A Four-Dimensional Variational Constrained Neural Network-based Data Assimilation Method

Wuxin Wang[1,2], Kaijun Ren[1,2], Boheng Duan[2], Junxing Zhu[2], Xiaoyong Li[2], Weicheng Ni[1,2], Jingze Lu[1,2]and Taikang Yuan[2]

[1]College of Computer Science and Technology, National University of Defense Technology, Changsha, China
[2]College of Meteorology and Oceanography, National University of Defense Technology, Changsha, China

Key Points:

- A physics-informed neural network trained without ground truths can provide accurate initial fields for numerical prediction.
- The system's kinetic features are embedded into the model through our four-dimensional variational form loss function.
- We show on Lorenz96 that the proposed method can be used directly for accurate data assimilation at a low computational cost.

Corresponding author: Kaijun Ren, `renkaijun@nudt.edu.cn`

Corresponding author: Boheng Duan, `bhduan@nudt.edu.cn`

Abstract

Advances in data assimilation (DA) methods and the increasing amount of observations have continuously improved the accuracy of initial fields in numerical weather prediction during the last decades. Meanwhile, in order to effectively utilize the rapidly increasing data, Earth scientists must further improve DA methods. Recent studies have introduced machine learning (ML) methods to assist the DA process. In this paper, we explore the potential of a four-dimensional variational (4DVar) constrained neural network (NN) method for accurate DA. Our NN is trained to approximate the solution of the variational problem, thereby avoiding the need for expensive online optimization when generating the initial fields. In the context that the full-field system truths are unavailable, our approach embeds the system's kinetic features described by a series of analysis fields into the NN through a 4DVar-form loss function. Numerical experiments on the Lorenz96 physical model demonstrate that our method can generate better initial fields than most traditional DA methods at a low computational cost, and is robust when assimilating observations with higher error outside of the distributions where it is trained. Furthermore, our NN-based DA model is effective against Lorenz96 physical models with larger variable numbers. Our approach exemplifies how ML methods can be leveraged to improve both the efficiency and accuracy of DA techniques.

Plain Language Summary

The use of machine learning (ML) to approximate mappings from data has made a significant impact on numerical weather prediction. In the data assimilation (DA) process, several recent studies have applied ML to accelerate or improve the accuracy of DA output. In this paper, we investigate the potential of employing physical constraints based on four-dimensional variational (4DVar) DA to further enhance the accuracy of an end-to-end ML-based DA model. Our objective is to determine whether the 4DVar-constrained ML model can perform the DA task more efficiently and produce comparable accuracy to the traditional DA methods. We trained our NN-based model without true values as labels and test it on the Lorenz96 physical model. Several experiments have been applied to verify that the 4DVar-constrained ML model can be used as a potential substitute for the DA process.

1 Introduction

Numerical weather prediction (NWP) is an initial-value problem, and the discrepancy between the initial field and the true state of Earth can lead to errors in NWP models. To address this issue, data assimilation (DA) techniques have been developed and applied to NWP, resulting in notable improvements in accuracy (Gustafsson et al., 2018). In particular, the development and operational use of three-dimensional and four-dimensional variational assimilation (3D/4DVar) methods (Courtier et al., 1994), the more recent development of ensemble DA approaches (Evensen et al., 2009), and other variational-ensemble hybrid methods have been significant milestones in NWP (Bocquet, 2016; Bannister, 2017). Most of the top operational centers for NWP and reanalysis use variations of these techniques (Hersbach et al., 2020; Compo et al., 2011; Clayton et al., 2013). In addition, the expansion of the amount and diversity of observations is also essential to NWP. In the future, increasing observations with a higher spatial and temporal resolution and greater accuracy (Gettelman et al., 2022) presents a tremendous opportunity to further enhance the quality of initial fields, while the challenge of extracting all relevant information using traditional methods is becoming more severe (Düben et al., 2021). Furthermore, the growing grid number of the numerical models also makes DA approaches increasingly computational cost in many realistic situations (Carrassi et al., 2018). Consequently, it necessitates Earth scientists to consider improving DA efficiency further (Huang et al., 2021).

The application of machine learning (ML) techniques (Goodfellow et al., 2016) to a variety of tasks, such as image recognition (Han et al., 2022), neural language processing

