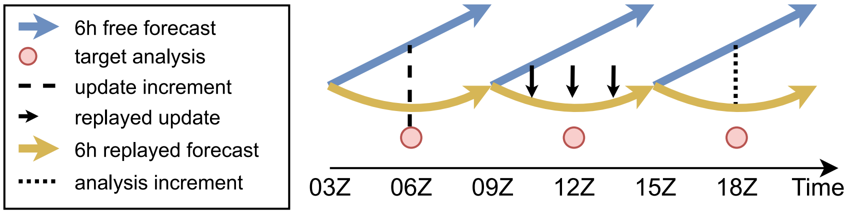

Figure 1. Schematic illustration of the replay system.

and medium-range forecasts, in which the performance metric is the forecast error reduction. Both offline and online evaluations are performed in an independent testing period that is not included in the training process.

2.1. Model

We use the NOAA operational NWP model (FV3-GFS; UFS Community, 2020), which is comprised of a finite volume cubed sphere dynamical core (FV3; see e.g., Lin, 2004; Putman & Lin, 2007) and the NOAA GFS physics. We use the FV3-GFS at a reduced C192 resolution (≈50 km), which is coarsened from the operational resolution of C768 (≈13 km). The atmospheric column is discretized into 127 vertical levels in FV3-GFS.

2.2. Data

To simulate the DA process with reduced computational cost, we use a "replay" system to constrain the forecast using an externally provided full-field analysis instead of directly assimilating observations. Figure 1 shows a schematic of the replay system. Given a 6h forecast as background (blue arrow), an "update increment" (dashed line) is computed by the difference between the background forecast and a target analysis (red dot) at the analysis valid time (e.g., 06Z, 12Z, 18Z in the schematic). A forcing to the tendency equations (black arrow) is then obtained by dividing the update increment by 6 hr to match the update frequency. We obtain the replayed trajectory (yellow arrow) by restarting the model from the same initial condition of the forecast segment (3 hr before the valid time of the target analysis, e.g., 03Z, 09Z, 15Z) with the additional forcing term. We further define the difference between the background and the replayed trajectory at the analysis valid time as the "analysis increment" (dotted line). This replay process is similar to the incremental analysis update (IAU; e.g., Bloom et al., 1996; Lei & Whitaker, 2016) method, which was developed to provide a better-balanced DA update by nudging forecasts over a fixed-size window (e.g., 6h). Bengtsson et al. (2019) showed that the replay methodology allows for rapid generation of training data sets that reveal the nature of the model error even if the model is replayed to an external analysis.

The target for the replay system can be supplied from a cycled DA system using the same model (i.e., a "self-analysis") or from an external source that uses a different model. The advantage of using the self-analysis is that it is available in real-time at the operational center, while the benefit of using the external analysis is that it may reduce correlations between the background and the analysis.

In this study, we use the operational IFS analysis from ECMWF, an external analysis, as the replay target. An earlier Cy41r2 version of the same model powered the latest European center reanalysis product (ERA5; Hersbach et al., 2020). We do not directly use the update increment to train the NNs because the resulting correction will likely replace the FV3-GFS error with the IFS error. Instead, we use the analysis increment (dotted line), the difference between the background forecast (yellow arrow) and the replayed trajectory (blue arrow) at the analysis valid time, as the training target. Because the update is applied through the forcing term, the replayed trajectory is not the same as directly replacing the states with the target analysis. This results in the differences between the update increment and the replayed analysis increment.

The replay and analysis increments are computed over a 15-month period from 20 November 2019 to 1 March 2021. The first 10 days are discarded as a spin-up period, and the following 12 months are used for training and validation, while the remaining 3-month period is reserved for independent testing. To capture the annual and seasonal cycles in both the training and validation process, we withhold the initial 15 days of each season (every 120-day period) of those 12 months for validation.

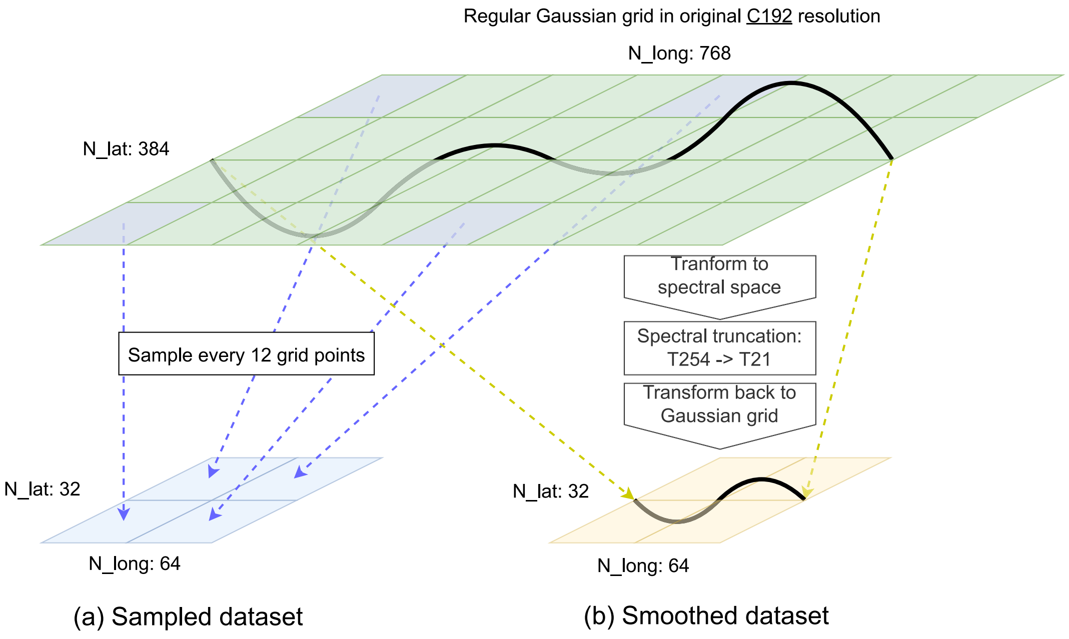
Regular Gaussian grid in original C192 resolution
N_long: 768
N_lat: 384
Tranform to spectral space
Spectral truncation: T254 -> T21
Transform back to Gaussian grid
Sample every 12 grid points
N_lat: 32
N_long: 64
N_lat: 32
N_long: 64
(a) Sampled dataset
(b) Smoothed dataset

Figure 2. Illustration of the two data reduction approaches from the original C192 resolution to an equivalent of T21: (a) sampling of every 12 grid points and (b) smoothing by spectral truncation. Note that both the reduced data sets are of the same size in the regular Gaussian grid space.

To reduce the computational cost of the NN training, a data reduction is applied to the Gaussian grid of size 768 × 384 corresponding to the original C192 resolution by either sampling grid points or by applying a smoothing of the global data fields to a 64 × 32 Gaussian grid corresponding to T21 as illustrated in Figure 2. For the former approach, we sample from the original Gaussian grid every 12 grid points. This approach preserves finer details from the original resolution to some extent. Alternatively, the smoothing approach (spectral truncation) converts the data from the original Gaussian grid to spectral space, truncates the higher wave numbers to the T21 resolution, and then converts it back to the corresponding Gaussian grid. Such a truncation approach assumes that the more easily diagnosed model errors are larger in scale and thus removes information not represented in T21 resolution.

The learning tasks in our study are different from most ML applications: the signal-to-noise ratio is unusually low because the analysis increments contain not only the model error information but also the inhomogeneity and irregularity of the observation network distribution in space and time, initial condition error of the forecasts, observational errors, etc. (see an indication in Figure 7, showing model error correction as filtered analysis increment). Therefore the goal is not to learn everything in the analysis increments but to extract only the information that is dependent on the input features. From this perspective, the smoothing approach is intended to remove some of these sources of noise.

