Towards a Self-contained Data-driven Global Weather Forecasting Framework

Yi Xiao [1 2] **Lei Bai** [2] **Wei Xue** [1 3] **Hao Chen** [2] **Kun Chen** [2 4] **Kang Chen** [2] **Tao Han** [2] **Wanli Ouyang** [2]

Abstract

Data-driven weather forecasting models are advancing rapidly, yet they rely on initial states (i.e., analysis states) typically produced by traditional data assimilation algorithms. Four-dimensional variational assimilation (4DVar) is one of the most widely adopted data assimilation algorithms in numerical weather prediction centers; it is accurate but computationally expensive. In this paper, we aim to couple the AI forecasting model, FengWu, with 4DVar to build a self-contained data-driven global weather forecasting framework, FengWu-4DVar. To achieve this, we propose an *AI-embedded* 4DVar algorithm that includes three components: (1) a 4DVar objective function embedded with the FengWu forecasting model and its error representation to enhance efficiency and accuracy; (2) a spherical-harmonic-transform-based (SHT-based) approximation strategy for capturing the horizontal correlation of background error; and (3) an auto-differentiation (AD) scheme for determining the optimal analysis fields. Experimental results show that under the ERA5 simulated observational data with varying proportions and noise levels, FengWu-4DVar can generate accurate analysis fields; remarkably, it has achieved stable self-contained global weather forecasts for an entire year for the first time, demonstrating its potential for real-world applications. Additionally, our framework is approximately 100 times faster than the traditional 4DVar algorithm under similar experimental conditions, highlighting its significant computational efficiency.

[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China [2]Shanghai Artificial Intelligence Laboratory, Shanghai, China [3]Qinghai University and Intelligent Computing and Application Laboratory of Qinghai Province, China [4]School of Information Science and Technology, Fudan University, Shanghai, China. Correspondence to: Wei Xue <xuewei@tsinghua.edu.cn>, Lei Bai <baisanshi@gmail.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

Weather forecasting is the cornerstone of human society and profoundly impacts all aspects of our daily lives and economic activities. Accurately predicting weather conditions in advance is beneficial to various sectors, ranging from agricultural planning to renewable energy generation and disaster preparedness.

Traditional numerical weather forecasting methods rely on building partial differential equations based on physical rules and solving these equations for accurate predictions (Kalnay, 2003). However, it is difficult to accurately resolve complex physical processes like clouds and convection, which makes medium-range forecasts less accurate (Hourdin et al., 2017; Donner et al., 2011). Moreover, solving these equations is computationally expensive, leading to significant investment in supercomputers for weather forecasting (Bauer et al., 2015). In the past few years, a multitude of Artificial Intelligence (AI) weather forecasting models have emerged as a promising alternative, such as FourCastNet (Pathak et al., 2022), Pangu Weather (Bi et al., 2023), GraphCast (Lam et al., 2022), FengWu (Chen et al., 2023a), FuXi (Chen et al., 2023b), etc. These data-driven models are informed by modern neural network structures (Vaswani et al., 2017; Liu et al., 2021; Zhao et al., 2023; Wang et al., 2023), and their forecast accuracy rivals or even surpasses traditional methods like Integrated Forecasting System (IFS) developed by European Centre for Medium-Range Weather Forecasts (ECMWF). Notably, they exhibit forecasting efficiency orders of magnitude higher than that of traditional algorithms.

Despite the potential success of AI forecasting models, most previous works have overlooked a crucial component of weather forecasting systems: *data assimilation*. Data assimilation is a statistical technique that combines observational information with numerical models to achieve an optimal estimate of the current state (also known as the analysis state), which then serves as the initial value for predicting future states. In previous studies, the establishment of initial states is usually achieved by assimilating the forecast fields of physics-based forecasting models, as shown in Figure 1(a). However, these initial states often face issues such as inconsistency with AI forecasting models and high computational complexity, which hinder the accuracy and

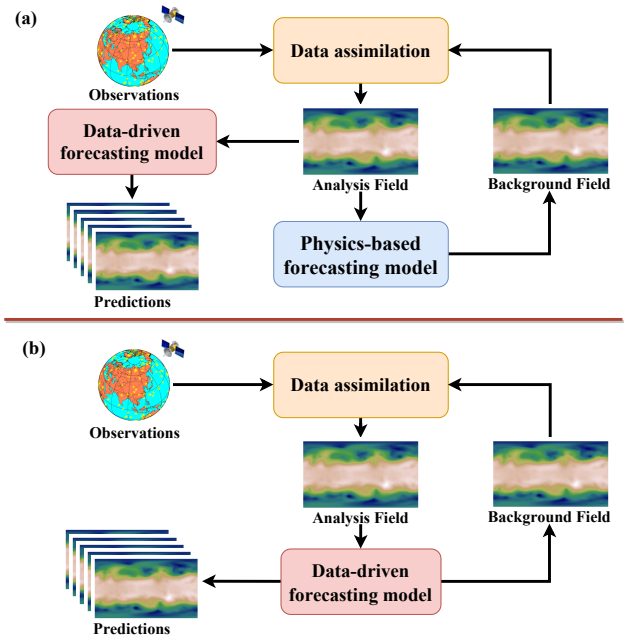

Figure 1. (a) **Previous works.** The physics-based weather forecasting model is required to provide initial states. (b) **Our work.** The data-driven weather forecasting model can operate in a self-contained manner.

efficiency of the entire data-driven weather prediction system. The approach to resolving this issue is to couple the AI weather forecasting model with data assimilation and build a data-driven weather forecasting framework capable of realizing continuous forecasts in a *self-contained* manner, as shown in Figure 1(b).

Four-dimensional variational assimilation (4DVar) is one of the most popular data assimilation algorithms and has been successfully adopted in numerical weather prediction centers worldwide (Rabier et al., 1998; 2000). The main problem of 4DVar lies in its high computational cost. In the traditional 4DVar algorithm, the physics-based forecasting model is involved in its objective function to represent the so-called flow dependency (Rabier et al., 1998). Due to the high computational complexity of the physics-based model itself, the computational cost of the entire algorithm is also very high. For example, in the China Meteorological Administration, realizing 4DVar on the 1° flow dependency resolution takes about 50 minutes on 256 processors of the PI-SUGON high-performance computer (Zhang et al., 2019). Therefore, directly coupling the traditional 4DVar algorithm with the AI forecasting model will greatly hinder the efficiency gain of the AI forecasting model.

Geer (2021) suggests that by introducing AI forecasting models to 4DVar as the flow dependency and solving the optimal analysis field with the aid of auto-differentiation, it is possible to reduce the computational cost. Although this method has achieved some success on simple dynamical systems (Dong et al., 2022), several challenges remain when scaling to the real-world global weather forecasting system. First, in a high-dimensional forecasting system, the correlation of the background error is very complex, and if we follow the convention of AI data assimilation on low-dimensional systems and use the diagonal matrix to approximate the error covariance, a lot of information will be lost and the assimilation will not work (Kalnay, 2003; Fisher, 2003). Second, compared with low-dimensional dynamical systems, the errors of global AI forecasting models grow relatively quickly (Bi et al., 2023), which will greatly reduce the representative accuracy of flow dependencies in the 4DVar objective function, thereby hindering the final assimilation accuracy.

In this paper, we aim to resolve these issues and design an *AI-embedded* 4DVar algorithm on the global weather system. For the first time, the 4DVar algorithm is coupled with a global AI forecasting model to achieve a self-contained data-driven weather forecasting framework. We leverage three techniques to achieve this goal. First, inspired by Geer (2021), we embed the AI forecasting model into the flow dependency of 4DVar to reduce the computational cost of physics-based models. Considering that the error accumulation of AI forecasting models is relatively faster than the traditional model, we go one step further and add the error covariance term to the objective function to improve the assimilation accuracy. Second, we take advantage of the spherical harmonic transform to implement the differentiable spherical convolution for approximating horizontal correlations of the background error. Third, we utilize the auto-differentiation technique to solve the data assimilation problem and find the optimal analysis states, eliminating the need of manually coding adjoint models, which is required by the traditional 4DVar.

We conduct this research on the global AI weather forecasting model, FengWu, and couple it with AI-embedded 4DVar to implement an AI weather forecasting framework, FengWu-4DVar[1]. Our experiments are conducted with the FengWu forecasting model at a 1.4° resolution and the observations simulated from the ERA5 reanalysis data. When the observation proportion is between 5% and 15% and the assimilation window is set to 6 hours, FengWu-4DVar is able to generate reasonable analysis fields and achieve stable and efficient cyclic assimilation and forecasting for at least one year. With an observation proportion of 15%, the accuracy of the analysis fields is comparable to that of the 6-hour forecast of IFS. Moreover, assimilating observations in a 6-hour window can be realized in less than 30

[1]The code of FengWu-4DVar is available at https://github.com/OpenEarthLab/FengWu-4DVar.

seconds on one NVIDIA A100 GPU, 100 times faster than the traditional 4DVar on 256 processors of the PI-SUGON high-performance computer.

2. Fundamentals of 4DVar

Denote $\mathbf{x}_t$ the physical states and $\mathbf{y}_t$ the observations at time t. Then, the 4DVar algorithm estimates the optimal physical state at time $t = 0$ by minimizing the following objective function:

$$\begin{aligned} J(\mathbf{x}_0) = &\frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}^b\right)^{\mathrm{T}} \mathbf{B}^{-1}\left(\mathbf{x}_0 - \mathbf{x}^b\right) \\ &+ \frac{1}{2}\sum_{\tau=0}^{T-1}\left(\mathcal{H}\left(\mathbf{x}_\tau\right) - \mathbf{y}_\tau\right)^{\mathrm{T}} \mathbf{R}_\tau^{-1}\left(\mathcal{H}\left(\mathbf{x}_\tau\right) - \mathbf{y}_\tau\right) \\ \mathbf{x}_\tau = &\mathcal{M}_{0\to\tau}(\mathbf{x}_0) \end{aligned} \tag{1}$$

The objective is to compute a maximum likelihood estimate for the initial state $\mathbf{x}_0$ of a trajectory $(\mathbf{x}_0, \ldots, \mathbf{x}_{T-1})$ evolved through a physical model $\mathcal{M}$, given a sequence of observations $\{\mathbf{y}_\tau\}_{\tau=0}^{T-1}$ and a prior estimate $\mathbf{x}^b$. We note here that the subscript $0 \to \tau$ of $\mathcal{M}$ stands for integration from time 0 to time τ. Since we only consider *autonomous* systems (Strogatz, 2018) in this paper, $\mathcal{M}_{0\to\tau} = \mathcal{M}_{t\to t+\tau}$ holds for any t, thus we may also rewrite it as $\mathcal{M}_\tau$. The observation operator $\mathcal{H}$ maps physical states into the observation space. For example, physical fields are often modeled on a regular grid, while the positions of observation stations are typically distributed irregularly. Observation operators can map the values of the physical field at regular grid points to the positions of observation stations.

The loss function $J(\mathbf{x}_0)$ characterizes both the initial condition and the conditional distribution of observations as multivariate normal distributions. The first term incorporates a guess for the initial state $\mathbf{x}_0$ (referred to as the background field $\mathbf{x}^b$), where $\mathbf{B}$ is a background covariance matrix representing the uncertainty associated with this assumption. The second term incorporates the observations at different time steps, and the error variance of observations at time τ is represented by the matrix $\mathbf{R}_\tau$. The *flow dependency* refers to the feature of the 4DVar objective that the initial state is integrated (by the forecasting model $\mathcal{M}$) to generate a sequence of states to evaluate its deviation from the observation sequence over the entire time interval. In Equation 1, the interval $[0,\ T-1]$ is often referred to as the *assimilation window* (Trémolet, 2006).