(Kenton & Toutanova, 2019), and video prediction (Oprea et al., 2020), has been widely reported. In the earth science domain, ML also offers a powerful toolkit to improve the computational efficiency of models and extract information from large amounts of data about Earth (Düben et al., 2021). Further, Bocquet (2023) and Cheng et al. (2023) highlight the potential of ML and DA for improving the accuracy and efficiency of models in Earth sciences. In addition, the synergy between ML and DA has been highlighted by Boukabara et al. (2020), while Bocquet, Brajard, et al. (2020) numerically demonstrated that ML and DA both act as coordinate descent minimization for the specified loss function. This has enabled the training of neural networks (NNs) to directly minimize a pixel-wise distance measure for a regression task. However, such pixel-by-pixel ground truth is unavailable for DA applications in NWP, making the direct comparison between the NN's output and the system truth impossible. As a result, most of these studies have trained NNs as approximators of the traditional DA methods to alleviate the computational burden associated with the NWP's initializing process. For example, Wu et al. (2021) used a multilayer perceptron (MLP) to learn the relationship between the observed data and the dynamic model solution, and learned to minimize the mean square error (MSE) between the MLP output and the 4DVar method result to speed up the DA process. Arcucci et al. (2021) trained a recurrent neural network with the state of the dynamical system and the results of the DA process to learn the assimilation process, using the distance between the dynamical system prediction and the DA results as the training loss function. Fablet et al. (2021) proposed an appealing solution to learn the unknown dynamic mapping in the variational formulation jointly to computationally efficient solvers for the DA problem and achieved superb reconstruction performances. While these trends are encouraging, these NNs are not explicitly grounded in physics, making it challenging to produce initial fields consistent with the kinetic features of the system.

This paper aims to enhance the accuracy of initial fields by integrating a 4DVar-form physical constraint into the NN while keeping the computational cost low. A loss function based on analysis-based 4DVar is derived to train the DA model using the NN. Furthermore, we compare our NN-based DA model with several traditional DA methods and the 4DVarNet model (Fablet et al., 2021). The primary contributions of our study are as follows:

- The proposed NN-based DA method in this study combines two essential elements. The first element is an NN architecture constructed using residual convolutional NNs and incorporates one-dimensional channel attention. This architecture enables end-to-end DA. The second element involves utilizing an analysis-based 4DVar-form loss function. This loss function is designed to provide the NN access to long-term kinetic information about the dynamic system.
- This end-to-end DA method offers novel techniques for the ML-based DA model training without the need for ground truths as training labels.
- When establishing the initial fields for numerical predictions, the trained model can avoid expensive online optimization regarding the cost function and produce initial fields that are comparable to that of the traditional DA method.

We evaluate our approach using the Lorenz96 physical model (Lorenz, 1996), a system of nonlinear differential equations that models atmospheric chaos and serves as a standard benchmark for DA and ML tasks in Earth science (Hassanzadeh et al., 2019; Brajard et al., 2020; Huang et al., 2021; Nonnenmacher & Greenberg, 2021; Dong et al., 2022). Lorenz96 is a good test case for our purposes, as it can be accurately differentiated automatically by any deep learning (DL) framework. We systematically investigate how the prediction skill depends on the DA method and how observational errors affect the method. We also demonstrate how the scalability of our NN for Lorenz96 physical models varies with different variable numbers. Our work incorporates insights and techniques from previous studies using NNs to approximate DA methods (Wu et al., 2021; Arcucci et al., 2021) but is the first to our knowledge to break the performance upper bound of the ML-based DA method when tested on the Lorenz96 physical model.

The rest of this paper is organized as follows. Section 2 introduces the architecture design of our NN-based DA model and the theoretical derivation of our loss function. Section 3 presents the experimental design of our work. Section 4 shows the experimental results compared with traditional DA methods. Section 5 discusses how the proposed method relates to and is different from previous works, as well as the 4DVar method. Section 6 concludes our work.

2 Methods

2.1 Preliminaries

This study considers a chaotic system that describes the changing atmospheric states throughout time (e.g., atmospheric fluctuations across a spatial grid). The following explicit, fixed-time step numerical model can represent the system:

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\Big|_{\mathbf{x}=\mathbf{x}(t)} &= f(\mathbf{x}(t)), & (1)\\ \mathbf{x}_k = \mathcal{M}_{k:k-1}(\mathbf{x}_{k-1}) &\approx \mathbf{x}(t_{k-1}) + \textstyle\int_{t_{k-1}}^{t_k} f(\mathbf{x}(t))\mathrm{d}t, & (2) \end{aligned}$$

where $\mathbf{x}_k = \mathbf{x}(t_k) \in \mathbb{R}^m$ denotes the system state at t_k moment, and m denotes the system space's grid number. The integration model, $\mathcal{M}_{k:k-1} : \mathbb{R}^m \mapsto \mathbb{R}^m$, is usually a chaotic partial differential equation, which maps the system state at t_{k-1} moment into the state at t_k moment. The system is assumed to be Lipschitz continuous. The Picard–Lindelöf theorem (Coddington & Levinson, 1984) demonstrates that such an initial value problem has a unique solution. In discrete time, the system state can be observed through