2.3. Error Correction Methods

We devise two column-based NN error correction methods, modified after the column approach of BL20. The first method, which we refer to as the column NN hereafter, is trained using the sampled data set as it does not require any neighboring information for input. The main difference of our column NN from the BL20 is the additional input of some ancillary information about physical processes such as radiative fluxes, land-sea-ice mask, etc. Further, the longitude, the time of the day, and the day of the year information are transformed into sine and cosine forms. An obvious drawback of the column NN is that it does not incorporate information about the horizontal structure of the background forecast as input to predict the analysis increment correction. This means that the column NN only sees the local input information and neglects the surrounding large-scale weather pattern (e.g., horizontal wind structures). To incorporate the spatial relationship in the error field, we also consider a CNN. CNN has had great success in computer vision applications; it scans through 2D fields with a moving

Table 1
Input Variables for Neural Network

ID	Variable name
0–126	Temperature
127	log(surface pressure)
128–254	u-component wind
255–381	v-component wind
382–508	Specific humidity
509	Aerodynamic conductance
510	Canopy water evaporation gfs lsm
511	Direct evaporation from bare soil gfs lsm
512	Sublimation (evaporation from snow) gfs lsm
513	Snow phase change heat flux gfs lsm
514	Snow cover gfs lsm
515	Surface storm water runoff gfs lsm
516	Transpiration gfs lsm
517	Surface temperature
518	Surface temperature over ice
519	2 m specific humidity
520	Averaged potential evaporation rate
521	Surface roughness
522	Averaged albedo
523	Clear sky downward long wave flux
524	Clear sky downward short wave flux
525	Clear sky upward long wave flux
526	Clear sky upward long wave flux at toa
527	Clear sky upward short wave flux
528	Clear sky upward short wave flux at toa
529	Land-sea-ice mask
530	Latitude
531	sin(longitude)
532	cos(longitude)
533	sin(hour of the day)
534	sin(day of the year)
535	cos(hour of the day)
536	cos(day of the year)

window (also known as a kernel) assuming an invariant input-output relationship across the field. Thus for comparison, we also adopt a convolutional architecture in the horizontal directions for the same column-base NN trained against the smoothed data set. We refer to this approach as a low-res CNN because the convolution architecture is trained to learn the large-scale spatial structure in the truncated resolution and can only operate in that same resolution. The low-res CNN mainly focuses on the errors in the large scales and includes the adjacent grid information when predicting the center grid column, using a kernel size larger than 1. The hidden layers and the output layer have the same horizontal domain size of 64 × 32 as the input.

Performance of the NN methods is assessed against three additional linear baseline methods similar to the method used by Crawford et al. (2020). We use the annual average, the seasonal (3-month) moving average, and the hourly seasonal moving average of the analysis increments. All three linear baselines are computed only from the training period for a fair comparison with the NN methods. The linear baseline methods represent tradeoffs. The hourly seasonal average baseline is algorithmically simpler than the NN methods. However, when implemented at the same resolution as the operational model, the volume required for storing a full year of global data for each variable can be prohibitive in an operational environment. The training of NNs can be viewed as a compression of this huge amount of data.

2.4. Training the NNs

2.4.1. Training Setup

The NNs are trained to predict separate corrections to each model state variable within a vertical column: temperature, specific humidity, and u- and v-wind, which are prognostic variables of the atmospheric model. The training target is the collection of analysis increments obtained from the replay data set. Additional inputs to the NN include ancillary information such as time of the day, latitude, longitude, land-sea mask, radiative fluxes, etc. (see Table 1 for a complete list of all input features). To improve the interpolation of the temporal and spatial information, the time of the day, the day of the year, and longitude information are transformed into sine and cosine forms. The input and output data are normalized using the mean and standard deviation calculated from the training data set. The stochastic gradient descent method is used to minimize a mean square error (MSE) loss function. The two NN methods share common hyperparameters (see Table 2 for the search space), which we optimize using the validation data set. To make the training more efficient and to prevent overfitting, we use an early stopping criterion that terminates the training if the validation score does not improve during the last 20 epochs. After training, we then perform independent testing on the NNs using both offline and online evaluation.

2.4.2. Offline Evaluation

The performance metric for offline evaluation is the explained percentage of the target analysis increment, a normalized MSE, defined as:

$$1 - \sum (y_{truth} - y_{pred})^2 / \sum y_{truth}^2, \tag{1}$$

where y_{truth} is the target analysis increment, and y_{pred} is the predicted correction from the error correction methods. Having an explained percentage of 100% represents a perfect prediction, and having 0% means the correction method neither improves nor degrades the forecast. Negative values indicate that the correction has degraded the forecast skill.

Table 2
Hyperparameter Search Space for Neural Network Training

	Column NN	Low-res CNN	
Data reduction	Sampling	Smoothing	
Kernel sizes	1	1	3, 5
Minibatch size	8	8	1
Dropout probability	0.25, 0.5, 0.75		
Learning rate	1e−5, 1e−4, 1e−3		
Weight decay	0.01, 0.05, 0.25		
Channel number/hidden neuron	2,048, 4,096, 8,192		
Number of layers	3, 4, 5		

Performance of the NN methods is assessed by comparison to the three linear baselines, which are also computed both from the sampled and smoothed data sets (the same data sets used for NN training) to ensure a fair comparison with the NN methods. All error correction methods are evaluated using both reduced data sets in the full 3 months of the independent testing period for offline testing. For this offline evaluation, we include also a close replica of the BL20 setup (see Section 2.3 for its main differences from the column NN).

We use the analysis increment in the testing period as "truth" for offline evaluation so that the NNs can be evaluated without being integrated with the FV3-GFS model. The performance metric is aggregated over the whole globe and the entire testing period. It should be emphasized that the column NN and the low-res CNN are trained with the sampled and smoothed data sets, respectively, and hence the truth for evaluating the performance of the NNs is specific to each data set. For this reason, separate baselines are created for each data set for a fair comparison, and thus we do not compare the column NN and low-res CNN directly in the offline evaluation.

2.4.3. Online Evaluation

For the online evaluation, we examine the forecast error changes resulting from the corrections predicted by the NNs. To achieve this, the error correction needs to be integrated with the model workflow. This integration would normally require interfacing between the FORTRAN-based FV3-GFS and the typically Python-based ML libraries (e.g., Ott et al., 2020). To circumvent this software engineering challenge and develop a prototype, we use temporary intermediate files to exchange data between the FV3-GFS model and the trained NN. Using the FV3-GFS utility for ingestion of DA update files in the Gaussian grid space, all error corrections are applied directly to the forecast fields.

As the analysis increment embeds the information of errors that accumulates over 6h interval, it is pragmatic to make this file-based update at the end of each 6h forecast segment using the NN predicted corrections. This approach is not ideal for an operational forecast, as it would require stopping the model integration and initializing the ML package and the NNs every 6 hr.

Only the hourly seasonal moving average baseline is included in the online evaluation. Here the linear baseline is computed from the data set in the original model resolution (not the reduced data set used for NN training). The linear baseline and the column NN are straightforward to integrate into the forecast workflow. Although the column NN is trained from the sampled data set, it can be applied directly to each column in the original C192 resolution since the data reduction simply extracts a subset from the original column data, and column NN does not require neighboring grid information. In contrast, additional spectral operations are required for using the low-res CNN for online correction in the original resolution (C192), because it is trained to operate at a lower resolution and depends on neighboring information. The learned spatial dependencies within the kernel are not applicable across different resolutions. Figure 3 illustrates the data processing pipeline for performing a low-res CNN correction online. Starting from the input, the background forecast is truncated to T21 spectral resolution. The resulting CNN-predicted corrections at T21 resolution are then upscaled (through zero padding of higher harmonics) to the original T192 truncation before ingesting them into the FV3-GFS forecast model.

To evaluate the online performance of the error correction methods, we examine two tasks essential for operational NWP: (a) sequential DA and (b) 10-day free forecasts. Their workflows are integrated with the error correction methods, as illustrated in Figure 4. We use 3D-Var as a relatively low-cost option for DA. The error correction is applied to the model forecast to correct the background fields before the assimilation of observations. Ideally, an improved background should also lead to improved analysis and subsequent forecasts. For the extended free forecast, we apply the error correction to a 6h forecast segment, from which we initiate the subsequent 6h segment until a full 10-day forecast is obtained. To examine the quality of the background produced by 3D-Var and also the 10-day forecasts, the ECMWF IFS analysis data are used as "verifying truth" to compute forecast errors. This is appropriate because the quality of the analysis produced by 3D-Var and the 10-day forecasts at reduced resolution are significantly lower than the operational IFS analysis. Due to resource limitations, the DA experiment spans