Function Optimization In 4DVar, the objective function is minimized via gradient-based optimization algorithms like L-BFGS (Jorge & Stephen, 2006). The gradient of $J(\mathbf{x}_0)$ can be formulated as

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{x}_0} = &\mathbf{B}^{-1}\left(\mathbf{x}_0 - \mathbf{x}^b\right) \\ &+ \sum_{\tau=0}^{T-1} \mathbf{M}_{\tau\to 0}^{\mathrm{T}} \mathbf{H}^{\mathrm{T}} \mathbf{R}_\tau^{-1}\left(\mathcal{H}\left(\mathbf{x}_\tau\right) - \mathbf{y}_\tau\right), \end{aligned} \tag{2}$$

where $\mathbf{M}_{\tau\to 0}^{\mathrm{T}} = \left(\frac{\partial \mathcal{M}_{0\to\tau}(\mathbf{x})}{\partial \mathbf{x}}\right)^{\mathrm{T}}$ is the adjoint model (Rabier & Liu, 2003) of $\mathcal{M}_{0\to\tau}$ and $\mathbf{H} = \frac{\partial \mathcal{H}(\mathbf{x})}{\partial \mathbf{x}}$ is the linearized observation operator. In the traditional 4DVar, $\mathcal{M}$ corresponds to the physics-based forecasting model and its adjoint model $\mathbf{M}_{\tau\to 0}^{\mathrm{T}}$ is coded manually (Trémolet, 2006). Due to the high computational complexity of the forward forecasting model, the adjoint model also bears a substantial computational burden. This load is further amplified in optimization algorithms, which require multiple gradient calculations through iterative processes.

Cyclic Forecasting In operational weather forecasting centers, establishing a self-contained forecasting system is achievable through alternatively operating model forecasts and data assimilation. In the data assimilation stage, the observational data is utilized to correct the prediction field at the current moment (i.e., background field) to obtain more accurate analysis field. In the prediction stage, starting from the analysis field at the current moment, the numerical prediction model is applied to integrate to obtain the forecast field (background field) at the subsequent moment. As time goes by, new observations at the subsequent moment can be obtained, starting a new round of "analysis-prediction" cycle. This process is called cyclic forecasting.

3. AI-embedded 4DVar on a Global Forecasting Model

3.1. AI-embedded 4DVar Objective Function

The major computational cost of 4DVar lies in the calculation of $\mathcal{M}$ and its adjoint $\mathbf{M}^{\mathrm{T}}$. Inspired by Dong et al. (2022) and Geer (2021), we substitute the physics-based forecasting model $\mathcal{M}$ in the objective function with a data driven model $\mathcal{M}^{ml}$ to reduce the computational cost of the 4DVar algorithm. Another advantage of the AI-embedded 4DVar objective function is that the flow dependencies are consistent with the AI forecasting model, which will lead to better cyclic forecasting results.

Moreover, considering that the errors of global AI forecasting models grow relatively quickly (Bi et al., 2023), the flow dependency in the objective function is inaccurate, which hinders the final assimilation accuracy. To deal with this issue, we explicitly take model error into account in the 4DVar objective function to improve assimilation accuracy. Specifically, assuming that the error of the forecasting

model for integrating τ steps follows a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{Q}_\tau)$ (Howes et al., 2017) and that the observation operator is linear, i.e., $\mathcal{H}(\mathbf{x}) = \mathbf{H}\mathbf{x}$, we can re-derive the objective function according to the Bayes' theorem. We let readers refer to the Appendix for the detailed derivation. The form of the new objective function is almost identical to the original one, except that the model error covariance matrices are added to the observation covariance matrices to give smaller confidence to the future observations. That is,

$$\begin{aligned} J(\mathbf{x}_0) = &\frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}^b\right)^{\mathrm{T}} \mathbf{B}^{-1}\left(\mathbf{x}_0 - \mathbf{x}^b\right) \\ &+ \frac{1}{2}\sum_{\tau=0}^{T-1} \mathbf{d}_\tau^{\mathrm{T}}\left(\mathbf{R}_\tau + \mathbf{H}\mathbf{Q}_\tau\mathbf{H}^{\mathrm{T}}\right)^{-1}\mathbf{d}_\tau \\ \mathbf{d}_\tau = &\mathcal{H}\left(\mathcal{M}^{ml}_{0\to\tau}\left(\mathbf{x}_0\right)\right) - \mathbf{y}_\tau. \end{aligned} \tag{3}$$

3.2. SHT-based Horizontal Correlation Approximation

The background covariance matrix $\mathbf{B}$ describes the uncertainty of the forecasting model. In most previous works of AI data assimilation (Fablet et al., 2021; Frerix et al., 2021; Dong et al., 2022), the covariance matrix is assumed to be diagonal, that is, only the variance is taken into consideration and the correlational information is ignored. Such an assumption simplifies the algorithm implementation, yet compromises assimilation accuracy as how observations at a single point affect the estimate of its neighbourhood is not encoded into the $\mathbf{B}$ matrix. In the global weather forecasting system, the observations are sparse; therefore, it's infeasible to make such a simplification.

To address this issue, we introduce the horizontal correlation into the $\mathbf{B}$ matrix. Due to the large horizontal spacing, the vertical correlation is relatively weak and is not considered in this work. Inspired by Bannister (2008), Barker et al. (2004) and Descombes et al. (2015), instead of assuming $\mathbf{B}$ to be diagonal, we assume that $\mathbf{B}$ can be decomposed as $\mathbf{B} = \mathbf{U}\mathbf{U}^{\mathrm{T}}$, where $\mathbf{U}$ is a sparse matrix for representing horizontal correlations. By defining the control variable $\mathbf{u}_0 = \mathbf{U}^{-1}\left(\mathbf{x}_0 - \mathbf{x}^b\right)$, we can transform the original objective function, where $\mathbf{x}_0$ serves as the variable, into an objective function, where $\mathbf{u}_0$ is the variable:

$$\begin{aligned} \tilde{J}(\mathbf{u}_0) = &\frac{1}{2}\mathbf{u}_0^{\mathrm{T}}\mathbf{u}_0 + \frac{1}{2}\sum_{\tau=0}^{T-1} \mathbf{d}_\tau^{\mathrm{T}}\left(\mathbf{R}_\tau + \mathbf{H}\mathbf{Q}_\tau\mathbf{H}^{\mathrm{T}}\right)^{-1}\mathbf{d}_\tau \\ \mathbf{d}_\tau = &\mathcal{H}\left(\mathcal{M}^{ml}_{0\to\tau}\left(\mathbf{U}\mathbf{u}_0 + \mathbf{x}^b\right)\right) - \mathbf{y}_\tau. \end{aligned} \tag{4}$$

The key problem lies in the calculation of $\mathbf{U}\mathbf{u}_0$. Related works suggest that the matrix multiplication of $\mathbf{U}\mathbf{u}_0$ can be approximated by convolving $\mathbf{u}_0$ with a Gaussian distribution kernel, the parameters of which are obtained by the NMC method (Descombes et al., 2015). Since we represent the horizontal correlation on the sphere, the convolution needs to be a spherical one. According to the convolution theorem, the convolution operation in the physical domain can be equivalently converted to the multiplication operation in the spectral domain (Driscoll & Healy, 1994). Denoting $\mathcal{F}$ the spherical harmonic transform (SHT), κ the convolution kernel and u the field to be convolved, the convolution can be achieved according to the following formula:

$$\mathcal{F}[\kappa \star u](l, m) = 2\pi\sqrt{\frac{4\pi}{2l+1}}\mathcal{F}[u](l, m)\cdot\mathcal{F}[\kappa](l, 0). \tag{5}$$

In our work, the "torch-harmonics" package developed by NVIDIA is applied to efficiently implement the differentiable SHT (Bonev et al., 2023).

3.3. AD Scheme for Solving the Analysis Field

To optimize the 4DVar objective function, we construct a new "neural network" with $\mathbf{u}_0$ (the control variable of the analysis fields to be optimized) as the input, and use this "neural network" to calculate the "loss" $\tilde{J}(\mathbf{u}_0)$. Noting that $\mathcal{M}^{ml}$ is a differentiable AI model and other calculations like matrix multiplication and addition are also differentiable, if we fix the parameters of the AI model $\mathcal{M}$ and regard the input $\mathbf{u}_0$ as the "parameters" of our "neural network", the back-propagation algorithm can be implemented through auto-differentiation and thereafter the gradient $\frac{\partial\tilde{J}}{\partial\mathbf{u}_0}$ can be directly obtained. With this approach, we no longer need to build the adjoint model manually, which saves substantial engineering effort. In the Appendix, we provide further explanation on the equivalence between AD and manual coding. After calculating the gradient, the optimizer packages in PyTorch can be applied to solve the function optimization problem. Different from neural network training in which a batch of samples are utilized for optimization, in AI-embedded 4DVar, only one sample is involved. Therefore, batch optimization algorithms like SGD and ADAM are not feasible for this task. Quasi-Newton optimization algorithms like L-BFGS are employed in this work instead.

3.4. Coupling with the Forecasting Model

Denoting $\tilde{J}\left(\cdot\middle|\mathbf{x}^b, \{\mathbf{y}_\tau\}_{\tau=0}^{T-1}\right)$ the objective function with respect to observations $\{\mathbf{y}_\tau\}_{\tau=0}^{T-1}$ and the background field $\mathbf{x}^b$, the coupling of the AI-embedded 4DVar algorithm with model forecasts is implemented as shown in Algorithm 1, where the outputs $\{\mathbf{x}^b_{lT}\}_{l=0}^{L}$ and $\{\mathbf{x}^a_{lT}\}_{l=0}^{L}$ represent the sequences of background fields and analysis fields, respectively.

4. Results

4.1. Experimental Setup

Forecasting Model Setup Our experiments are conducted with FengWu (Chen et al., 2023a), a data-driven global medium-range weather forecasting model. We choose this

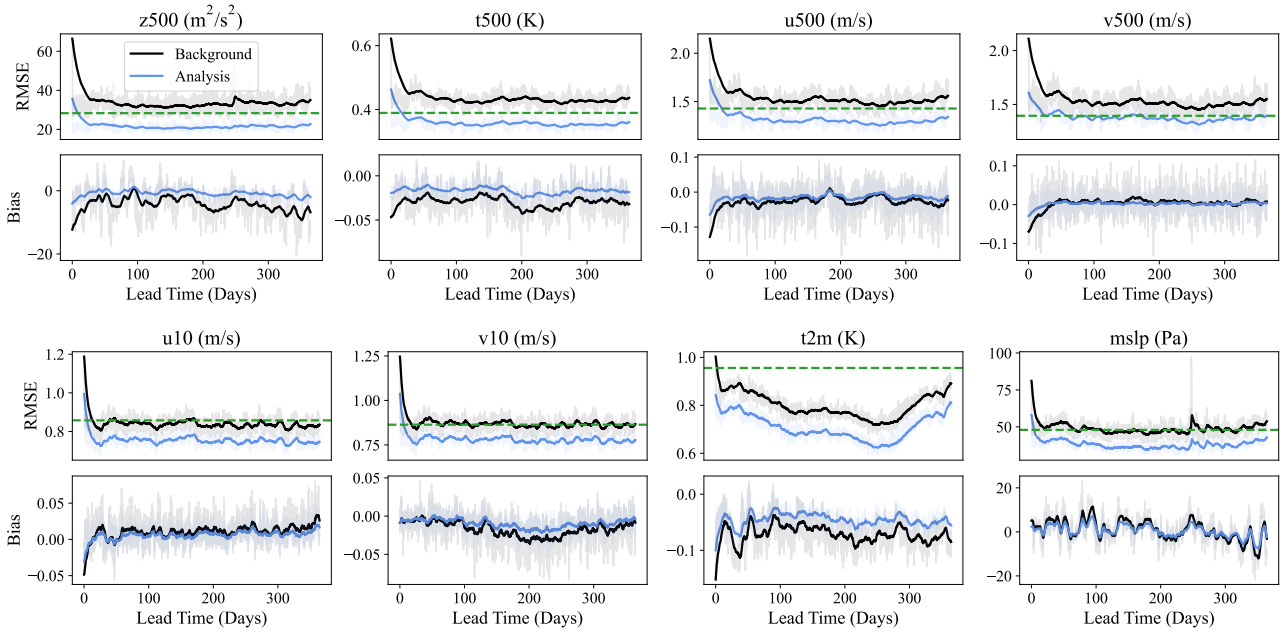

Figure 2. **Cyclic forecasting results of FengWu-4DVar.** The x-axis in each sub-figure represents lead time, at a 6-hour interval over a one-year lead time. The y-axis represents RMSE and Bias. The blue lines correspond to the analysis fields ($\mathbf{x}^a$ in Algorithm 1); the black lines correspond to the background fields ($\mathbf{x}^b$ in Algorithm 1); the green dotted lines correspond to the average RMSE of the IFS six-hour forecast fields.