$$\mathbf{y}_k = \mathcal{H}(\mathbf{x}_k) + \varepsilon_k^o, \tag{3}$$

where $\mathcal{H} : \mathbb{R}^m \mapsto \mathbb{R}^n$ denotes the observation operator and n denotes the observation space's grid number. The observation operator $\mathcal{H}$ is utilized to observe a set of local points from the whole system. The observation error is expressed as a system-independent random error ε_k^o, mainly comprising instrumentation and representation errors. Assuming that the observation errors follow a Gaussian distribution, *i.e.,* $\mathcal{H} = \mathbf{diag}(a_1, a_2, \cdots, a_m)$ and $\varepsilon_k^o \sim \mathcal{N}(0, \mathbf{R})$, where $\mathbf{R} = \sigma_o^2 \mathbf{I}_{n\times n}$ denotes the observation error covariance matrix (Frei & Künsch, 2013; Bocque et al., 2015). The observation error covariance matrix was set to the same matrix $\mathbf{R}$ when performing the assimilation. The background field is the short-term prediction by the numerical model. It can be defined as follows:

$$\mathbf{x}_k^b = \mathcal{M}_{k:k-1}(\mathbf{x}_{k-1}^a), \tag{4}$$

where $\mathbf{x}_{k-1}^a$, called the "analysis field", is obtained from a DA method. Our study needs to fuse the background fields and observations to provide accurate initial fields using the NN $\mathcal{N}$:

$$\mathcal{N}(\mathbf{x}_0^b, \mathbf{y}; \theta) : \mathbf{x}_0^b, \mathbf{y} \mapsto \mathbf{x}_0^n, \tag{5}$$

where $\mathbf{x}_0^n$ denotes the NN's assimilation result, $\mathbf{y}$ describes the observations in the assimilation window, and the goal is to make the NN output to approach the true system state $\mathbf{x}_0^t$ at the initial time t_0, i.e., $(\|\mathbf{x}_0^n - \mathbf{x}_0^t\| \sim 0)$. The θ indicates all the parameters in our NN-based model.

2.2 Architecture Design of Our NN-based DA Model

In this work, we develop a residual fully convolutional NN (ResFCNN) model to fuse background fields and observations into accurate initial fields. Our NN-based DA model architecture incorporates three common properties of DA systems: (1) spatial structure, (2) local dependencies, and (3) increment of the analysis fields. Figure 1 illustrates the overall architecture of the model. To capture spatial structure, we use convolutional NNs with stacked layers of trainable convolutional filters followed by Gaussian error linear unit

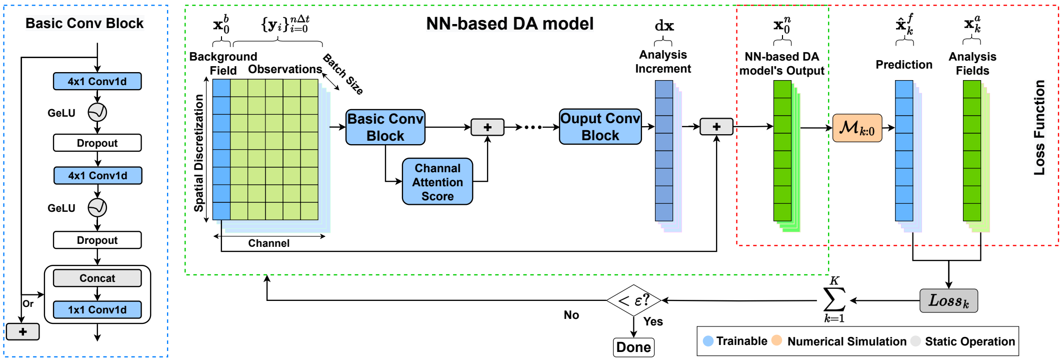

Figure 1. The overall training framework of our proposed NN-based DA model. The blue dashed box shows the structure of the basic convolutional block of the model. The green dashed box represents the pipeline of the model. The background field ($\mathbf{x}_0^b$) and the series of observations ($\mathbf{y}$) are concatenated together as input, and the analysis increment ($\mathrm{d}\mathbf{x}$) is obtained by a stack of basic convolutional blocks and channel attention, which are then added to the background field to obtain the fusion product ($\mathbf{x}_0^n$). The red dashed box illustrates the training logic of the model. The fusion product ($\mathbf{x}_0^n$) is used as the initial field of the numerical model ($\mathcal{M}_{k:0}$). Physical constraints constrain the numerically predicted trajectories using the analysis field provided by traditional DA methods.

(GeLU) (Hendrycks & Gimpel, 2016) as the activation function. GeLU is a smooth and monotonic activation function that has been shown to improve the performance of NNs. Our basic convolution block has two repeating convolution-activation-dropout structures, a skip-connection structure, and a final convolution composition. The one-dimensional convolution layers in our basic blocks are set to have kernel sizes equal to the neighborhood size, such that an element depends only on the state of the system in a local neighborhood around it. In particular, in the Lornz96 physical model, the neighborhood size is equal to 4. Our model uses a residual block (He et al., 2016) to combine the incremental field with the background field. The residual block is a type of NN structure that allows for the addition of the input of a model to its output. This allows for the model to easily learn the change from the background field to the final initial field. Finally, to integrate information on each channel, we reform the channel attention mechanism (CAM) (Hu et al., 2018) into a one-dimensional CAM (1DCAM) block. The 1DCAM block uses global average pooling to generate channel-wise statistics of features $\mathbf{z}$ coming from the output of a basic convolution module (see Figure 2). The 1DCAM block allows the model to learn the importance of each channel, which can then be used to improve the performance of the model. The channel feature aggregation module can be expressed as follows:

$$\mathbf{s} = \sigma(\mathbf{W}_2\delta(\mathbf{W}_1\mathbf{z})), \tag{6}$$

where σ denotes the sigmoid function, δ denotes the ReLU function (Glorot et al., 2011), $\mathbf{W}_1 \in \mathbb{R}^{C\times\frac{C}{r}}$ and $\mathbf{W}_2 \in \mathbb{R}^{\frac{C}{r}\times C}$ denote the squeeze and extend linear layers, respectively. In our study, r is set to be 8. The 1DCAM is set to be followed by each basic convolution module and the attention score ($\mathbf{s}$) is multiplied by the output of the basic convolutional block to enhance the channel-wise feature.

An assimilation cycle using our NN-based DA model can be represented as follows:

$$\text{Forecast Step} \qquad \mathbf{x}_k^b = \mathcal{M}_{k:k-1}\mathbf{x}_{k-1}^n, \tag{7}$$

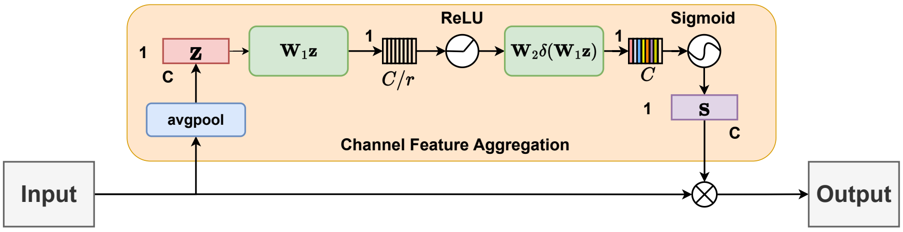

Figure 2. The schematic diagram of 1DCAM. The channel feature aggregation module is described by equation (6).

$$\text{Analysis Step} \qquad \mathbf{x}_k^n = \mathcal{N}(\mathbf{x}_k^b, \mathbf{y}; \theta) = \mathbf{x}_k^b + \mathcal{F}_{\text{inc}}(\mathbf{x}_k^b, \mathbf{y}_k; \theta), \tag{8}$$

where $\mathcal{N}$ denotes the model, $\mathcal{F}_{\text{inc}}$ represents the incremental field extraction component, and $\mathbf{y}$ denotes the observations during the assimilation window. To describe the detailed model architecture, we propose an architecture inspired by the model used in the work of (Nonnenmacher & Greenberg, 2021). Our basic block of the architecture consists of two 4×1 convolutional kernels followed by the GeLU activation function and a 1D-CAM that yield a k-channel feature map. It has a convolutional stride of 1 and employs circular padding we denote it as $b4s1$-k. Additionally, the dk notation indicates a layer that implements a 4×1 convolutional kernel, producing k-channel feature maps at both inputs and outputs. The model architecture with 4 blocks is then

$$\mathbf{x}_k^b, \mathbf{y}_k \rightarrow \text{b4s1}{-}32 \rightarrow \text{d32} \rightarrow \text{b4s1}{-}32 \rightarrow \text{b4s1}{-}64 \rightarrow \text{d64} \rightarrow \text{b4s1}{-}128 \rightarrow \text{d1} \rightarrow \mathcal{F}\text{inc}. \tag{9}$$

2.3 Loss Function for Model Training

Our overarching strategy aims to develop an ML-based DA method without ground truths as training labels. The objective is to achieve comparable or even higher accuracy than the SOTA traditional DA methods. Thus, the *prediction accuracy is the best evaluator* for a DA method. In the DA domain, the 4DVar method is successfully implemented by using a prediction task as a cost function to improve the initial field. In the widely used strong-constraint 4DVar (Le Dimet & Talagrand, 1986), the following cost function is optimized,

$$\mathcal{J}^{4DVar}(\mathbf{x}_0) = \mathcal{J}^B + \mathcal{J}^O = \frac{1}{2}\|\mathbf{x}_0 - \mathbf{x}^b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{l=0}^{L} \|\mathbf{y}_l - \mathcal{H} \circ \mathcal{M}_{k:0}(\mathbf{x}_0)\|^2_{\mathbf{R}^{-1}}, \tag{10}$$

where the $\circ$ symbol represents the composition of operators and $\|\cdot\|^2_{\mathbf{X}} = (\cdot)^T \mathbf{X}(\cdot)$ represents the Mahalanobis norm, l denotes the index of the observations during the DAW and L is the DAW length. $\mathbf{B}$ denotes the background error covariance matrix and $\mathbf{R}$ denotes the observation error covariance matrix. However, the 4DVar method is a computationally costly online learning strategy. To reduce the computational cost of 4DVar but maintain the meaningful physical constraint, we train the NN by using a 4DVar-form loss function. Figure 3 illustrates the process of 4DVar and our loss. Unlike 4DVar, our loss uses the analysis fields rather than the observations as the fitting objective. These fields are generated by traditional DA methods such as Ensemble Kalman filter (EnKF) (Evensen et al., 2009), 4DVar, iterative ensemble Kalman smoothing (IEnKS) (Bocquet & Sakov, 2014), etc., and are used exclusively for training the model. The background field $\mathbf{x}_{k+1}^b$ at time t_{k+1} is the result of the numerical prediction using the analysis field $\mathbf{x}_k^a$ at time t_k as the initial value. Assume that predictions are made with our NN-based DA model's result $\mathbf{x}_k^n$ at time