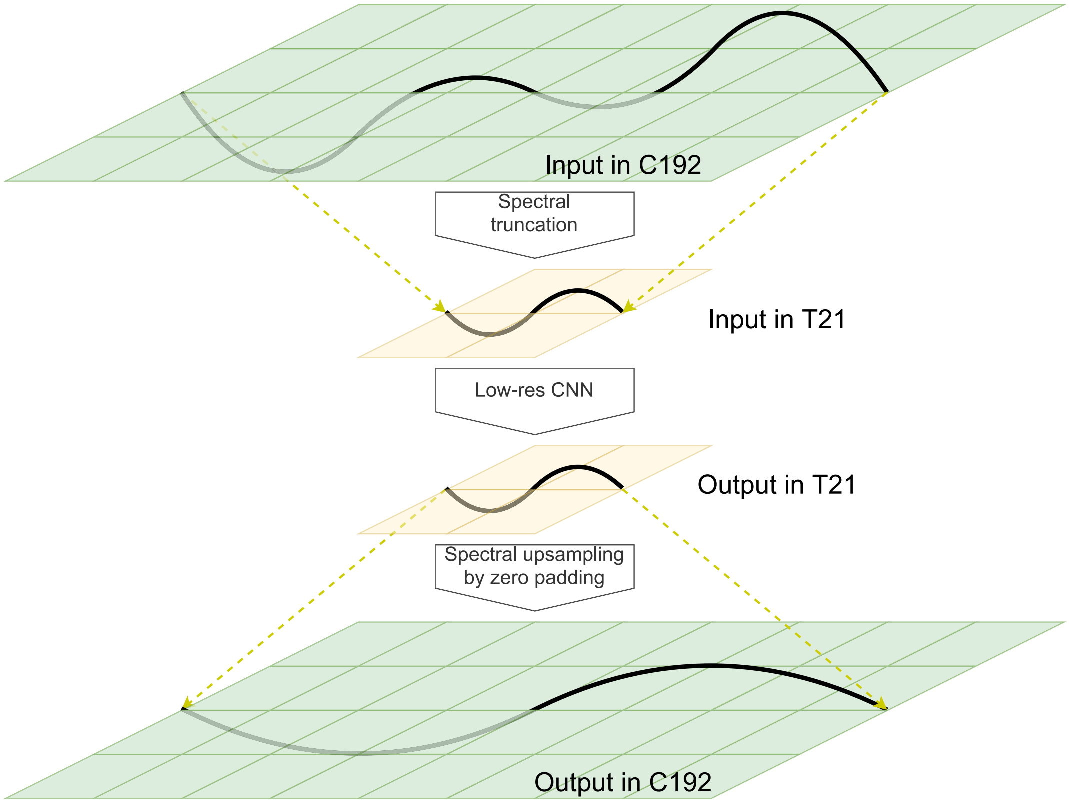

Figure 3. Data processing pipeline for the online low-res convolutional neural network (CNN) correction. Starting from the forecast input in the original C192 Gaussian grid space, the data are down-sampled by spectral truncation to T21 Gaussian grid space. The CNN takes in forecast fields in T21 and predicts the corresponding correction fields in T21. The predicted error correction in T21 is then up-sampled by zero padding in spectral space.

only the second month of the testing period (January 2021), and 10-day forecast experiments are run only once per day at 18Z of the same month (31 cases in total).

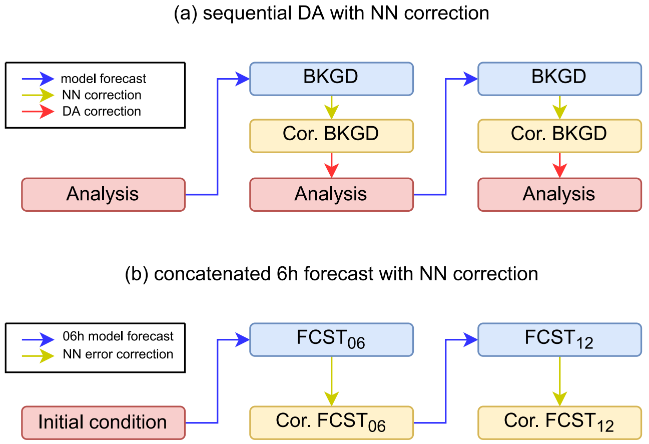

Figure 4. Schematic illustration of the integration of the error corrections with the workflow of (a) sequential data assimilation and (b) concatenated 6h free forecasts.

3. Results

3.1. Offline Performance

We first examine the offline performance of the linear baselines and the NN approaches in predicting the analysis increments in the testing period of the two reduced data sets. To understand how our NNs perform, we compare the skill of the annual average (blue), seasonal moving average (red), hourly seasonal moving average (yellow), a close replica of the setup of BL20 (green), and our two NN approaches (orange) in Figure 5.

All NN approaches substantially outperform the linear baselines for all variables in both reduced data sets. The hourly seasonal average is generally the best performing linear baseline method and will be examined in online correction experiments in a later section. The low-res CNN (Figure 5b) and the column NN (Figure 5a) slightly outperform our replica of BL20 in the smoothed and sampled data sets respectively. The corrections in temperature and specific humidity appear to be more predictable than that in the winds. Comparing the performance of the linear baselines and the NNs for each variable reveals the predictability originating from the average, seasonal cycle, diurnal cycle, and state-dependent components. For instance, the hourly seasonal baseline method (yellow) reveals the periodic model error components, while the NNs

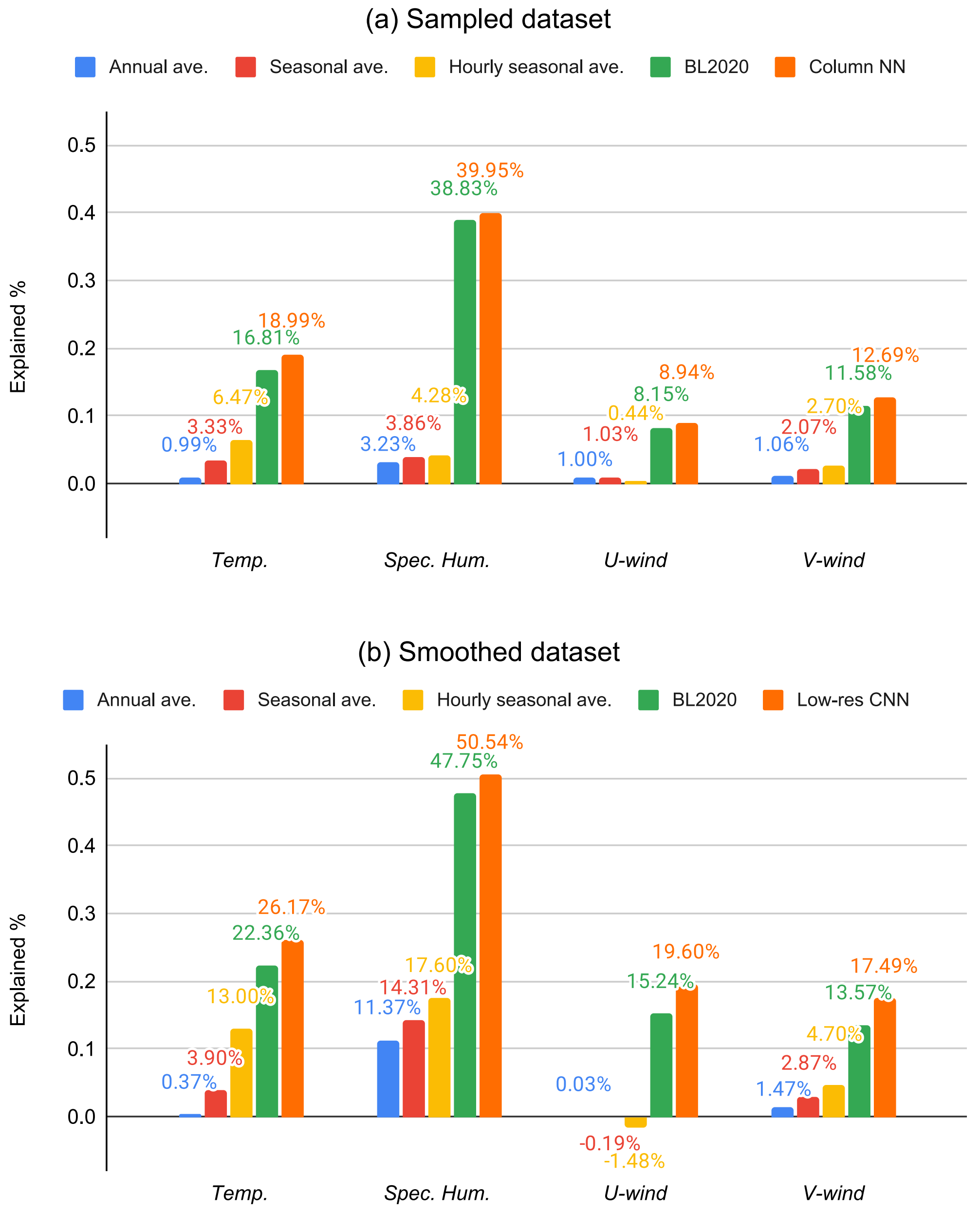

Figure 5. Offline performance of linear baselines (annual average, seasonal moving average, hourly seasonal moving average), a close replica of the BL20 approach, and the neural network implemented in this study in predicting the analysis increment of temperature, specific humidity, u- and v-component wind in the testing period of the (a) sampled data set and (b) smoothed data set. Performance metric is the global explained percentage formulated in Equation 1.