Algorithm 1 Cyclic Forecasting with FengWu-4DVar

input Prior estimate of the initial state $\mathbf{x}_0^b$, AI forecasting model $\mathcal{M}^{ml}$, observations, background covariance matrix, observation covariance matrices, window size T, total steps L

1: $t \leftarrow 0$ {Initialize the time stamp}
2: **for** *step* from 0 to L **do**
3: $\mathbf{u}_t^a \leftarrow \arg\min_{\mathbf{u}} \tilde{J}\left(\mathbf{u} \middle| \mathbf{x}_t^b, \{\mathbf{y}_\tau\}_{\tau=t}^{t+T-1}\right)$ {Solve 4DVar to obtain the analysis field control variable.}
4: $\mathbf{x}_t^a = \mathbf{U}\mathbf{u}_t^a + \mathbf{x}^b$ {Recover the analysis field.}
5: $\mathbf{x}_{t+T}^b \leftarrow \mathcal{M}_{t\rightarrow t+T}^{ml}(\mathbf{x}_t^a)$ {A forecast is made to obtain the background field for the next time step. The lead time is equal to the assimilation window size.}
6: $t \leftarrow t + T$
7: **end for**

model because it is a classic AI weather forecasting model known for its outstanding forecasting capabilities, extending beyond ten days. We simulate five atmospheric variables (each with 13 pressure levels) and four surface variables, resulting in a total of 69 predictands. In this paper, the atmospheric variables are geopotential (z), specific humidity (q), zonal component of wind (u), meridional component of wind (v), and air temperature (t); the 13 sub-variables at different vertical levels are presented by abbreviating their short name and pressure levels (e.g., z500 denotes the geopotential height at a pressure level of 500 hPa). The four surface variables are 2-meter temperature (t2m), 10-meter u wind component (u10), 10-meter v wind component (v10), and mean sea level pressure (mslp). The spatial resolution we test is 128×256. We have trained two models using ERA5 dataset of year 1979-2015, including a 1-hour forecasting model $\mathcal{M}_1$, which is embedded into the 4DVar algorithm for representing flow dependencies, and a 6-hour forecasting model $\mathcal{M}_6$, which is employed for making forecasts.

Observation Setup In this study, all observations are simulated observations generated from the ERA5 reanalysis dataset (Hersbach et al., 2020). Two modifications are made to the reanalysis fields to make them as close as possible to the real-world observations. First, we introduce a random mask into the reanalysis field and fix the random mask at different time steps to simulate the sparse distribution of observation stations in real scenarios. Additionally, we add Gaussian noise to the reanalysis field to simulate measurement errors at observation stations. Unless stated otherwise, the mask proportion in our experiments is 15%, indicating that only 15% of the locations have observations. The standard deviation of observation noise is 0.001 times the standard deviation of the variable distribution.

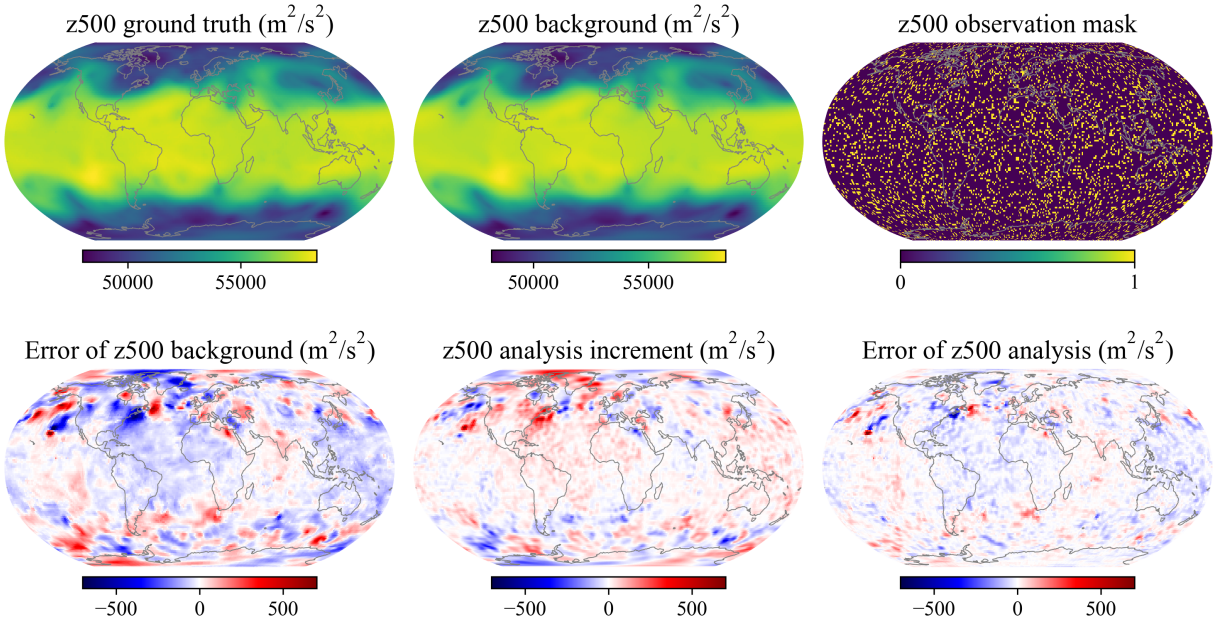

Figure 3. **Visualization of z500 at time 2018-01-01 00:00.** The ERA5 ground truth, the background field and the observation mask are demonstrated in the first row; the error of the background field(background field minus ground truth), the analysis increment (analysis field minus background field) and the error of the analysis field (analysis field minus ground truth) are shown in the second row. In the image of the observation mask, a pixel with a value of 1 indicates that there is an observation at that location, and a pixel with a value of 0 indicates an absence of any observation at that location.

Cyclic Forecasting Setup The initial state for starting our cyclic forecasting is obtained from the ERA5 dataset. In our experiments, FengWu-4DVar is initiated from 00:00 on January 1, 2018. To obtain the background field $\mathbf{x}_0^b$, we start from the ERA5 reanalysis field at 00:00 on December 30, 2017, integrate it using the 6-hour forecasting model for eight steps, and use the resulting fields to start the cyclic forecasting. We run FengWu-4DVar for one year, concluding its operation at 23:00 on December 31, 2018.

4.2. Analysis Field Evaluation

Figure 2 demonstrates the cyclic forecasting results of FengWu-4DVar. Four atmospheric variables at geopotential height 500 hPa and four surface variables are reported on two metrics (RMSE and Bias), which we let readers refer to the Appendix for detailed definitions (Rasp et al., 2020). It can be found that our AI-embedded 4DVar algorithm is capable of increasing the quality of the initial field. Take z500 as an example: the RMSE of z500 at the initial moment is over $60\,\mathrm{m}^2/\mathrm{s}^2$; after one step of assimilation, the RMSE of the background field drops below $40\,\mathrm{m}^2/\mathrm{s}^2$. The error remains stable after convergence, proving that our cyclic forecasting can operate stably in the long term. Furthermore, the RMSE of the analysis fields is smaller than the average RMSE of the IFS six-hour forecast fields on most variables, indicating that the analysis fields generated by FengWu-4DVar are accurate. The Bias indicator of the analysis fields is also improved compared with the background fields across all variables, showing that our AI-embedded 4DVar algorithm is capable of mitigating model bias in AI forecasting models.

In Figure 3, we visualize the assimilation results at time 2018-01-01 00:00, when the background error is the largest, corresponding to the most challenging case for data assimilation algorithms. It can be seen that the pattern of the z500 analysis increment resembles that of the background error, indicating that our AI-embedded 4DVar is capable of correcting the background error. Moreover, by comparing the error of the analysis fields and the background fields, it can be found that the error is significantly reduced after assimilation, which further validates the effectiveness of the assimilation process.

4.3. Evaluation Under Different Experimental Settings

In this section, we conduct experiments under different experimental settings to test the robustness of FengWu-4DVar.

Effect of Initial States We choose different initial states, corresponding to $\mathbf{x}_0^b$ in Algorithm 1, to test the performance of the cyclic forecasting framework under worse or better initial conditions. To achieve this, we select reanalysis fields

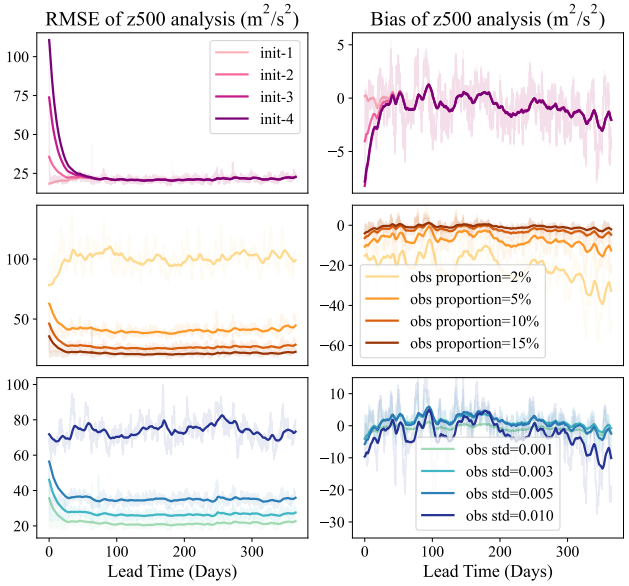

Figure 4. **Evaluation under different experimental settings.** The RMSE and Bias of the z500 analysis field are reported for experiments with (1) different initial states (in the first row), (2) different observation proportions (in the second row) and (3) different observation noise levels (in the last row). We let readers refer to the main text for the meanings of the labels.

from different timestamps and integrate them over different time steps. Specifically, we integrate the reanalysis field from 6 hours prior for one step, from 2 days prior for 8 steps, from 4 days prior for 16 steps, and from 6 days prior for 24 steps, and use these resulting states as initial conditions for assimilation experiments. These experiments are labeled as "init-1", "init-2", "init-3", and "init-4", with the original version of the experiment corresponding to "init-2".

The RMSE and Bias results on z500 variable are reported in the first row of Figure 4. It is shown that regardless of the magnitude of the initial state error, as we iterate the cyclic forecasting for around 20 days, our data assimilation framework consistently reduces the error to the same level as the original experiment setting. This indicates that our AI-embedded 4DVar algorithm is robust to the initial states.

Effect of Observation Proportions In this experiment, we reduce the observation proportion from 15% to 5% and evaluate the cyclic forecasting results. As shown in Figure 4, FengWu-4DVar adapt well to different observation proportions: when the observation proportion is only 5%, the error of z500 analysis field still converges. The RMSE after convergence is about $40\,\mathrm{m}^2/\mathrm{s}^2$, larger than the case where the observation proportion is 15%, but it is still a small value, equivalent to the 18-hour forecast error of IFS.

Effect of Observation Noise Levels This experiment evaluates the impact of observation noise intensity. We make the observations noisier by increasing the standard deviation of the observation noise from 0.001 to 0.01. The results are shown in the last row of Figure 4. It can be found that increasing the observation noise intensity has a negative impact on the assimilation results, with the RMSE of z500 rising from around $20\,\mathrm{m}^2/\mathrm{s}^2$ to over $70\,\mathrm{m}^2/\mathrm{s}^2$. Despite this, our FengWu-4DVar framework can still work stably in this situation and will not crash.

4.4. Comparison with the 3DVar Baseline

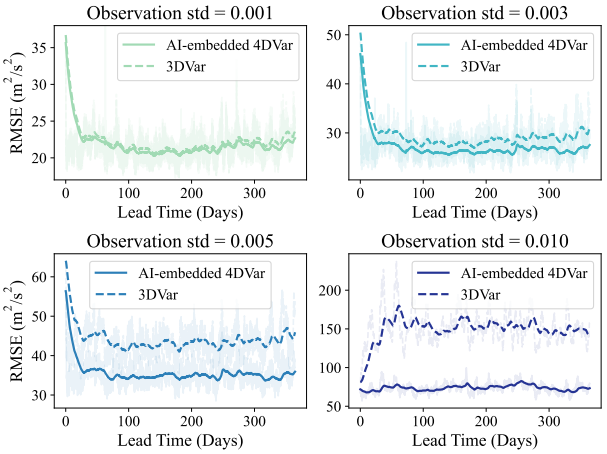

Figure 5. **Comparison with the 3DVar baseline.** The RMSE of the z500 analysis field is reported for AI-embedded 4DVar and 3DVar in experiments with different observation noise. The observation proportion is fixed to 15% across all the experiments.