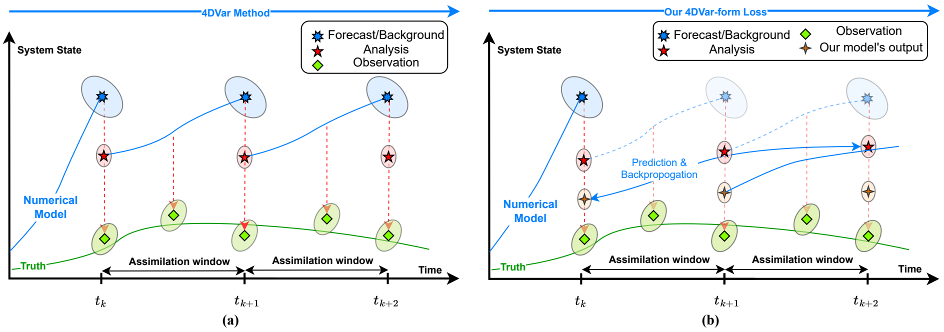

Figure 3. Illustration of 4DVar and our loss function. Figure 3(a) shows the pipeline of the 4DVar method. The control variable (blue multi-pointed stars) is the state at the beginning of the assimilation window $\mathbf{x}^b_{t_k}$. The whole cost function of 4DVar contains two parts, the background part $\mathcal{J}^B$ and the observation part $\mathcal{J}^O$. $\mathcal{J}^B$ is the distance between the analysis field (red five-pointed stars) and the background field. $\mathcal{J}^O$ is constructed by the distance between the prediction trajectory and observations (green diamonds). Corrections are computed at the time of observation but then propagated back to the start of the assimilation window using the adjoint model. Once the cost function is optimized, the analysis field at t_k is used to run a prediction until t_{k+1}. Figure 3(b) represents the main idea of our 4DVar-form loss function. The output of our NN-based DA model at t_k (the four-vertex orange star) is used as the initial field to run a numerical prediction until t_{k+2}. The prediction trajectory (blue double arrow curve) is moved forward to the analysis fields (red five-pointed stars) generated by traditional methods. Corrections are computed at t_{k+1} and t_{k+2} but then propagated back to optimize the parameters of the model using the backpropagating process of a DL framework.

t_k as the initial field. Suppose that the predictions approach the analysis fields $\mathbf{x}^a_{k+1}, \mathbf{x}^a_{k+2}$ at time $t_k + 1$ and $t_k + 2$. It is reasonable to conclude that our NN-based DA model is better suited to generate initial fields than the NNs trained to just approximate traditional DA methods. This is because the analysis field combines the information from both the numerical predictions and the observations, and is in general much closer to the true system state than the background field. Thus, we can embed optimal representations of the system's kinetic features into the NN by using analysis fields as training labels. Accordingly, by assuming that the analysis fields at each moment are independent of each other, we can derive our loss function as follows.

Predictions from an NWP model and observations can be fused using a well-known traditional DA method to obtain analysis fields $\mathbf{x}^a$. Let $\mathbf{x}^t$ be the system state truth and let $\widetilde{\mathbf{x}}^a = \mathbf{x}^a - \mathbf{x}^t$. The error covariance matrix $\mathbf{A} = \mathbb{E}[\widetilde{\mathbf{x}}^a(\widetilde{\mathbf{x}}^a)^T]$ is positive definite. The error is assumed to follow a Gaussian distribution, i.e., $\mathbf{x}^a \sim \mathcal{N}(\mathbf{x}^t, \mathbf{A})$. Then, the probability density function of the occurrence of the historical analysis fields $\mathbf{x}^a$ is

$$p(\mathbf{x}^a) = \frac{1}{(2\pi)^{\frac{m}{2}} (\det\mathbf{A})^{1/2}} \exp[-\frac{1}{2}\|\mathbf{x}^a - \mathbf{x}^t\|^2_{\mathbf{A}^{-1}}]. \tag{11}$$