(green and orange), with both forecast and time information inputs, extract the state-dependent components in addition to the periodic components of the model errors. The difference between the performance of the two methods measures to some extent the predictability originating from the state-dependent error components learned by the NNs. For temperature, the annual average provides little skill for prediction, whereas the seasonal cycle and the diurnal cycle contribute some prediction skill, especially on large scales (shown in the smoothed data set). In this case, the prediction skill of the state-dependent component from the NNs provides an additional 10% of the explained percentage to the hourly seasonal average. On the other hand, the annual average of the specific humidity itself provides a significant portion of predictability in the linear methods, and the NNs add another 30% to the performance metric on top of the linear baseline. The linear components are not predictive for the winds (especially for the u-wind), and the state-dependent components in the winds yield also roughly 10%

additional skill, similar to that for the temperature. When comparing across the two reduced data sets, the skill for the smoothed data set is generally higher owing to the smoothing effect, indicating that the large-scale features are more predictable. For this different nature in the data sets, we do not make a direct comparison between the performance of low-res CNN and column NN in the offline evaluation.

3.2. Sensitivity Analysis

To understand the error characteristics captured by the NNs, we examine the averaged gradients of the column NN subject to all training samples. The gradient $\frac{\partial \hat{\mathbf{F}}}{\partial X_i}|_{X_i=x_{i,n}}$ is one of many methods (Mamalakis et al., 2022) that allows one to examine how the output of the learned function $\hat{\mathbf{F}}$ by NN depends on each of the input variables X_i at specific sample n. The gradients are further averaged over the training samples and visualized in Figure 6 with the vertical and horizontal axes representing the input and output respectively. Note that each column of blocks in the figure represents a NN trained separately for predicting different variables. When training the NNs, the input and output data are normalized using the mean and standard deviation calculated from the training data set, so the values are non-dimensional and the sensitivities are realized at the forecast mean for each level. For simplicity, we refer to the normalized inputs as forecast anomalies as they are deviations from the mean value of each level. Positive (negative) sensitivity values indicate that the NN adds corrections of the same (reverse) sign as the anomalous forecast.

Figure 6 (top) shows the sensitivity of the NN predicted corrections to the temperature, specific humidity, u- and v-wind forecast inputs. The highest sensitivity appears to be on the diagonal blocks, meaning that the corrections are most sensitive to the forecasts of the same variable. The diagonal pattern of negative values across all variables indicates that the column NN reduces local forecast anomaly, except for the block of u-wind, which appears to only have gradients at some of the top levels (e.g., above 10 hPa). The immediate parallels of the diagonal with positive values show that the forecast anomalies at levels right above and below increase anomalies at the levels in between (e.g., below 150 hPa around the diagonal line of the temperature diagonal block). Around the diagonal line, there are several parallels with alternating signs that fade away as the vertical distance from the diagonal line increases (e.g., around the diagonals of the diagonal blocks of temperature, specific humidity, and v-wind), indicating the vertically localized influences of the forecast input features. Notice that the widths of the diagonal parallels are thinner in the stratosphere than in the troposphere (below 150 hPa).

The off-diagonal blocks represent the cross-variable sensitivities. The sensitivity of the specific humidity correction to the temperature forecast input is the largest off-diagonal block, followed by the sensitivity of temperature corrections to the tropospheric forecasts of specific humidity, showing that the model errors of the two variables are closely related to each other. The diagonals of these two off-diagonal blocks are positive, meaning that the anomaly of one variable will increase the anomaly of the other. The wind forecasts also provide some information for predicting the temperature and specific humidity corrections, but not the other way around. The wind corrections do not depend on the forecast of other variables. We also note that the entire matrix of blocks is non-symmetric. For example, the prediction of humidity correction is more sensitive to the temperature forecast than the other way around.

On the right-lower (troposphere) quarter of the blocks corresponding to the prediction of temperature and specific humidity corrections, the horizontal patterns suggest a homogeneous response of a thick tropospheric layer to a single level of tropospheric forecasts. Except for the tropospheric homogeneous response, note that both the off-diagonal blocks and the off-diagonal elements of each block are mostly blank, suggesting that the sensitivities are sparse and are local in both vertical direction and variable space. Such a sparse pattern indicates that a NN that spans the entire atmospheric column may not be the most efficient implementation for predicting the correction, and a vertically localized NN may have improved performance.

Figure 6 (bottom) shows the sensitivity of the NN predicted corrections to the ancillary inputs. Against our intuition and the observed strong diurnal components in the temperature errors, the hour of the day information is the least important among all the ancillary input information. This insensitivity could result from the inclusion of thermodynamical variables, such as radiative fluxes, that may provide a sufficient source of information representing the diurnal cycle. Many of the large responses are either only in the upper levels or only in the troposphere, which is consistent with the diagonal pattern of localization in Figure 6 (top). Only a few input

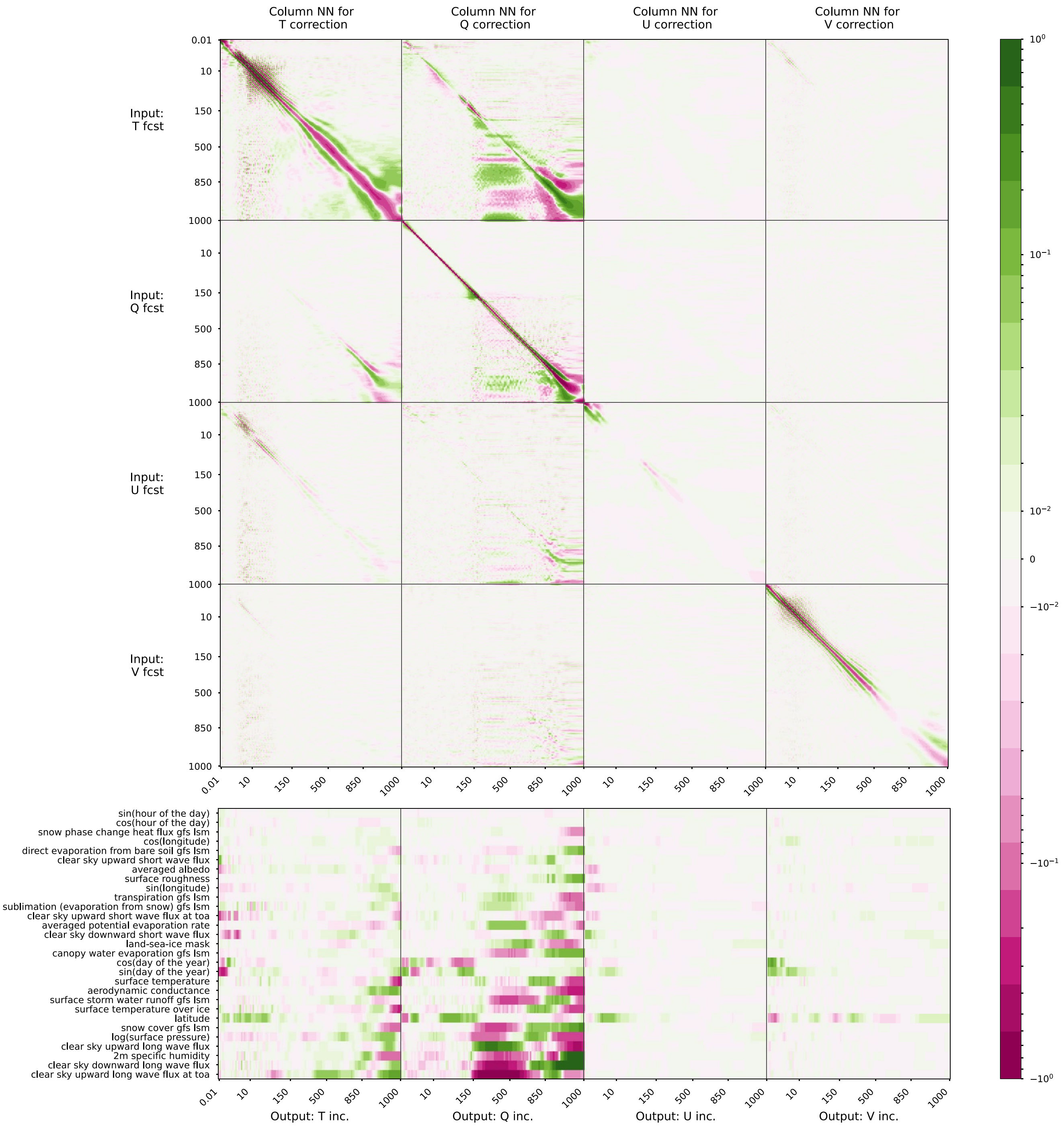

Figure 6. Sensitivity of predicted corrections to the (top) atmospheric and (bottom) ancillary (e.g., boundary condition) input variables measured by the averaged gradient of the column neural network that performs the best for each T, Q, U, and V variables.

features (e.g., *clear sky upward longwave flux at toa* for temperature and *latitude* for v-wind) show approximately the same response magnitude to both above and below 150 hPa. Given that most of the selected input features are hydrological and thermodynamic variables, they are most helpful in predicting the temperature and specific humidity corrections, but not the wind corrections.