In this section, we compare the results of AI-embedded 4DVar with a traditional three-dimensional variational assimilation (3DVar) baseline. In the 3DVar algorithm, only observations at the same time as the background fields are assimilated and no flow-dependencies are taken into consideration. Its objective function is shown below:

$$J(\mathbf{x}_0) = \frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}^b\right)^{\mathrm{T}} \mathbf{B}^{-1}\left(\mathbf{x}_0 - \mathbf{x}^b\right) + \frac{1}{2}\left(\mathcal{H}\left(\mathbf{x}_0\right) - \mathbf{y}_0\right)^{\mathrm{T}} \mathbf{R}_0^{-1}\left(\mathcal{H}\left(\mathbf{x}_0\right) - \mathbf{y}_0\right) \tag{6}$$

As demonstrated in Figure 5, our AI-embedded 4DVar algorithm consistently outperforms the 3DVar algorithm in terms of assimilation accuracy. As the observation becomes noisier, the accuracy gain of AI-embedded 4DVar gets greater. This is because when the observation quality deteriorates, AI-embedded 4DVar can use observations in the future to compensate for the quality defects. This comparison demonstrates the superiority of AI-embedded 4DVar over 3DVar.

4.5. Ablation Study

Additional experiments have been conducted in this section to demonstrate the effectiveness of our proposed AI-embedded 4DVar algorithm.

Effect of Horizontal Correlation Two experiments are conducted to evaluate the effect of our proposed horizontal correlation approximation stragegy. In the first experiment, horizontal correlation is considered, but we approximate the horizontal correlation using ordinary convolutions instead of spherical convolutions. In the second experiment, the horizontal correlation is removed and the background covariance matrix is assumed to be diagonal, as done in most previous AI data assimilation works.

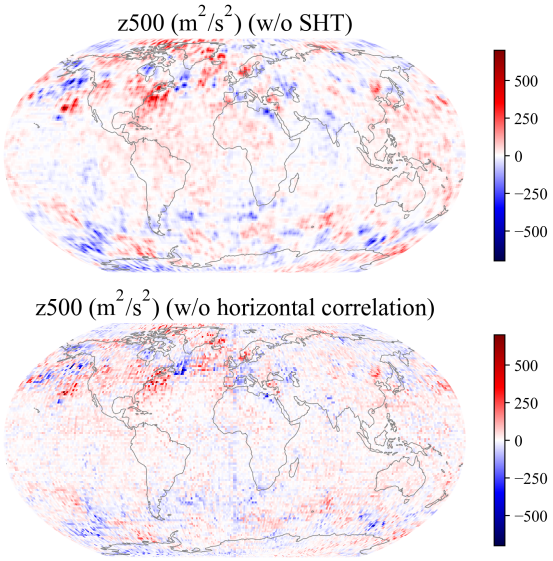

Figure 6. **Effect of horizontal correlation.** The analysis increment of z500 at time 2018-01-01 00:00 for two control experiments are demonstrated. In the first panel, the ordinary convolution is employed for approximating the horizontal correlation; in the second one, the horizontal correlation is removed. The observation proportion is set to 15% and the standard deviation is set to 0.001 in both experiments.

The analysis increments of these two experiments are shown in Figure 6. It can be found that when the ordinary convolution is used, the patterns of the analysis increment appear more fragmented compared with the results of applying the spherical convolution (as shown in the middle bottom panel of Figure 3), and the situation in polar regions is more serious. This is because when spherical convolution is used, the nonlinear equirectangular projection (ERP), which makes the actual distance between adjacent grid points smaller as it gets closer to the pole, is taken into account; whereas after switching to the ordinary convolution, the spherical field is treated as an ordinary two-dimensional image and the nonlinear transformation is not considered, which leads to worse assimilation results. When the horizontal correlation is removed, the analysis increment patterns appear even more discontinuous and most of the correction increments are concentrated at or near the observation points. These two experiments highlight the significance of our proposed SHT-based horizontal correlation approximation strategy.

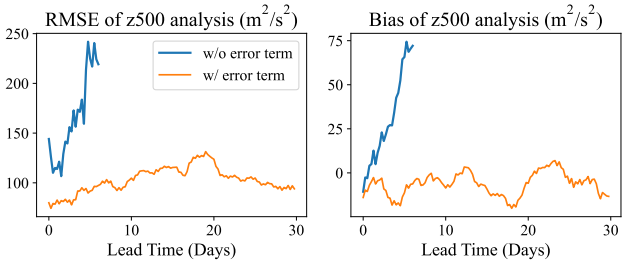

Figure 7. **Effect of the model error term.** The RMSE and Bias of the z500 analysis field are reported in both scenarios: one with the model error term (for a month) and another without it (for a week). The observation proportion is set to 15% and the standard deviation is set to 0.001 in both experiments.

Effect of the Model Error Term In this experiment, we remove the model error term in the objective function and demonstrate the assimilation results. As shown in Figure 7, after removing the model error term in the objective function, the RMSE of z500 rises to above $200\,\mathrm{m}^2/\mathrm{s}^2$ after a week of cyclic forecasting. This is because the 4DVar algorithm learns too much from observations at future moments, which should have been given less weight during assimilation. This ablation study indicates that the model error term plays a crucial role in the effective assimilation of AI-embedded 4DVar.

4.6. Computational Cost

FengWu-4DVar is implemented using auto-differentiation, and no additional neural network training is required once the forecasting model has been trained. The primary computational expenses of the data assimilation algorithm arise from the calculations involving auto-differentiation and the updates to the analysis fields using the L-BFGS optimization algorithm implemented by PyTorch. In our experiments, both auto-differentiation and gradient optimization updates are carried out on one GPU card of NVIDIA A100, with an average runtime of 29.3 seconds for assimilating over 300 thousand observations within a 6-hour assimilation window. As a comparison, realizing the traditional 4DVar algorithm on the 1° flow dependency resolution costs about 50 minutes on 256 processors of the PI-SUGON high-performance computer. This proves that our AI-embedded 4DVar algorithm achieves significant efficiency gain.

5. Conclusion

In this paper, we propose an AI-embedded 4DVar data assimilation algorithm, which consists of three components to deal with the challenge of building an AI data assimilation algorithm on a real-world weather forecasting system. By coupling this algorithm with the FengWu forecasting model, we build a self-contained data-driven global weather forecasting framework, FengWu-4DVar. This framework is evaluated with the forecasting model at 1.4° resolution and the ERA5 simulation observations. Under different observational settings and different initial conditions, FengWu-4DVar is capable of generating reasonable analysis fields and achieving stable and efficient cyclic assimilation and forecasting for at least one year, and the error of the analysis fields is smaller than both the 3DVar algorithm and the 6-hour forecast error of IFS. In addition, the computational efficiency of our algorithm greatly exceeds that of the traditional 4DVar algorithm.

We admit certain limitations of our current work. First, our experiments are conducted on simulated observations based on ERA5 and the effectiveness of our framework on real-world observations is yet to be verified. Second, we have not performed an end-to-end comparison between our proposed AI-embedded 4DVar algorithm and an operational 4DVar algorithm. We will address these limitations in future work.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (NO.U2242210). This work is also supported by Shanghai Artificial Intelligence Laboratory.

We gratefully acknowledge the ERA5 dataset provided by the European Centre for Medium-Range Weather Forecasts (ECMWF), whose efforts in data collection, archiving, and dissemination made this study possible.

We appreciate the Research Support, IT, and Infrastructure team at the Shanghai AI Laboratory for their provision of computational resources and network support.

We would also like to express our gratitude to Prof. Juanjuan Liu from the University of Chinese Academy of Science for her assistance and insightful discussions during the course of this research. Additionally, we thank Wanghan Xu from the Shanghai AI Laboratory for his help in conducting experiments for the rebuttal. Their contributions have significantly enhanced the quality of this work.

Impact Statement

This paper presents work whose goal is to advance the field of weather forecasting. Improving the accuracy and efficiency of weather forecasts can bring significant social benefits to various industries. First, enhanced weather forecasts can contribute to public safety by providing more reliable information for disaster preparedness and response. Accurate forecasts help authorities and communities make timely decisions, evacuate vulnerable areas and allocate resources effectively when faced with severe weather events such as hurricanes, floods or storms.

In the agricultural sector, precise weather forecasts are crucial for farmers to optimize planting and harvesting schedules, manage irrigation efficiently, and mitigate the impact of extreme weather conditions on crops. This leads to increased agricultural productivity, better resource management, and ultimately, food security.

Additionally, industries such as transportation and logistics rely heavily on accurate weather forecasts for planning and operational efficiency. Airlines, shipping lines and ground transportation services can optimize routes, schedules and fuel consumption, reducing costs and improving overall reliability.

In the energy sector, particularly in renewable energy production, precise weather predictions are essential for optimizing the output of solar and wind farms. This allows for better integration of renewable energy sources into the power grid, contributing to the transition to sustainable and environmentally friendly energy systems.

Overall, the potential social benefits of this work extend to public safety, food security, economic productivity, and sustainable resource management.

References

Bannister, R. N. A review of forecast error covariance statistics in atmospheric variational data assimilation. i: Characteristics and measurements of forecast error covariances. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 134(637):1951–1970, 2008.

Barker, D. M., Huang, W., Guo, Y.-R., Bourgeois, A., and Xiao, Q. A three-dimensional variational data assimilation system for mm5: Implementation and initial results. *Monthly Weather Review*, 132(4):897–914, 2004.

Bauer, P., Thorpe, A., and Brunet, G. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., and Tian, Q. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

Bonev, B., Kurth, T., Hundt, C., Pathak, J., Baust, M., Kashinath, K., and Anandkumar, A. Spherical fourier

neural operators: Learning stable dynamics on the sphere. *arXiv preprint arXiv:2306.03838*, 2023.

Chen, K., Han, T., Gong, J., Bai, L., Ling, F., Luo, J.-J., Chen, X., Ma, L., Zhang, T., Su, R., et al. Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. *arXiv preprint arXiv:2304.02948*, 2023a.

Chen, L., Zhong, X., Zhang, F., Cheng, Y., Xu, Y., Qi, Y., and Li, H. Fuxi: A cascade machine learning forecasting system for 15-day global weather forecast. *arXiv preprint arXiv:2306.12873*, 2023b.

Descombes, G., Auligné, T., Vandenberghe, F., Barker, D., and Barre, J. Generalized background error covariance matrix model (gen_be v2. 0). *Geoscientific Model Development*, 8(3):669–696, 2015.

Dong, R., Leng, H., Zhao, J., Song, J., and Liang, S. A framework for four-dimensional variational data assimilation based on machine learning. *Entropy*, 24(2):264, 2022.

Donner, L. J., Wyman, B. L., Hemler, R. S., Horowitz, L. W., Ming, Y., Zhao, M., Golaz, J.-C., Ginoux, P., Lin, S.-J., Schwarzkopf, M. D., et al. The dynamical core, physical parameterizations, and basic simulation characteristics of the atmospheric component am3 of the gfdl global coupled model cm3. *Journal of Climate*, 24(13):3484–3519, 2011.

Driscoll, J. R. and Healy, D. M. Computing fourier transforms and convolutions on the 2-sphere. *Advances in applied mathematics*, 15(2):202–250, 1994.

Fablet, R., Chapron, B., Drumetz, L., Mémin, E., Pannekoucke, O., and Rousseau, F. Learning variational data assimilation models and solvers. *Journal of Advances in Modeling Earth Systems*, 13(10):e2021MS002572, 2021.

Fisher, M. Background error covariance modelling. In *Seminar on Recent Development in Data Assimilation for Atmosphere and Ocean*, pp. 45–63. Shinfield Park, Reading, 2003.