The analysis fields were used as the labels to train the model to simulate a mapping relationship from the background fields and observations to an optimal estimate of the system state truth. Under the condition that the above error distribution is satisfied and the NWP model error is ignored, the probability of the analysis fields being the predicted result should be the highest. The probability of the analysis fields occurrence at moments after the initial moment is as follows:

$$\begin{aligned} p(\mathbf{x}_k^a) &= \frac{1}{(2\pi)^{\frac{m}{2}}(\det\mathbf{A})^{1/2}}\exp[-\tfrac{1}{2}\|\mathbf{x}_k^a-\mathbf{x}_k^t\|^2_{\mathbf{A}^{-1}}] && (12)\\ &= \frac{1}{(2\pi)^{\frac{m}{2}}(\det\mathbf{A})^{1/2}}\exp[-\tfrac{1}{2}\|\mathbf{x}_k^a-\mathcal{M}_{k:0}(\mathbf{x}_0^t)\|^2_{\mathbf{A}^{-1}}]. && (13)\end{aligned}$$

Thus, the probability density function of the fit to the series of analysis fields is as follows:

$$\begin{aligned} \prod_{k=1}^{K} p(\mathbf{x}_k^a) &= \textstyle\prod_{k=1}^{K}\frac{\exp[-\frac{1}{2}\|\mathbf{x}_k^a-\mathcal{M}_{k:0}(\mathbf{x}_0^t))\|^2_{\mathbf{A}^{-1}}]}{(2\pi)^{\frac{m}{2}}(\det\mathbf{A})^{1/2}} && (14)\\ &= C\cdot\exp[-\tfrac{1}{2}\textstyle\sum_{k=1}^{K}\|\mathbf{x}_k^a-\mathcal{M}_{k:0}(\mathbf{x}_0^t))\|^2_{\mathbf{A}^{-1}}], && (15)\end{aligned}$$

where C denotes a positive constant term. The analysis field $\mathbf{x}_k^a$ obtained by solving using the traditional assimilation method will theoretically maximize the above probability, thus, when we have obtained the analysis field $\mathbf{x}_k^a$, we only need to optimize the following cost function to obtain $\mathbf{x}_0^n$ that satisfies the objective of this work:

$$\mathcal{J}(\hat{\mathbf{x}}_0)=\frac{1}{K}\sum_{k=1}^{K}\|\mathbf{x}_k^a-\mathcal{M}_{k:0}(\mathbf{x}_0^n)\|^2_{\mathbf{A}^{-1}}, \tag{16}$$

where $\mathbf{x}_0^n$ denotes the accurate initial field to be found. In our loss function, the background term is ignored. The proof of the effectiveness of this loss function can be found in Supporting Information. Moreover, by considering the norm of the matrix $\mathbf{A}^{-1/2}$, such that

$$\|\mathbf{A}^{-1/2}\|^2=\max_{\mathbf{x}\neq\mathbf{0}}\frac{\|\mathbf{A}^{-1/2}\mathbf{x}\|^2}{\|\mathbf{x}\|^2}=\max_{\mathbf{x}\neq\mathbf{0}}\frac{\mathbf{x}^T\mathbf{A}^{-1/2^T}\mathbf{A}^{-1/2}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}=\max_{\mathbf{x}\neq\mathbf{0}}\frac{\|\mathbf{x}\|^2_{\mathbf{A}^{-1}}}{\|\mathbf{x}\|^2}=\lambda_{\max}\left(\mathbf{A}^{-1}\right), \tag{17}$$

where $\lambda_{\max}(\mathbf{A}^{-1})>0$ is the largest eigenvalue of $\mathbf{A}^{-1}$.

Adherence to the equations above enables us to derive the following inequation:

$$\|\mathbf{x}\|^2_{\mathbf{A}^{-1}}\leq\lambda_{\max}(\mathbf{A}^{-1})\|\mathbf{x}\|^2. \tag{18}$$

Thus, we have

$$\mathcal{J}(\hat{\mathbf{x}}_0)=\sum_{k=1}^{K}\|\mathbf{x}_k^a-\mathcal{M}_{k:0}(\mathbf{x}_0^n)\|^2_{\mathbf{A}^{-1}}\leq\lambda_{\max}(\mathbf{A}^{-1})\sum_{k=1}^{K}\|\mathbf{x}_k^a-\mathcal{M}_{k:0}(\mathbf{x}_0^n)\|^2. \tag{19}$$

This means that the cost function $\mathcal{J}(\mathbf{x}_0)$ has an upper bound $\lambda_{\max}(\mathbf{A}^{-1})\sum_{k=1}^{K}\|\mathbf{x}_k^a-\mathcal{M}_{k:0}(\mathbf{x}_0^n)\|^2$, and we can consider optimizing this upper bound to implicitly optimize the corresponding cost function. Thus we use the unit array $\mathbf{I}$ to implement the loss function rather than directly computing the unknown analysis error covariance matrix A to implement it, thus transforming the loss function into:

$$\mathcal{L}=\frac{1}{K}\sum_{k=1}^{K}\|\mathbf{x}_k^a-\mathcal{M}_{k:0}(\mathbf{x}_0^n)\|^2. \tag{20}$$