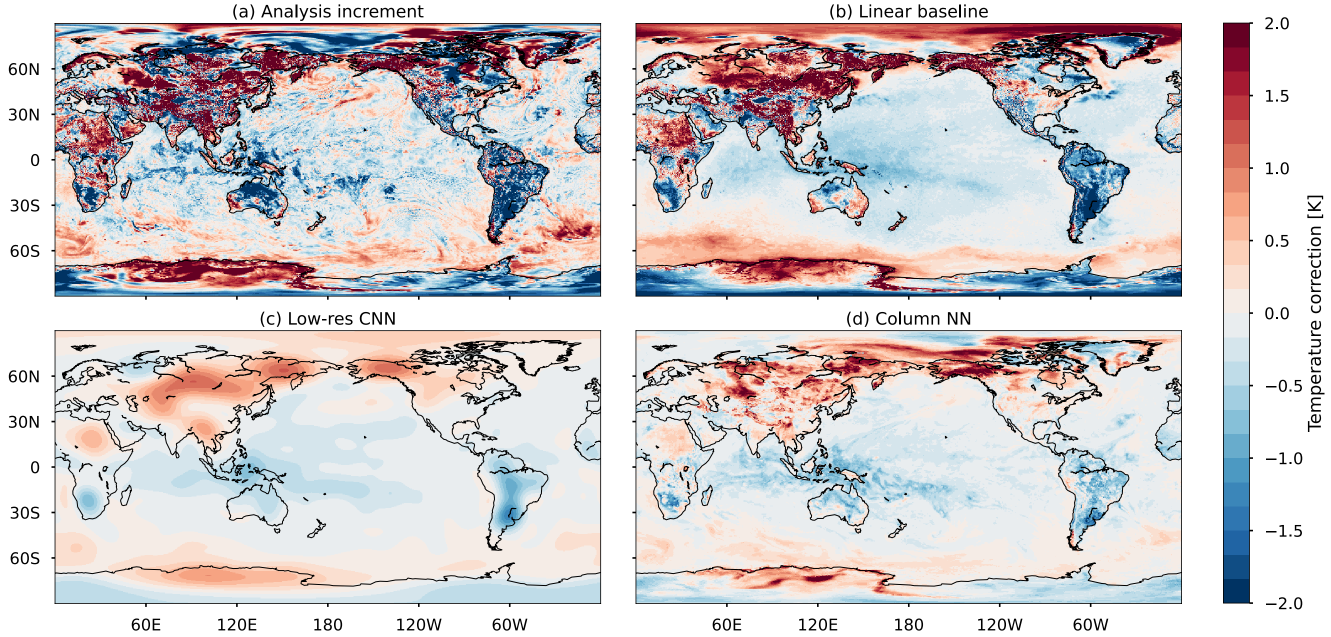

Figure 7. Comparison of the (a) original surface temperature analysis increment and the corresponding corrections [K] generated from the three error correction methods: (b) linear baseline (hourly seasonal moving average), (c) convolutional neural network trained on the smoothed data set, and (d) column neural network trained on the sampled data set.

3.3. Online Testing Performance

Here we compare the hourly seasonal average (will be referred to as linear baseline hereafter), low-res CNN, and column NN applied as online forecast error corrections.

We point out that due to the different data reduction methods and the different NN architectures, the predicted corrections from the three methods appear quite differently in the original resolution for online testing. Figure 7 compares in original resolution the prediction of surface temperature corrections from the three methods for a case extracted from the 10-day forecast experiment. The original analysis increment is also included for reference.

The corrections from the three methods appear to be filtered versions of the analysis increment, retaining different degrees of detail. This smoothing effect may be corresponding to the low signal-to-noise ratio as the analysis increment contains information other than the model error.

The linear baseline correction has granular spatial features with detailed information since it is simply a moving average of analysis increments centered on the same day of the year of the corresponding forecasts. On the other hand, the spectral data reduction of the low-res CNN smooths out all the fine features smaller than the resolved wave number. With data reduction using regular sampling, the column NN balances between the two and preserves many of the fine spatial features using the same amount of training data as the low-res CNN. All three methods agree well with one another on the larger scales. We note that the differences in fine features between the methods are smaller in higher model levels, and hypothesize that the primary source of these differences may originate from the inhomogeneity in surface conditions. At this point, it is unclear from Figure 7 whether the fine spatial features of the linear baseline and column NN are valid corrections or simply noise that should be removed. The online experiments in the following sections, which actually apply the corrections to the forecasts, will allow us to quantify the impact of these small-scale features and whether they actually reduce the forecast error.

3.3.1. Correcting Sequential 3D-Var

The improvement to the background as a function of model pressure level is shown in Figure 8. The gray shading area shows for reference the magnitude of the control RMSE (calculated against ECMWF analysis), where no corrections were applied to the forecasts. For temperature and specific humidity, the column NN correction generally outperforms the other two methods except at the surface boundary layer below 950 hPa, where the

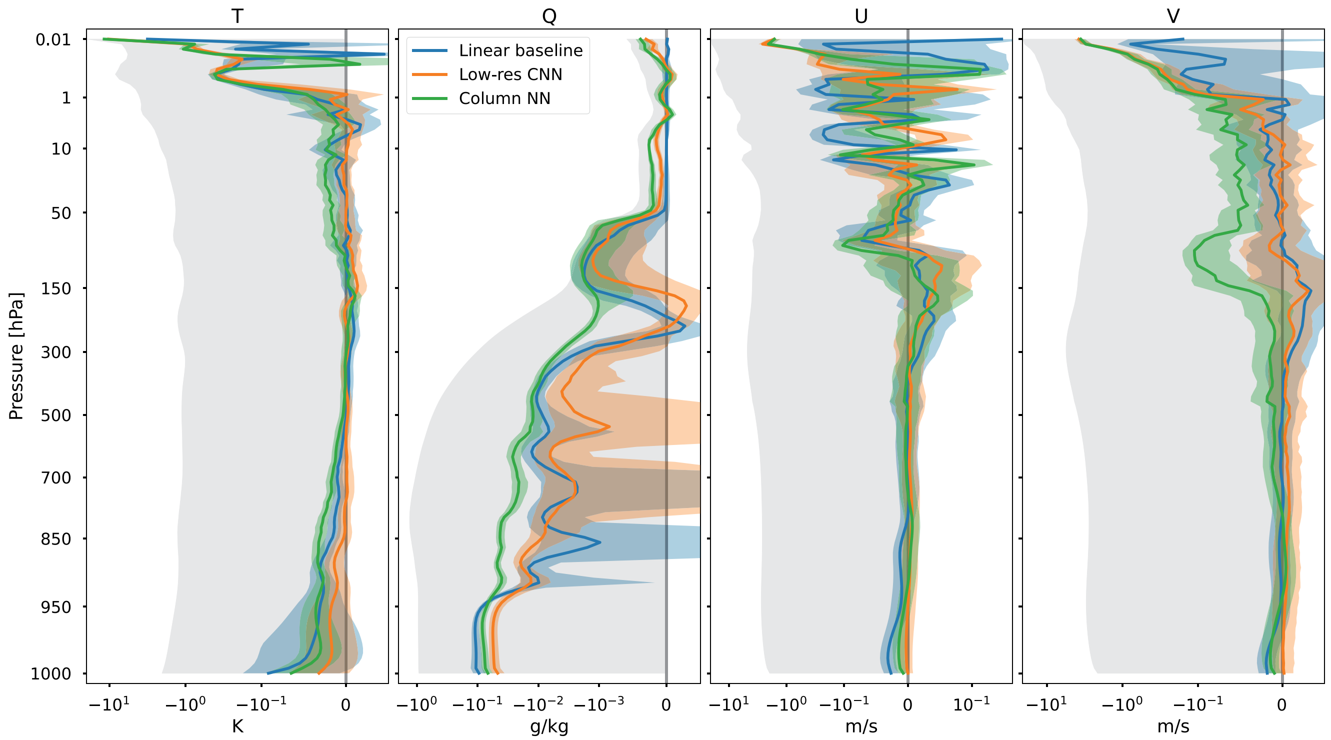

Figure 8. Averaged changes in RMSE in background forecasts for all cases as a function of model pressure level (vertical axis) by applying the error correction methods (Linear baseline: blue, Low-res convolutional neural network: yellow, and Column neural network: green) to the sequential 3D-Var experiment. The standard deviations of the RMSE changes of all cases for each method are shown as shading in the corresponding colors. Forecast improvements are shown as negative values (indicating error reduction). Changes in temperature, specific humidity, u-component wind, and v-component wind are shown respectively in columns from left to right. For reference, the light gray shading indicates the magnitude of the averaged RMSE of the control experiment, mirrored to the negative side of the *x*-axis for ease of comparison. Symmetric logarithmic scale beyond a certain threshold (0.001 for Q; 0.1 for T, U, and V) is used for the horizontal axis for accommodating large vertical variations in RMSE.