Frerix, T., Kochkov, D., Smith, J., Cremers, D., Brenner, M., and Hoyer, S. Variational data assimilation with a learned inverse observation operator. In *International Conference on Machine Learning*, pp. 3449–3458. PMLR, 2021.

Geer, A. J. Learning earth system models from observations: machine learning or data assimilation? *Philosophical Transactions of the Royal Society A*, 379(2194):20200089, 2021.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., et al. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730): 1999–2049, 2020.

Hourdin, F., Mauritsen, T., Gettelman, A., Golaz, J.-C., Balaji, V., Duan, Q., Folini, D., Ji, D., Klocke, D., Qian, Y., et al. The art and science of climate model tuning. *Bulletin of the American Meteorological Society*, 98(3): 589–602, 2017.

Howes, K., Fowler, A. M., and Lawless, A. Accounting for model error in strong-constraint 4d-var data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143(704):1227–1240, 2017.

Jorge, N. and Stephen, J. W. *Numerical optimization*. Springer, 2006.

Kalnay, E. *Atmospheric modeling, data assimilation and predictability*. Cambridge university press, 2003.

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Pritzel, A., Ravuri, S., Ewalds, T., Alet, F., Eaton-Rosen, Z., et al. Graphcast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*, 2022.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

Rabier, F. and Liu, Z. Variational data assimilation: theory and overview. In *Proc. ECMWF Seminar on Recent Developments in Data Assimilation for Atmosphere and Ocean, Reading, UK, September 8–12*, pp. 29–43, 2003.

Rabier, F., Thépaut, J.-N., and Courtier, P. Extended assimilation and forecast experiments with a four-dimensional variational assimilation system. *Quarterly Journal of the Royal Meteorological Society*, 124(550):1861–1887, 1998.

Rabier, F., Järvinen, H., Klinker, E., Mahfouf, J.-F., and Simmons, A. The ecmwf operational implementation of four-dimensional variational assimilation. i: Experimental results with simplified physics. *Quarterly Journal of the Royal Meteorological Society*, 126(564):1143–1170, 2000.

Rasp, S., Dueben, P. D., Scher, S., Weyn, J. A., Mouatadid, S., and Thuerey, N. Weatherbench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020.

Strogatz, S. H. *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering*. CRC press, 2018.

Trémolet, Y. Accounting for an imperfect model in 4d-var. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 132(621):2483–2504, 2006.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, X., Chen, G., Qian, G., Gao, P., Wei, X.-Y., Wang, Y., Tian, Y., and Gao, W. Large-scale multi-modal pre-trained models: A comprehensive survey. *Machine Intelligence Research*, 20(4):447–482, 2023.

Zhang, L., Liu, Y., Liu, Y., Gong, J., Lu, H., Jin, Z., Tian, W., Liu, G., Zhou, B., and Zhao, B. The operational global four-dimensional variational data assimilation system at the china meteorological administration. *Quarterly Journal of the Royal Meteorological Society*, 145(722): 1882–1896, 2019.

Zhao, Y., Zhang, J., and Zong, C. Transformer: A general framework from machine translation to others. *Machine Intelligence Research*, 20(4):514–538, 2023.

A. Constructing the Adjoint Model Through Auto-differentiation

Assuming that $\mathcal{M}:\mathbb{R}^d\to\mathbb{R}^d$ is a forecasting model (the physical field flattened into a one-dimensional vector), the adjoint model of $\mathcal{M}$ at state $\mathbf{x}_0$ is defined as $\left(\frac{\partial\mathcal{M}(\mathbf{x})}{\partial\mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_0}\right)^{\mathrm{T}}$. It is a $d\times d$ matrix.

Suppose $\mathbf{y}\in\mathbb{R}^{d\times d}$ is another arbitrary d-dimensional vector. Then we can build the following "neural network", as shown in Figure 8. It only consists of two steps. In the first step, $\mathbf{x}_0$ is fed into the forecasting model and the predicted state $\mathbf{x}_1$ is obtained. In the second step, we do a dot product between $\mathbf{x}_1$ and $\mathbf{y}$ and produce a scalar z as the output.

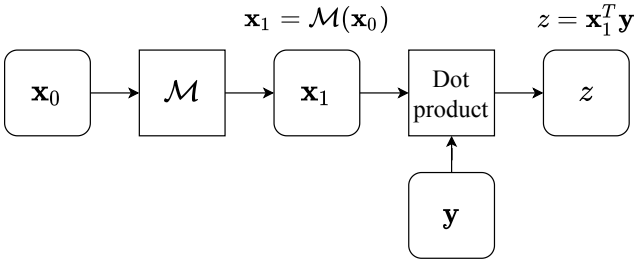

Figure 8. **"Neural Network" for calculating the adjoint model.**

Through auto-differentiation, we can obtain the gradient at node $\mathbf{x}_0$, that is $\frac{\partial z}{\partial\mathbf{x}_0}$. On the other hand, we can do the computational graph manually, and find out what the gradient stands for:

$$\begin{aligned}\frac{\partial z}{\partial\mathbf{x}_1}&=\mathbf{y}\\ \frac{\partial z}{\partial\mathbf{x}_0}&=\left(\frac{\partial\mathbf{x}_1}{\partial\mathbf{x}_0}\right)^T\frac{\partial z}{\partial\mathbf{x}_1}=\left(\frac{\partial\mathcal{M}(\mathbf{x})}{\partial\mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_0}\right)^T\mathbf{y}\end{aligned}\tag{7}$$

According to Equation 7, it can be found that the gradient at node $\mathbf{x}_0$ precisely represents the result of the adjoint model (defined at $\mathbf{x}_0$) acting on $\mathbf{y}$. Since both $\mathbf{x}_0$ and $\mathbf{y}$ are arbitrary, through this approach, we can calculate the results of the adjoint model, defined at any point, acting on any vectors. This concludes the proof that the adjoint of any differentiable forecasting model can be constructed through auto-differentiation.

B. Equivalence Between Two Optimization Methods

Define $g(\mathbf{x}_0,\mathbf{x}^b)=\frac{1}{2}(\mathbf{x}_0-\mathbf{x}^b)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}_0-\mathbf{x}^b)$, $f_i(\mathbf{x},\mathbf{y})=\frac{1}{2}(\mathcal{H}(\mathbf{x})-\mathbf{y})^{\mathrm{T}}\mathbf{R}_i^{-1}(\mathcal{H}(\mathbf{x})-\mathbf{y})$. Let $\mathcal{M}_{0\to1}=\mathcal{M}_{1\to2}=\cdots=\mathcal{M}_{T-2\to T-1}=\mathcal{M}$. Then the computational graph of calculating the 4DVar objective function can be constructed, as shown in Figure 9. For simplicity, Denote $\mathbf{H}_\tau=\frac{\partial\mathcal{H}(\mathbf{x})}{\partial\mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_\tau}$ and $\mathbf{M}_\tau=\frac{\partial\mathcal{M}(\mathbf{x})}{\partial\mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_\tau}=\frac{\partial\mathbf{x}_{\tau+1}}{\partial\mathbf{x}_\tau}$. We can simulate the computational graph back-propagation process and calculate the gradient sequentially:

$$\begin{aligned}\frac{\partial J}{\partial\mathbf{x}_{T-1}}&=\frac{\partial J_{R_{T-1}}}{\partial\mathbf{x}_{T-1}}\frac{\partial J}{\partial J_{R_{T-1}}}=\frac{\partial J_{R_{T-1}}}{\partial\mathbf{x}_{T-1}}=\mathbf{H}_{T-1}^{\mathrm{T}}\mathbf{R}_{T-1}^{-1}(\mathcal{H}(\mathbf{x}_{T-1})-\mathbf{y}_{T-1}),\\ \frac{\partial J}{\partial\mathbf{x}_{T-2}}&=\frac{\partial J_{R_{T-2}}}{\partial\mathbf{x}_{T-2}}\frac{\partial J}{\partial J_{R_{T-2}}}+\left(\frac{\partial\mathbf{x}_{T-1}}{\partial\mathbf{x}_{T-2}}\right)^{\mathrm{T}}\frac{\partial J}{\partial\mathbf{x}_{T-1}}=\frac{\partial J_{R_{T-2}}}{\partial\mathbf{x}_{T-2}}+\left(\frac{\partial\mathbf{x}_{T-1}}{\partial\mathbf{x}_{T-2}}\right)^{\mathrm{T}}\frac{\partial J}{\partial\mathbf{x}_{T-1}}\\ &=\mathbf{H}_{T-2}^{\mathrm{T}}\mathbf{R}_{T-2}^{-1}(\mathcal{H}(\mathbf{x}_{T-2})-\mathbf{y}_{T-2})+\mathbf{M}_{T-2}^{\mathrm{T}}\frac{\partial J}{\partial\mathbf{x}_{T-1}},\end{aligned}\tag{8}$$

Continuing in this manner, we can derive a general formula:

$$\begin{aligned}\frac{\partial J}{\partial\mathbf{x}_\tau}&=\frac{\partial J_{R_\tau}}{\partial\mathbf{x}_\tau}\frac{\partial L}{\partial J_{R_\tau}}+\left(\frac{\partial\tau_{k+1}}{\partial\mathbf{x}_\tau}\right)^{\mathrm{T}}\frac{\partial J}{\partial\mathbf{x}_{\tau+1}}=\frac{\partial J_{R_\tau}}{\partial\mathbf{x}_\tau}+\left(\frac{\partial\mathbf{x}_{\tau+1}}{\partial\mathbf{x}_\tau}\right)^{\mathrm{T}}\frac{\partial J}{\partial\mathbf{x}_{\tau+1}}\\ &=\mathbf{H}_\tau^{\mathrm{T}}\mathbf{R}_\tau^{-1}(\mathcal{H}(\mathbf{x}_\tau)-\mathbf{y}_\tau)+\mathbf{M}_\tau^{\mathrm{T}}\frac{\partial J}{\partial\mathbf{x}_{\tau+1}}.\end{aligned}\tag{9}$$

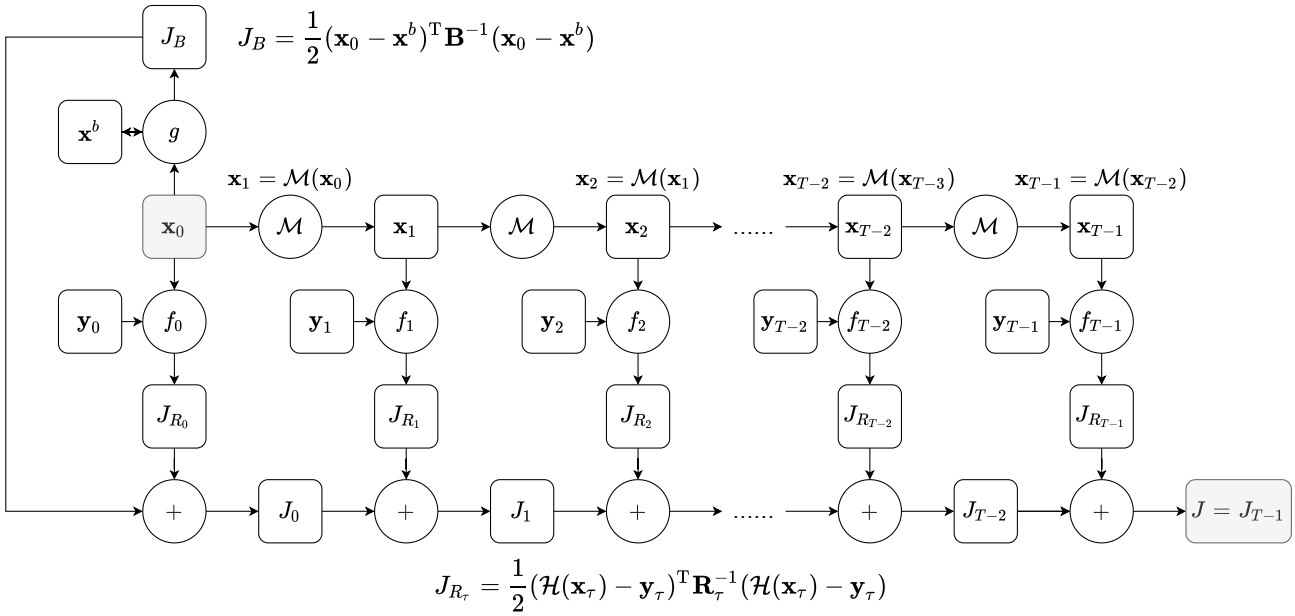

Figure 9. **"Neural Network" for calculating the 4DVar objective function.**

This corresponds to the gradient stored at the intermediate node $\mathbf{x}_\tau$, which is calculated through auto-differentiation.

Equation 9 holds for $1 \leq \tau \leq T-2$; when $\tau = 0$, an additional background term should be included:

$$\frac{\partial J}{\partial \mathbf{x}_0} = \frac{\partial J_{R_B}}{\partial \mathbf{x}_0}\frac{\partial J}{\partial J_B} + \frac{\partial J_{R_0}}{\partial \mathbf{x}_0}\frac{\partial J}{\partial J_{R_0}} + \left(\frac{\partial \mathbf{x}_1}{\partial \mathbf{x}_0}\right)^{\mathrm{T}}\frac{\partial J}{\partial \mathbf{x}_1} = \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_b) + \mathbf{H}_0^{\mathrm{T}}\mathbf{R}_0^{-1}(\mathcal{H}(\mathbf{x}_0) - \mathbf{y}_0) + \mathbf{M}_0^{\mathrm{T}}\frac{\partial L}{\partial \mathbf{x}_1}. \tag{10}$$

Up to this point, we have elucidated the mechanism through which auto-differentiation computes the gradient. By comparing this process with the methodology employed by traditional adjoint model-based methods for gradient calculation (Rabier & Liu, 2003), it becomes evident that these two processes are entirely identical.

C. Derivation of the Model Error Term in AI-embedded 4DVar

First, we assume that the error of the forecasting model follows a Gaussian distribution. It's worth noting that this assumption holds true only for linear models. For nonlinear models, the preservation of Gaussian distribution errors cannot be guaranteed during model integration. However, this does not prevent us from making such an approximation to make the problem manageable. Denote $\mathbf{Q}_\tau$ the error variance of the τ-step integration model, $\mathcal{M}_{0\to\tau}$ or $\mathcal{M}_\tau$, then $\mathbf{x}_\tau|\mathbf{x}_0 \sim \mathcal{N}(\mathcal{M}_\tau(\mathbf{x}_0), \mathbf{Q}_\tau)$. In practice, $\mathbf{Q}_\tau$ can also be estimated by computing the sampling statistics in a manner similar to the national meteorological center (NMC) method (Bannister, 2008). Since the observation at time τ follows the Gaussian distribution $\mathbf{y}_\tau|\mathbf{x}_\tau \sim \mathcal{N}(\mathbf{H}\mathbf{x}_\tau, \mathbf{R}_\tau)$, through the compound rule of Gaussian distributions, $\mathbf{y}_\tau|\mathbf{x}_0$ also follows the Gaussian distribution:

$$\mathbf{y}_\tau|\mathbf{x}_0 \sim \mathcal{N}(\mathcal{M}_\tau(\mathbf{x}_0), \mathbf{R}_\tau + \mathbf{H}\mathbf{Q}_\tau\mathbf{H}^{\mathrm{T}}) \tag{11}$$

According to Bayesian Theorem, we have

$$\arg\max_{\mathbf{x}_0} p(\mathbf{x}_0|\mathbf{y}_0, \cdots, \mathbf{y}_{T-1}) = \arg\max_{\mathbf{x}_0} \frac{p(\mathbf{y}_0, \cdots, \mathbf{y}_{T-1}|\mathbf{x}_0)p(\mathbf{x}_0)}{p(\mathbf{y}_0, \cdots, \mathbf{y}_{T-1})} = \arg\max_{\mathbf{x}_0} p(\mathbf{y}_0, \cdots, \mathbf{y}_{T-1}|\mathbf{x}_0)p(\mathbf{x}_0). \tag{12}$$

Since observations at different time steps are independent, the above formula can be simplified as follows:

$$
\begin{aligned}
&p(\mathbf{y}_0, \cdots, \mathbf{y}_{T-1}|\mathbf{x}_0)p(\mathbf{x}_0) = p(\mathbf{x}_0)\prod_{\tau=0}^{T-1} p(\mathbf{y}_\tau|\mathbf{x}_0) \\
=&C\exp\left(-\frac{1}{2}\left(\mathbf{x}_0-\mathbf{x}^b\right)^{\mathrm{T}}\mathbf{B}^{-1}\left(\mathbf{x}_0-\mathbf{x}^b\right)\right)\prod_{\tau=0}^{T-1}\exp\left(-\frac{1}{2}\left(\mathcal{M}_\tau(\mathbf{x}_0)-\mathbf{y}_\tau\right)^{\mathrm{T}}\left(\mathbf{R}_\tau+\mathbf{H}\mathbf{Q}_\tau\mathbf{H}^{\mathrm{T}}\right)^{-1}\left(\mathcal{M}_\tau(\mathbf{x}_0)-\mathbf{y}_\tau\right)\right),
\end{aligned}
\tag{13}
$$

where C is a constant. Taking negative logarithm of this likelihood yields the objective function,

$$
J(\mathbf{x}_0) = \frac{1}{2}\left(\mathbf{x}_0-\mathbf{x}^b\right)^{\mathrm{T}}\mathbf{B}^{-1}\left(\mathbf{x}_0-\mathbf{x}^b\right) + \frac{1}{2}\sum_{\tau=0}^{T-1}\left(\mathcal{M}_\tau(\mathbf{x}_0)-\mathbf{y}_\tau\right)^{\mathrm{T}}\left(\mathbf{R}_\tau+\mathbf{H}\mathbf{Q}_\tau\mathbf{H}^{\mathrm{T}}\right)^{-1}\left(\mathcal{M}_\tau(\mathbf{x}_0)-\mathbf{y}_\tau\right), \tag{14}
$$

which concludes the proof.

D. Evaluation Metrics

To evaluate the performance of our FengWu-4DVar framework, we regard the ERA5 dataset as the ground truth and compare the analysis fields sequence $\{\mathbf{x}^a_{lT}\}_{l=0}^L$ and the background fields sequence $\{\mathbf{x}^b_{lT}\}_{l=0}^L$ with it. The metrics we use are RMSE and Bias.

RMSE corresponds to the latitude-weighted root mean square error. It is a statistical metric widely used to assess the accuracy of a model's predictions across different latitudes. Denote $\hat{x}_{l,c,w,h}$ the predicted value of the l-th sample at channel c (It can either be the surface variable or the atmospheric variable at a certain pressure level.), and w and h represents the indices for each grid along the latitude and longitude indices. Denote $x_{l,c,w,h}$ the target value. Then the RMSE at channel c is defined as

$$
\mathrm{RMSE}(c) = \frac{1}{L}\sum_{l=1}^{L}\sqrt{\frac{1}{W\cdot H}\sum_{w=1}^{W}\sum_{h=1}^{H} W\cdot\frac{\cos\left(\alpha_{w,h}\right)}{\sum_{w'=1}^{W}\cos\left(\alpha_{w',h}\right)}\left(x_{l,c,w,h}-\hat{x}_{l,c,w,h}\right)^2}, \tag{15}
$$

where $\alpha_{w,h}$ is the latitude of point (w,h).

Bias corresponds to the latitude-weighted bias. It is widely used to assess the systematic bias of a model. Following the denotation above, the Bias at channel c is defined as

$$
\mathrm{Bias}(c) = \frac{1}{L}\sum_{l=1}^{L}\frac{1}{W\cdot H}\sum_{w=1}^{W}\sum_{h=1}^{H} W\cdot\frac{\cos\left(\alpha_{w,h}\right)}{\sum_{w'=1}^{W}\cos\left(\alpha_{w',h}\right)}\left(\hat{x}_{l,c,w,h}-x_{l,c,w,h}\right). \tag{16}
$$

E. Additional Results

Figure 10, 11 and 12 demonstrate the cyclic forecasting results of FengWu-4DVar under different observation proportions on 12 variables, with the observation standard deviation fixed to 0.001. Figure 13, 14 and 15 demonstrate the cyclic forecasting results of FengWu-4DVar under different observation standard deviations on 12 variables, with the observation proportions fixed to 15%.

F. Comparison with Traditional 4DVar

We also conducted experiments to compare the forecasting skills of the analysis fields generated by FengWu-4DVar with those of the IFS analysis fields. Using the analysis fields obtained from FengWu-4DVar, we performed forecasts with FengWu and calculated the RMSE (relative to ERA5 ground truth) at different lead times (up to 7 days). Similarly, we calculated the RMSE for the IFS forecast results and compared the errors. The results are presented in Figure 16. This comparison visually demonstrates the forecasting skills of the analysis fields generated by FengWu-4DVar. However, it does not establish that our algorithm is superior to the traditional 4DVar algorithm, as IFS utilizes real observational data for assimilation, whereas our system incorporates simulated observations.

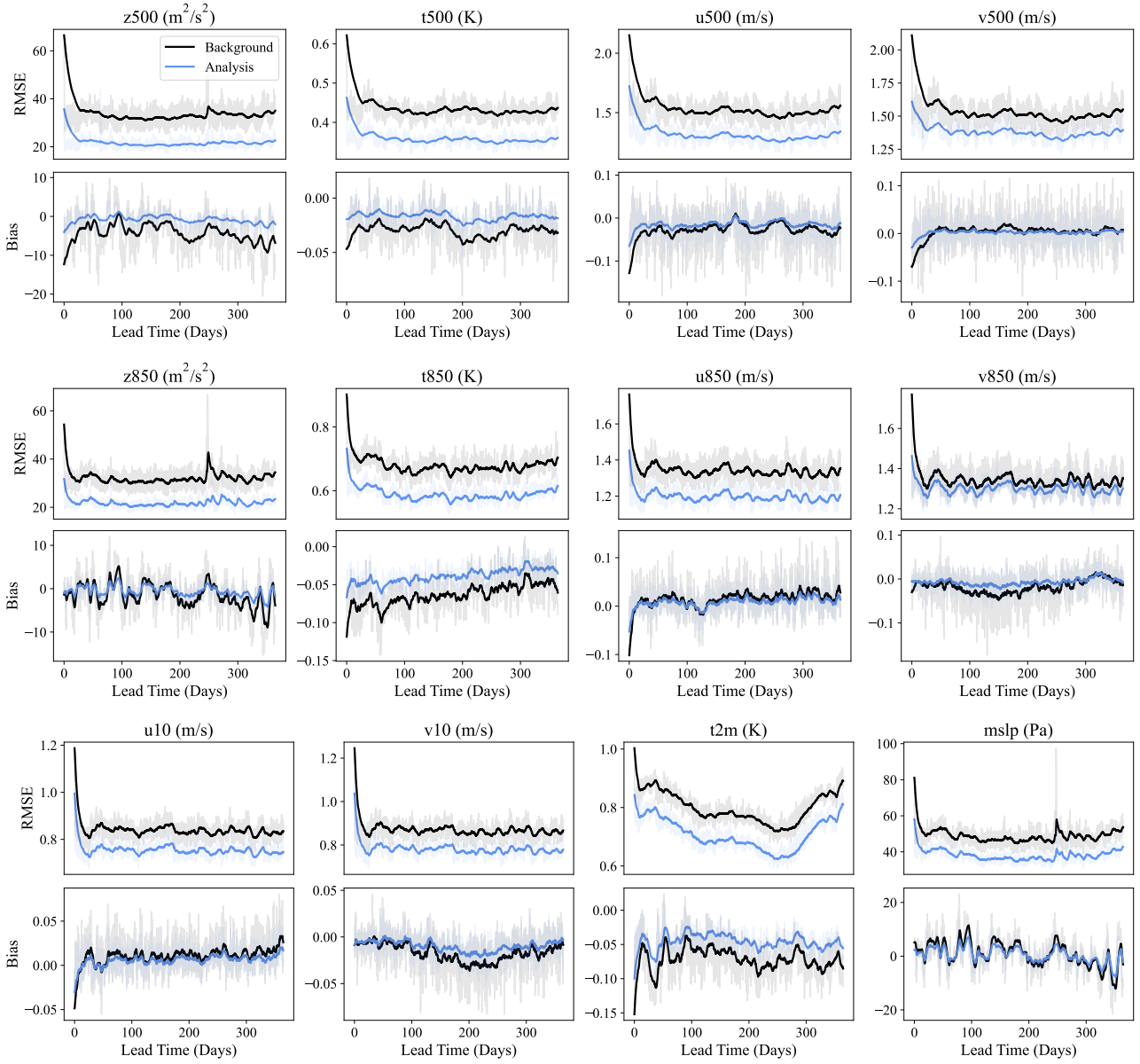

Figure 10. **Cyclic forecasting results of FengWu-4DVar with an observation proportion of 15% and an observation standard deviation of 0.001.** The x-axis in each sub-figure represents lead time, at a 6-hour interval over a one-year lead time. The y-axis represents RMSE and Bias.