linear baseline provides the largest error reduction. In the mid to upper troposphere, all three methods provide no improvement or even slight degradation to the temperature. The humidity correction reduces the forecast error from the surface to the upper troposphere by around 10% compared to the control background. The column NN and the low-res CNN correct a huge portion of the background error at the top few levels. For u-wind, all corrections fluctuate drastically between improvement and degradation in the upper levels and are nearly zero from the surface to the middle troposphere. The linear baseline provides only slight improvements in u-wind in the lower troposphere. When compared with the relatively skillful corrections in the offline evaluation, this poor online performance may indicate a generalization issue in predicting U-wind corrections. This issue could be associated with a simple overfitting problem, but it could also suggest a more complicated situation where there is other state-dependent information in the increments that is irrelevant to estimating the model error. It would require further analysis to understand the poor performance in predicting U-wind corrections. We will pursue this analysis in future work and would advise for now against including the NN predicted U-wind corrections (especially above tropopause) in relevant applications. The column NN outperforms the linear baseline above the middle troposphere for the v-wind, especially at the upper levels. Overall, the best performing method is the column NN. The linear baseline surprisingly provides the best correction in the boundary layer. This may be due to the strong periodic component of surface errors and the granular spatial features preserved by the linear baseline. In contrast, the low-res CNN in many cases performs the worst, perhaps due to the loss of detailed spatial information. The column NN strikes a balance between preserving the fine spatial features and reducing the data size.

The standard deviations of the improvements by each method are also shown in Figure 8. We would like to point out that the surprisingly large improvement in the boundary layer temperature provided by the linear baseline is accompanied by a large spread between cases. This indicates the linear baseline correction has a rather high chance of degrading the boundary layer temperature forecast in some cases. In contrast, the spread in the same

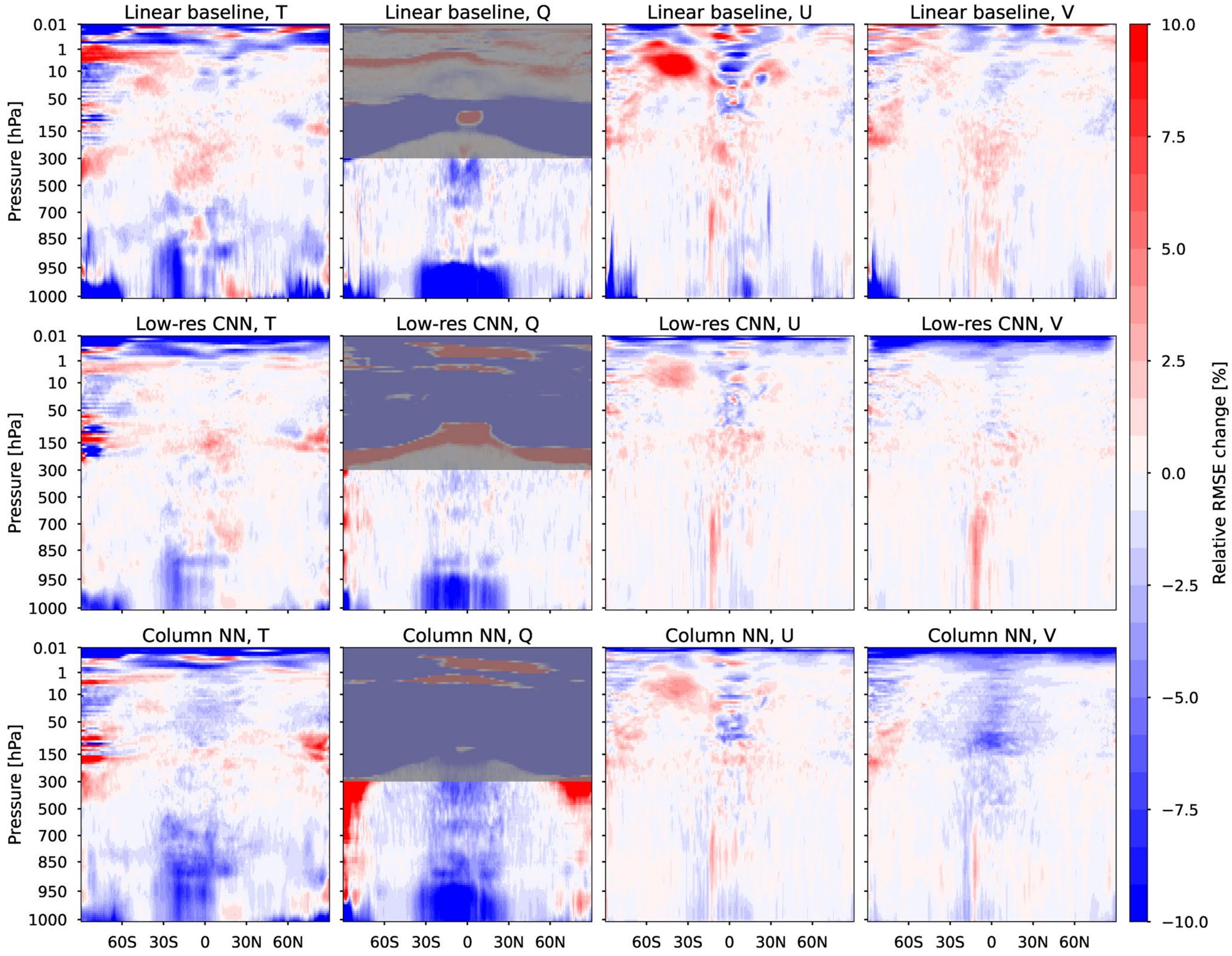

Figure 9. Zonal mean cross-section of background relative RMSE changes [%] by applying the error correction methods (Linear baseline: top, Low-res convolutional neural network: middle, and Column neural network: bottom) to the sequential 3D-Var experiment. Changes in temperature, specific humidity, u-component wind, and v-component wind are shown respectively in columns from left to right. Forecast error reductions are shown as negative values (blue). The specific humidity levels above 300 hPa are shaded owing to the trace amount of water vapor.

area is smaller for the column NN correction even though its mean improvement is not as large as that from the linear baseline.

To further examine the latitudinal distribution of the improvements, we show the relative RMSE changes in the zonal mean cross-section in Figure 9. For the temperature error, the largest improvements of all methods in near-surface levels are found in the southern tropical to subtropical regions and the higher latitude regions for both hemispheres. The southern tropical and subtropical temperature improvements extend upward to approximately 700–850 hPa. The temperature improvement is quite uniform in the top levels, except that there are a few levels with degradation in the tropics for the column NN. The improvement of specific humidity centers at the equator and extends poleward to 30°. Its vertical extension goes from the surface to 950 hPa for the linear baseline and the low-res CNN methods, but all the way to 300 hPa for the column NN. Note that the column NN degrades the forecast in the polar regions for nearly the entire troposphere column. We ignore the relative error changes for specific humidity above 300 hPa owing to the trace amount of water vapor at such high altitudes, where small changes would appear to be significant. The u-wind corrections are sporadically distributed in the surface boundary levels for linear baseline, in the top levels for the two NN methods, and in the stratosphere for all methods. For v-wind, the two NN methods both reduce the error uniformly in the top levels, and the column NN extends the improvement downward to the upper troposphere in the tropics. Note that there is a strong improvement from