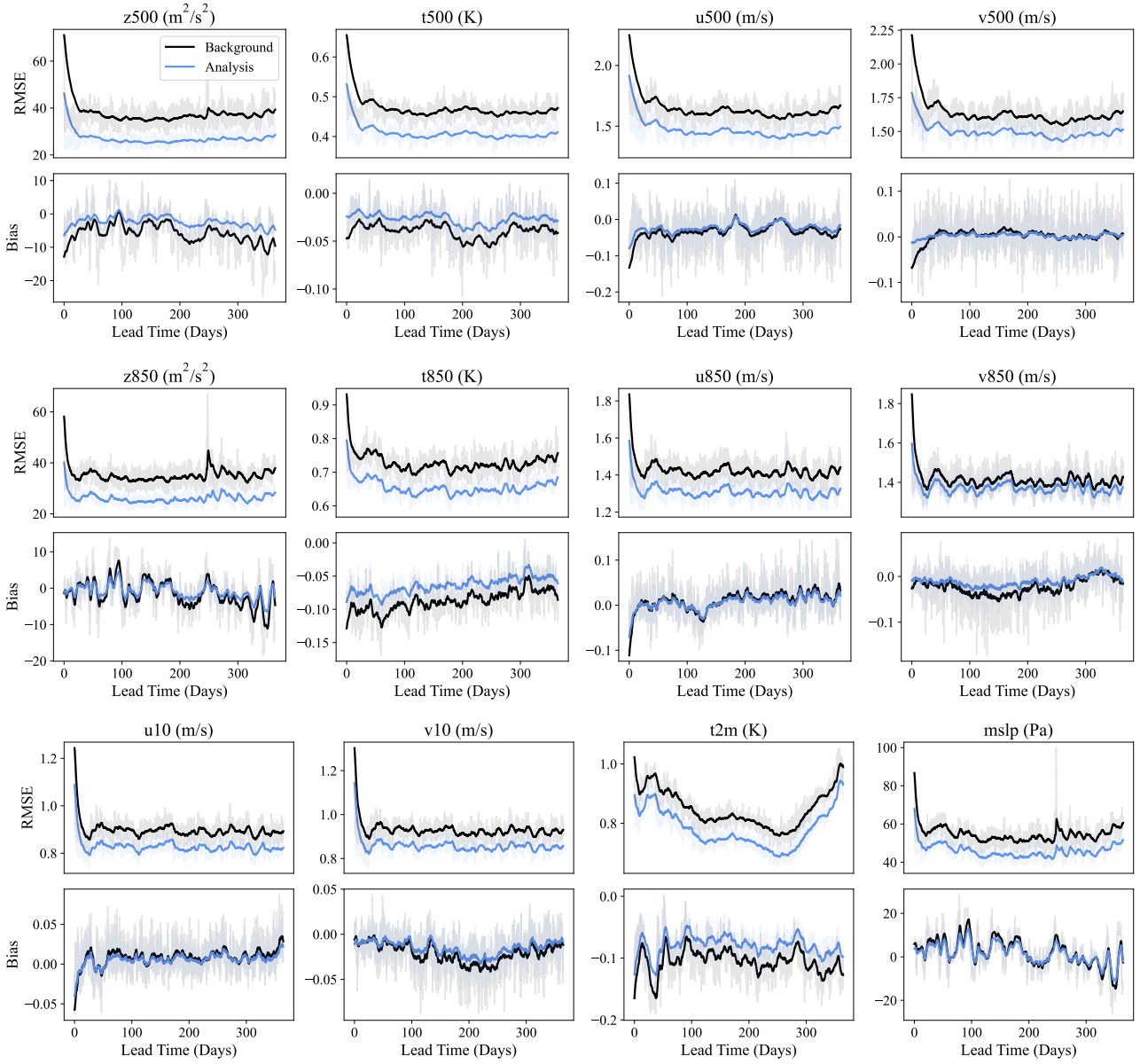

Figure 11. **Cyclic forecasting results of FengWu-4DVar with an observation proportion of 10% and an observation standard deviation of 0.001.** The x-axis in each sub-figure represents lead time, at a 6-hour interval over a one-year lead time. The y-axis represents RMSE and Bias.

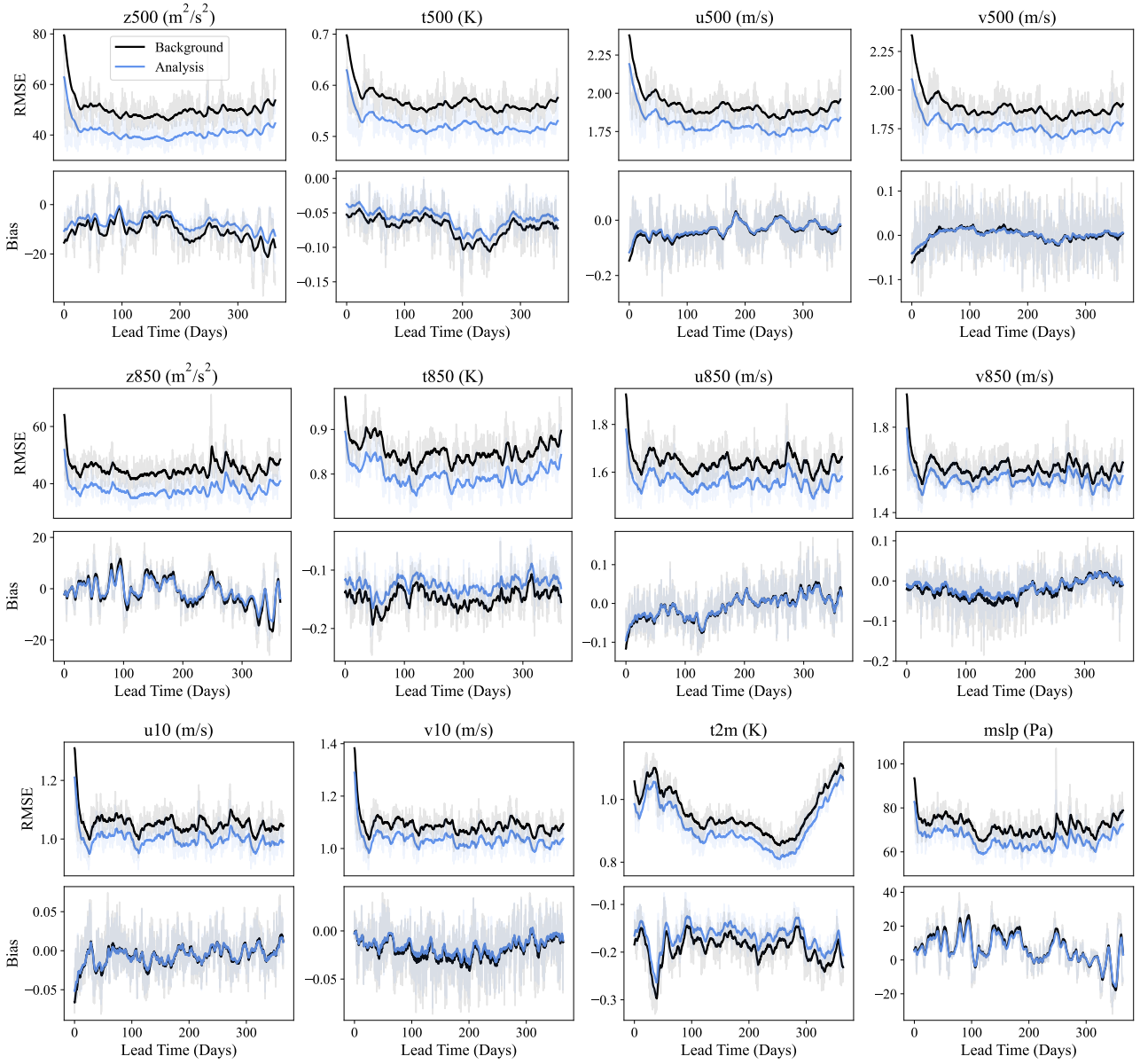

Figure 12. **Cyclic forecasting results of FengWu-4DVar with an observation proportion of 5% and an observation standard deviation of 0.001.** The x-axis in each sub-figure represents lead time, at a 6-hour interval over a one-year lead time. The y-axis represents RMSE and Bias.

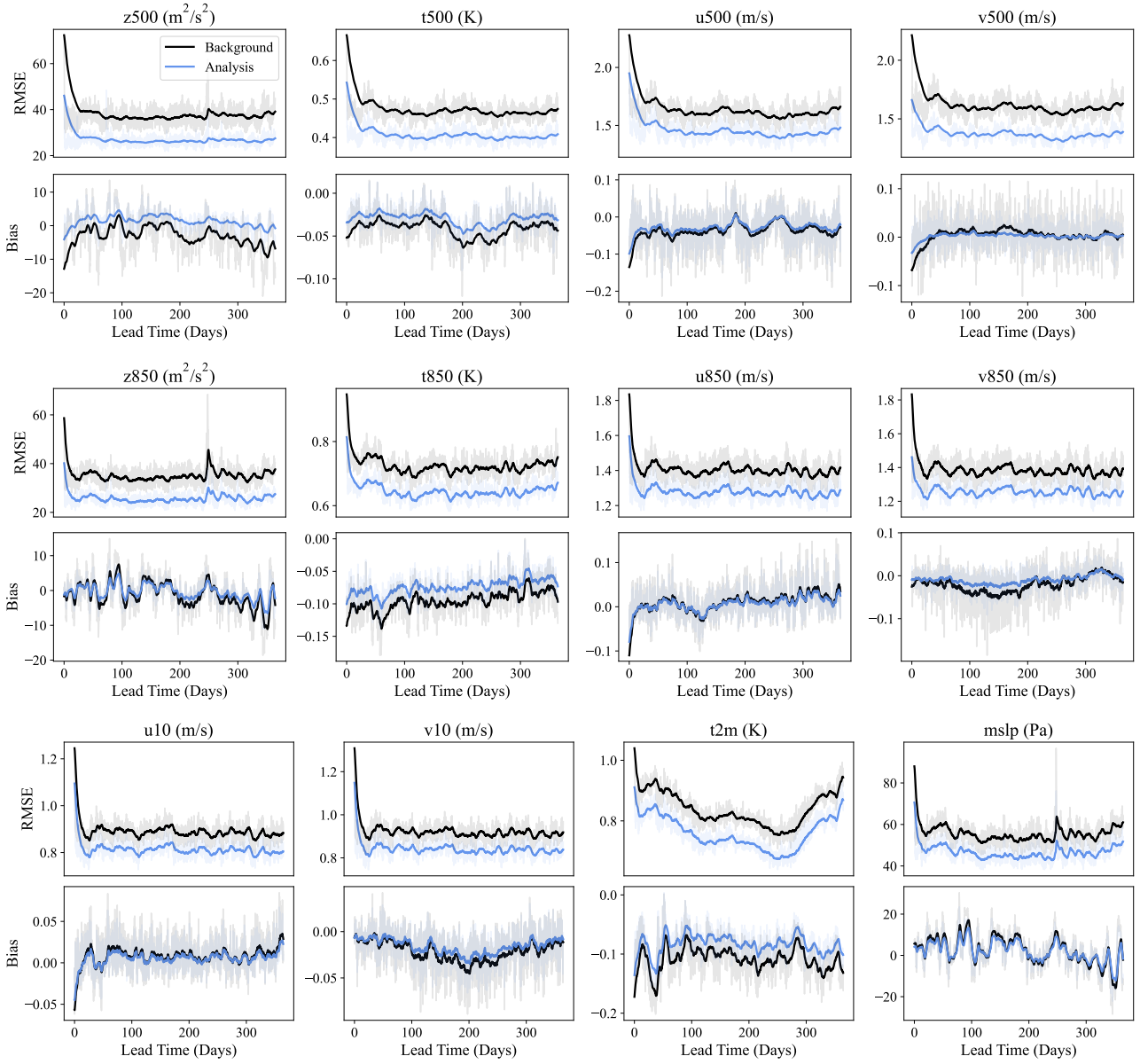

Figure 13. **Cyclic forecasting results of FengWu-4DVar with an observation proportion of 15% and an observation standard deviation of 0.003.** The x-axis in each sub-figure represents lead time, at a 6-hour interval over a one-year lead time. The y-axis represents RMSE and Bias.

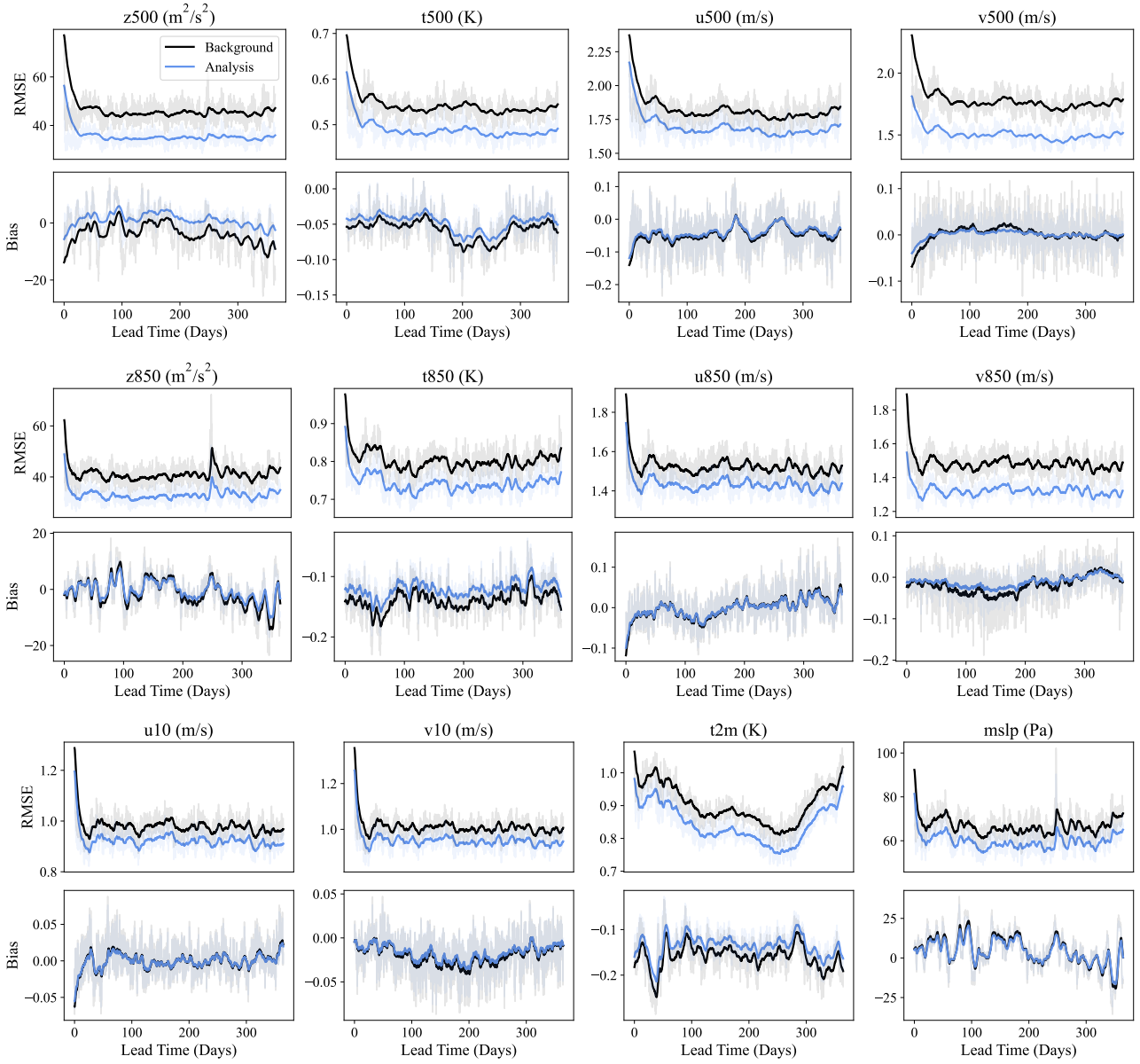

Figure 14. **Cyclic forecasting results of FengWu-4DVar with an observation proportion of 15% and an observation standard deviation of 0.005.** The x-axis in each sub-figure represents lead time, at a 6-hour interval over a one-year lead time. The y-axis represents RMSE and Bias.

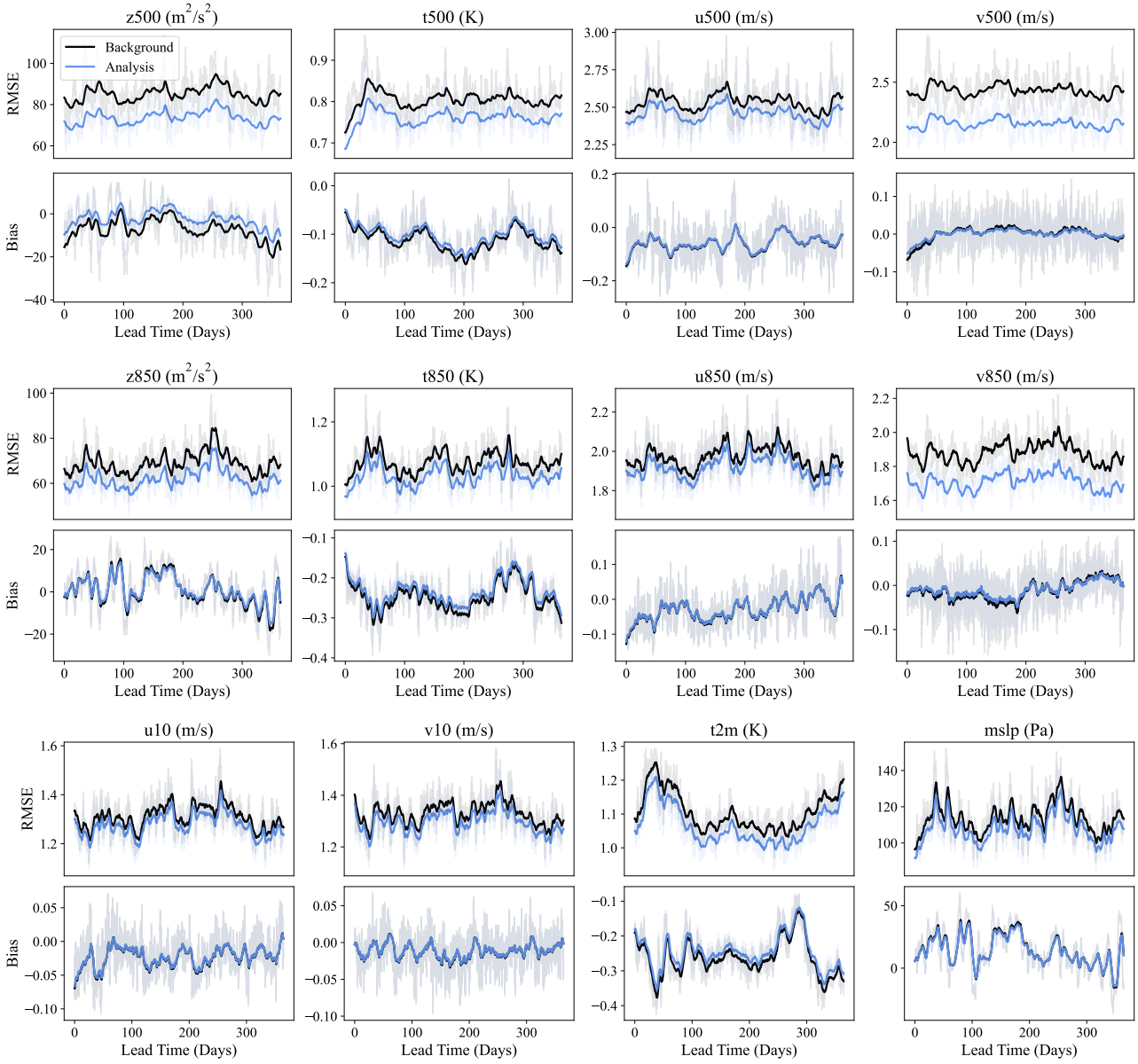

Figure 15. **Cyclic forecasting results of FengWu-4DVar with an observation proportion of 15% and an observation standard deviation of 0.01.** The x-axis in each sub-figure represents lead time, at a 6-hour interval over a one-year lead time. The y-axis represents RMSE and Bias.

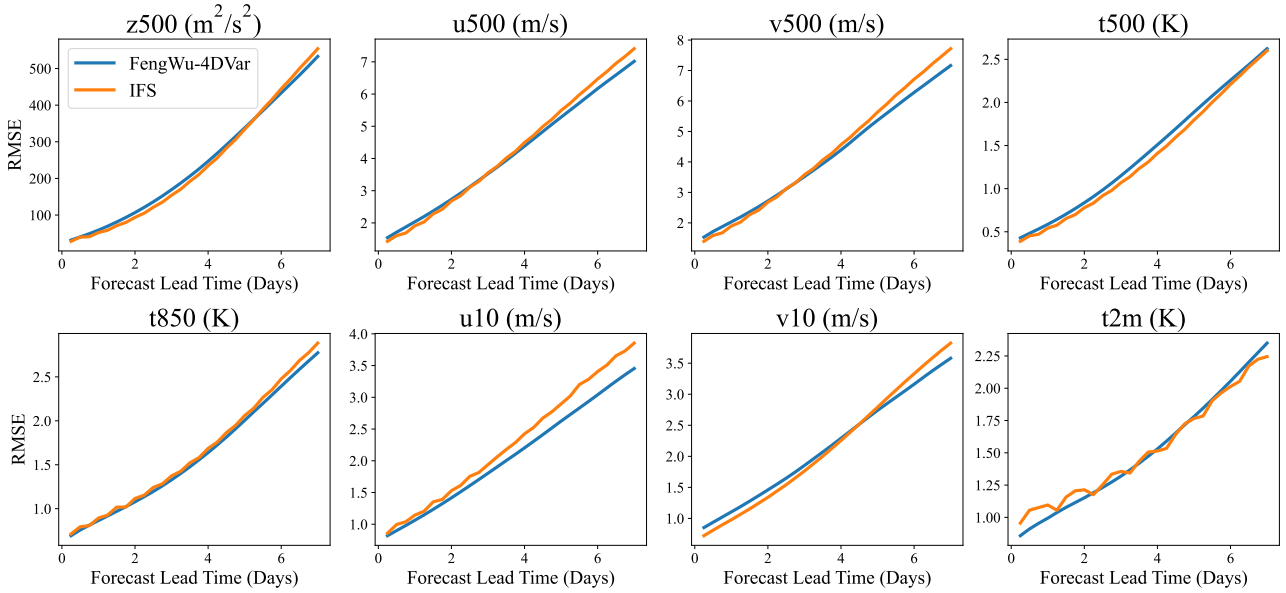

Figure 16. **Forecast skills of FengWu-4DVar's analysis fields and IFS's analysis fields.** The analysis fields of FengWu-4DVar are generated with an observation proportion of 15% and an observation standard deviation of 0.01. The x-axis in each sub-figure represents the lead time of forecasting, at a 6-hour interval over a one-week period. The y-axis indicates the RMSE, averaged over the analysis fields for the entire year.