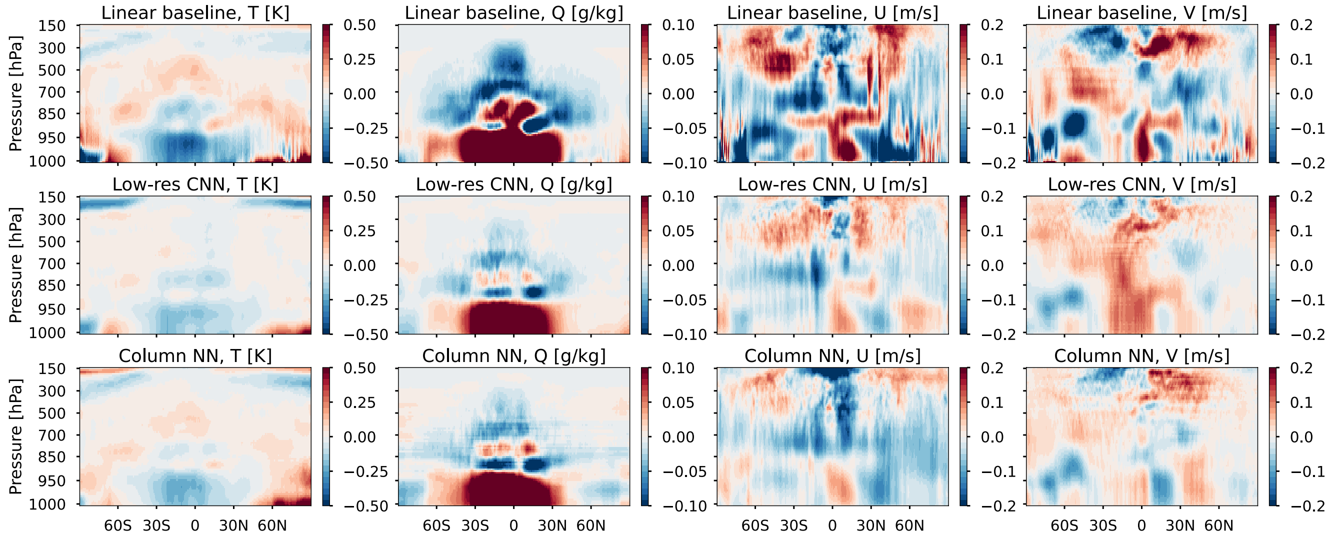

Figure 10. Zonal mean cross section of tropospheric corrections to temperature, specific humidity, U- and V- winds (left to right) from the linear baseline (top), low-res convolutional neural network (middle), and column neural network (bottom) methods. The corrections are averaged over the 3D-Var experimental period. Positive (red)/Negative (blue) value indicates that the correction increases/decreases forecast value.

the linear baseline in the southern polar region in the surface boundary levels for both u- and v-winds that are not captured by the NN methods. Overall, the column NN provides improvements to more areas, including the tropical troposphere, polar boundary layers for temperature and humidity, and upper levels for all variables, while the linear baseline captures the periodic error components and provides better surface boundary corrections.

In Figure 9, we observe a significant response in background improvement in the lower troposphere in the tropical/subtropical regions, especially in temperature and specific humidity fields. This motivates the examination of the temporally and zonally averaged corrections to each variable in the troposphere (Figure 10). Note that the overall distribution of the positive and negative correction is similar across different error correction methods, especially for the temperature and humidity fields. For temperature, all three methods show a negative correction from surface to 700 hPa and a positive correction above 700 hPa in the tropics. The specific humidity correction appears to have a similar pattern to the temperature corrections but with the sign reversed, which is consistent with a previous study (Figures 13a and 13c in Bengtsson et al., 2019). These features indicate the model has a consistently warm and dry bias in the lower boundary layer while having a cold and wet bias in the upper troposphere. The wind corrections are rather complicated, but the V-wind correction shows the error correction methods enhance a convergent flow below 950 hPa and a divergent flow between 150 and 400 hPa at the equator. These features in the averaged temperature, humidity, and V-wind corrections indicate a Hadley-like systematic error in the model. We also point out that the averaged linear baseline correction is equivalent to an average of increments, which corresponds to Figure 15 of Crawford et al. (2020), in which the specific humidity correction appears qualitatively similar to that in Figure 10.

3.3.2. Ten-Day Forecast Correction

Figure 11 compares the error changes caused by error correction methods as a function of model levels and forecast lead times for all variables. Overall, the NN methods provide improvements that increase with forecast lead time for most levels, except for one of the top levels for temperature (0.1 hPa) and another for u-wind (10 hPa). The column NN performs slightly better than the low-res CNN with a similar pattern. The linear baseline corrections are mixed with both improvement and degradation in the forecasts at different lead times. Some degraded levels start with a slight increase of error, but the error grows with the increased lead times, such as the layers around 10 hPa for temperature, u- and v-wind. Another interesting type of forecast degradation emerges at later forecast lead times from the earlier improvements, such as the temperature forecasts at 300–950 hPa and the specific humidity forecasts from 700 hPa to the surface. This interesting sign change takes place somewhere between 2 and 6 days and is an indication of the over-correction also observed by Crawford et al. (2020). The corrections to temperature and specific humidity in the lower troposphere (below 950 hPa) are the only few regions where

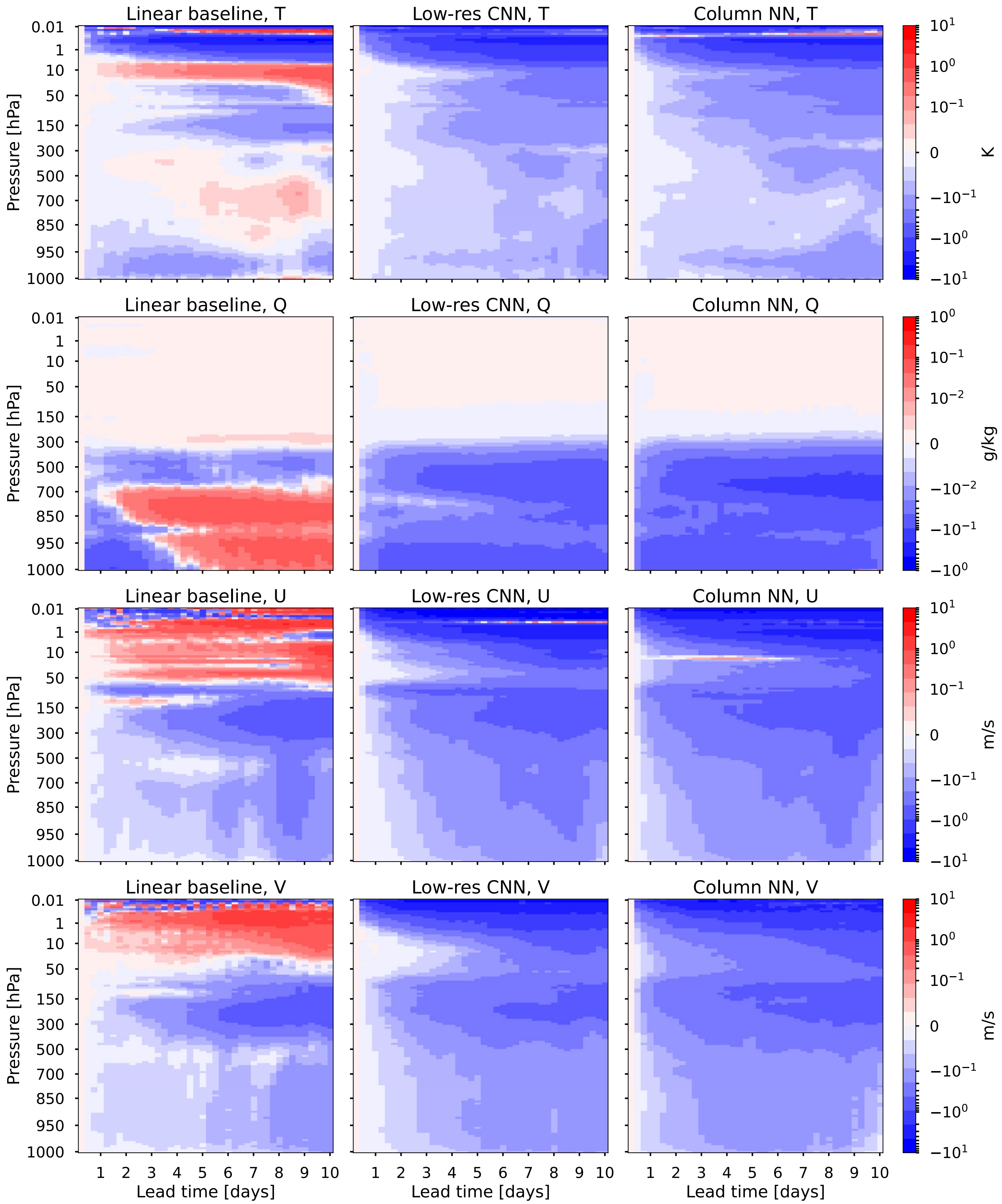

Figure 11. Forecast RMSE change averaged over 30 cases as a function of model pressure level (vertical axis) and forecast lead time (horizontal axis) by consecutively applying the error correcting methods (linear baseline: left, low-res convolutional neural network: center, Column neural network: right) to 10-day forecasts for every 6h segments. Changes in temperature, specific humidity, u-component wind, and v-component wind are shown respectively in rows from top to bottom. Blue represents the forecast improvement, whereas red indicates degradation.

the linear baseline outperforms the NN methods. However, the humidity corrections go from error reduction to error increase after 4 days. We observe no change of sign for the NN methods, indicating that the corrections are state-dependent and less likely to overcorrect the forecasts. The levels of the largest improvement at the early lead times are consistent with the 3D-Var results, except for the u-winds where the improvements were not obvious in the previous experiment but quite large in the 10-day forecast results. We suspect that the improvement in u-wind may come from the improvement in v-wind owing to their high correlation. This guess may be supported by the similarity between the u- and v-wind error changes in the figure. At later lead times, there are improved levels that appear to be an extension of the nearby levels that are largely improved from earlier lead times.

4. Conclusions

In this study, a NN-based online correction is applied to the NOAA FV3-GFS model with a relatively close-to-operation configuration for demonstrating, for the first time, the potential of reducing systematic model errors in NWP tasks, including cycling DA and medium-range forecasts.

We systematically compare the linear baseline similar to Crawford et al. (2020), a state-dependent 1D column NN similar to BL20, and a more complicated convolutional NN, which is an extension of the 1D column NN. Our study finds that the 1D column NN is capable of reconstructing the global variability of the systematic model error as revealed in our linear baseline (Figure 10). Similar to prior work (Crawford et al., 2020), this global variability has a Hadley-like structure and may correspond to the systematic error in tropical convection activities. When we compare linear baseline to state-dependent correction generated with the NN, we find state-dependent corrections considerably improve error predictions in all of our tests, including offline testing, cycling DA, and 10-day forecasts. We also find that state-dependent corrections provided by the NN avoid the problem of over-correction of bias in the extended range forecasts by the linear baseline (as was documented by Crawford et al. (2020) and replicated in this study). We attribute this to the capability of the NN on predicting the corrections conditioned to the forecast states. Comparisons between the 1D column NN corrections (originally introduced by BL20) and the more sophisticated convolution network (introduced in this paper) showed that the inclusion of horizontal information has a very limited positive impact in the offline tests but had a neutral impact on tests with cycling DA and 10-day forecasts. We infer that the nature of the short-term model error (as revealed in the analysis increments) is dominated by vertical processes such as moist physics, vertical mixing, cloud microphysics, radiation, and gravity wave drag.

We examine the sensitivity of the NN-predicted corrections to the input features and reveal a highly localized dependency structure in the vertical direction and in the variable space between the two. The temperature and specific humidity corrections are found to be highly dependent on each other's forecasts, and the corrections mostly depend on the forecasts in nearby vertical levels. Such a vertical localization of dependency is the strongest in the upper atmosphere, while both temperature and specific humidity in the troposphere show a rather homogeneous response of a thick layer to forecasts at certain levels. The sensitivity to the ancillary information reveals that the radiative fluxes may be a more generalizable input feature than time information indicated by the strong periodic components revealed by the linear baselines while the NNs are not particularly sensitive to the time of the day and day of the year input features.

Our sensitivity analysis points to a future direction for improving the NN structure. The sparse and localized features suggest multiple highly localized NN for different vertical levels may provide a more accurate and efficient prediction of the error corrections. Our results in the cycling DA and 10-day forecast cycles also encourage us to implement an online evaluator of the NN in the FV3-GFS model to avoid the need to start and stop the model to produce background forecast files for ingesting in stand-alone NN evaluators. Another promising application to extend this work is to address model biases in the context of the historic reanalysis. Specifically, we showed that it is possible to detect, learn, and correct model biases with a modern observing system. However, as reanalyses are extended backward in time the observational system becomes sparse and insufficient to correct for model biases. This was manifested in the previous reanalysis as discontinuities that correspond to the introduction of new observing systems. If one can apply systematic error corrections learned from the modern system to historic periods, one might be able to avoid these artificial discontinuities that complicate the use of the reanalysis products for studies of long-term climate trends. Lastly, the analysis increments may not be the only source for learning model errors. Observation innovations from certain trustworthy observations can also provide useful information about systematic model errors (e.g., Laloyaux et al., 2022).

Data Availability Statement

The source code for the FV3-GFS model can be found at https://github.com/ufs-community/ufs-weather-model. The data assimilation and replay workflows are available at https://github.com/jswhit/da_scripts and https://github.com/jswhit/replay_scripts. The data reduction and training scripts are available at https://github.com/NOAA-PSL/model_error_correction. The data was processed using the climate data operators (CDO; Schulzweida, 2022).

Acknowledgments

T.-C. Chen is supported by the National Oceanographic and Atmospheric Administration (NOAA) Cooperative Agreement with CIRES, NA17OAR4320101. S.G. Penny acknowledges support from NOAA Grants NA20OAR4600277, NA19NES4320002, and NA18NWS4680048, and the Office of Naval Research (ONR) Grants N00014-19-1-2522 and N00014-20-1-2580. R. P was supported by the National Science Foundation via the NSF STC Learning the Earth with Artificial Intelligence and Physics (LEAP), NSF Award 2019625.

References

Bengtsson, L., Dias, J., Gehne, M., Bechtold, P., Whitaker, J., Bao, J.-W., et al. (2019). Convectively coupled equatorial wave simulations using the ECMWF IFS and the NOAA GFS cumulus convection schemes in the NOAA GFS model. *Monthly Weather Review*, *147*(11), 4005–4025. https://doi.org/10.1175/MWR-D-19-0195.1

Bloom, S. C., Takacs, L. L., Silva, A. M. d., & Ledvina, D. (1996). Data assimilation using incremental analysis updates. *Monthly Weather Review*, *124*(6), 1256–1271. https://doi.org/10.1175/1520-0493(1996)124<1256:DAUIAU>2.0.CO;2

Bonavita, M., & Laloyaux, P. (2020). Machine learning for model error inference and correction. *Journal of Advances in Modeling Earth Systems*, *12*(12), e2020MS002232. https://doi.org/10.1029/2020MS002232

Brenowitz, N. D., & Bretherton, C. S. (2018). Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, *45*(12), 6289–6298. https://doi.org/10.1029/2018GL078510

Brenowitz, N. D., & Bretherton, C. S. (2019). Spatially extended tests of a neural network parametrization trained by coarse-graining. *Journal of Advances in Modeling Earth Systems*, *11*(8), 2728–2744. https://doi.org/10.1029/2019MS001711

Bretherton, C. S., Henn, B., Kwa, A., Brenowitz, N. D., Watt-Meyer, O., McGibbon, J., et al. (2022). Correcting coarse-grid weather and climate models by machine learning from global storm-resolving simulations. *Journal of Advances in Modeling Earth Systems*, *14*(2), e2021MS002794. https://doi.org/10.1029/2021MS002794

Crawford, W., Frolov, S., McLay, J., Reynolds, C. A., Barton, N., Ruston, B., & Bishop, C. H. (2020). Using analysis corrections to address model error in atmospheric forecasts. *Monthly Weather Review*, *148*(9), 3729–3745. https://doi.org/10.1175/MWR-D-20-0008.1

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., et al. (2020). The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, *146*(730), 1999–2049. https://doi.org/10.1002/qj.3803

Laloyaux, P., Kurth, T., Dueben, P. D., & Hall, D. (2022). Deep learning to estimate model biases in an operational NWP assimilation system. *Journal of Advances in Modeling Earth Systems*, *14*(6), e2022MS003016. https://doi.org/10.1029/2022MS003016

Lei, L., & Whitaker, J. S. (2016). A four-dimensional incremental analysis update for the ensemble Kalman filter. *Monthly Weather Review*, *144*(7), 2605–2621. https://doi.org/10.1175/MWR-D-15-0246.1

Lin, S.-J. (2004). A "vertically Lagrangian" finite-volume dynamical core for global models. *Monthly Weather Review*, *132*(10), 2293–2307. https://doi.org/10.1175/1520-0493(2004)132<2293:AVLFDC>2.0.CO;2

Mamalakis, A., Ebert-Uphoff, I., & Barnes, E. A. (2022). Neural network attribution methods for problems in geoscience: A novel synthetic benchmark dataset. *Environmental Data Science*, *1*, e8. https://doi.org/10.1017/eds.2022.7

Ott, J., Pritchard, M., Best, N., Linstead, E., Curcic, M., & Baldi, P. (2020). A Fortran-Keras deep learning bridge for scientific computing. arXiv:2004.10652 [cs]. Retrieved from http://arxiv.org/abs/2004.10652

Putman, W. M., & Lin, S.-J. (2007). Finite-volume transport on various cubed-sphere grids. *Journal of Computational Physics*, *227*(1), 55–78. https://doi.org/10.1016/j.jcp.2007.07.022

Schulzweida, U. (2022). CDO User Guide. Zenodo.https://doi.org/10.5281/ZENODO.7112925

UFS Community. (2020). UFS weather model [Dataset]. Zenodo. Retrieved from https://zenodo.org/record/4460292

Watt-Meyer, O., Brenowitz, N. D., Clark, S. K., Henn, B., Kwa, A., McGibbon, J., et al. (2021). Correcting weather and climate models by machine learning nudged historical simulations. *Geophysical Research Letters*, *48*(15), e2021GL092555. https://doi.org/10.1029/2021GL092555