Algorithm 1: The procedure to train our NN-based DA model.

```
Input  :  $\mathbf{x}^b$   background fields
          $\mathbf{y}$     observations
          $\mathbf{x}^a$   analysis fields generated by a kind of traditional DA method
Output:   $\mathbf{x}^n$   Initial fields generated by our NN model.
          $\mathbf{x}^p$   Numerical predictions by utilizing $\mathbf{x}^n$ as initial fields.
Result: Training of a NN that learns the optimal solution of our 4Var-form
    loss function in equation (20)
Initialization: Set the number of training epochs $n_e$, batch size $n_b$, Adam hyper-parameters $\alpha, \beta_1, \beta_2$, initial parameters for our NN-based DA model $\theta$, the prediction constrained window size $K$.
for $i = 1, 2, \cdots, n_e$ do
    Sample $n_b$ snapshots $\{(\mathbf{x}^b(j), \mathbf{y}(j)\}_{j=1}^{n_b}$;
    Sample $n_b$ training labels $\{(\mathbf{x}_1^a(j), \mathbf{x}_2^a(j), \cdots, \mathbf{x}_K^a(j)\}_{j=1}^{n_b}$;
    for $j = 1, 2, \cdots, n_b$ do
        $\mathbf{x}^n(j) = \mathcal{N}(\mathbf{x}^b(j), \mathbf{y}(j), \theta))$ ;
        for $k = 1, 2, \cdots, K$ do
            if $k = 1$ then
                $\mathbf{x}_k^p(j) = \mathcal{M}_{k:k-1}(\mathbf{x}_0^b(j))$
            else
                $\mathbf{x}_k^p(j) = \mathcal{M}_{k:k-1}(\mathbf{x}_{k-1}^p(j))$
            end
        end
    end
    $\mathcal{L} = \frac{1}{n_b}\sum_{j=1}^{n_b}\frac{1}{K}\sum_{k=1}^{K}\|\mathbf{x}_k^a(j) - \mathbf{x}_k^p(j)\|^2$
    $\frac{\partial\mathcal{L}}{\partial\theta} = \frac{1}{n_b}\sum_{j=1}^{n_b}\frac{1}{K}\sum_{k=1}^{K}\frac{\partial\mathcal{L}}{\partial\mathbf{x}_k^p(j)}\frac{\partial\mathbf{x}_k^p(j)}{\partial\mathbf{x}_0^n(j)}\frac{\partial\mathbf{x}_0^n(j)}{\partial\theta}$
    $= \frac{1}{n_b}\sum_{j=1}^{n_b}\frac{1}{K}\sum_{k=1}^{K}\frac{\partial\mathcal{L}}{\partial\mathbf{x}_k^p(j)}\mathbf{M}_{k:0}\frac{\partial\mathbf{x}_0^n(j)}{\partial\theta}$
    $\theta \leftarrow \mathrm{Adam}(\frac{\partial\mathcal{L}}{\partial\theta}, \theta, \alpha, \beta_1, \beta_2)$
end
* All gradients are computed using automatic differentiation, and $\mathbf{M}_{k:0}$ is the tangent linear of the physical model.
```

3 Experimental Design

The Lorenz96 physical model is used as the experimental object to test the proposed method. It can be expressed as follows:

$$\frac{\mathrm{d}\mathbf{x}_j}{\mathrm{d}t} = (\mathbf{x}_{j+1} - \mathbf{x}_{j-2})\mathbf{x}_{j-1} - \mathbf{x}_j + F, \tag{21}$$

where $j = 1, 2, \cdots, J$ and $\mathbf{x}_{-1} = \mathbf{x}_{J-1}, \mathbf{x}_0 = \mathbf{x}_J, \mathbf{x}_{J+1} = \mathbf{x}_1$. J denotes the number of discrete lattice points of the system, set to 40 in the common experiments, and the forcing term F is set to 8. This setting is the most widely used for a test system in DA algorithms (Bocquet et al., 2019; Brajard et al., 2020; Wu et al., 2021). In our general experiments, all ensemble assimilation methods were performed with an ensemble number of 20. One integration step of the model and the observing time interval were set to be 0.01 unit time (Huang et al., 2020). An assimilation window was set to be 0.05 unit time which simulates a 6-h window in the real world. The integration numerical scheme we adopted is the 4th-order Runge-Kutta method, as proposed by Lorenz in his 1996 paper (Lorenz, 1996). The assimilation cycle times are set to four years in all experiments, where each year has 365 days. After training the model with the data generated by 4DVar, EnKF, local ensemble transform Kalman filter (LETKF) (Hunt et al., 2007), and IEnKS, all evaluation experiments were compared with the corresponding algorithm in the same ten sets of initial fields for 4-year assimilation cycles. Further, the assimilation windows for 4DVar and IEnKS are both set to be 0.05. All results contained the mean and standard deviation (±). The DA and prediction codes are implemented based on the DAPPER (Raanes et al., 2018) framework with our trained model for proper validation. The metrics compared are the root mean square errors (RMSEs) of the analysis fields and predictions at the beginning of each assimilation window with the system state truths, which can be expressed as follows:

$$\mathbf{RMSE}_a = \sqrt{\frac{\sum_{i=1}^{n}(\mathbf{x}_i^a - \mathbf{x}_i^t)^2}{N_{cyc}}}, \tag{22}$$

$$\mathbf{RMSE}_f = \sqrt{\frac{\sum_{i=1}^{n}[\mathcal{M}(\mathbf{x}_{i-1}^a) - \mathbf{x}_i^t]^2}{N_{cyc}}}, \tag{23}$$

where $\mathbf{RMSE}_a$ denotes the RMSE between the analysis fields and system state truths, $\mathbf{RMSE}_f$ denotes the RMSE between the predictions and system state truths, and N_{cyc} is the number of assimilation cycles. Our NN-based DA model, as well as the auto-differentiable Lorenz96 physical model, are written in Pytorch. The AdamW (Loshchilov & Hutter, 2017) optimizer was used with a cosine adaptive learning rate strategy (Loshchilov & Hutter, 2016). In fully observed experiments, the learning rates were all set to be $3e-4$, while in partially observed experiments, the learning rates were all set to be $1e-3$. The NN-based 4DVar model was trained for 50 epochs, and early stopping was set to avoid overfitting. The models were trained on V100 GPUs and tested on Intel(R) Core(TM) i7-1065G7 CPUs. All experiments are performed simultaneously on both full observations and 75% of the observations (for processing of partial observations when using ML-based DA method, see Figure 4). This strategy of filling in unobserved grid points can offer a more comprehensive understanding of the data, even if there are discrepancies between the predicted data and the observed distribution. Additionally, the NN has the capability to remove the added noise.

3.1 Data Preparation

For the numerical experiments shown in the following sections, the database is made of $N_{exp} = 12$ trajectories. The initial values are set as $\{1, 0, 0, \cdots, 0\}$, and the 12 initial values are sampled from a Gaussian random vector space with 0 as the expectation and 0.001 as the standard deviation separately. In the experiment step, the first trajectory is used for training, the second is used for validation, and the 10 remainings are used for testing. In contrast to most ML studies, larger test datasets are used for testing to obtain reliable test metrics.

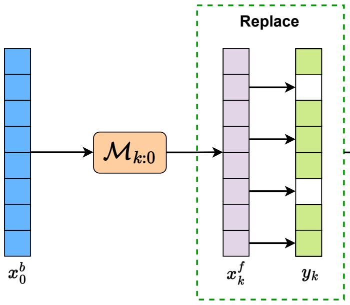

Figure 4. The partial observations are included in the input of the 4DVar-constrained NN-based DA model by replacing the missing values with the predicted values.

Table 1. HyperParametric search space for traditional DA methods.

Method	Parameter	Range
4DVar	B-Scale	0.02,0.04,0.06,0.08,0.1,0.2,0.4,0.6,0.8,1.0, 2.0, 4.0, 6.0, 8.0
EnKF	Inflation Rotation	1.0,1.02,1.04,1.06,1.08,1.1 True, False
LETKF	Inflation Rotation Localization Radiation	1.0,1.02,1.04,1.06,1.08,1.1 True, False 2, 4, 6, 8, 10
IEnKS	Inflation Rotation	1.0,1.02,1.04,1.06,1.08,1.1 True, Flase

3.1.1 Ground Truth and Observations

Starting from a set of initial conditions, $\mathbf{x}_0^{(i)}(i = 1, \cdots, N_{exp})$, of the true model, we computed two trajectories to generate the ground truth databases. In addition, we created two databases of observations as follows. The observation error follows a zero-mean Gaussian distribution with covariance matrix $\mathbf{R} = \sigma_o^2 \mathbf{I}_{n \times n}$. We set the observation standard deviations to 1 in the general experiments, while in the observation sensitive experiment, σ ranges from 1 to 3 with 0.5 as the step. The 75% partially observed data is generated by setting the second out of every four grid points as unobserved.

3.1.2 Background and Analysis Fields

We perturb the initial fields and take them as initial values. All background and analysis fields are obtained through the following analysis and prediction loops:

$$\text{Forecast Step} \qquad \mathbf{x}_k^b = \mathcal{M}_{k:k-1} \mathbf{x}_{k-1}^a, \tag{24}$$

$$\text{Analysis Step} \qquad \mathbf{x}_k^a = \mathcal{DA}(\mathbf{x}_k^b, \mathbf{y}_k), \tag{25}$$

where $\mathbf{x}_0^a = \mathbf{x}_0 + \xi$, $\mathbf{x}_0$ denotes the exact initial value with random Gaussian error $\xi \sim \mathcal{N}(0, \mathbf{\Xi})$, $\mathbf{\Xi}$ denotes the random initial error covariance, $\mathcal{DA}$ represents the compared traditional DA method, and $\mathbf{y}_k$ expresses the observations of the DA method used during the DAW. When assimilate observations utilizing the 4DVar method, $\mathbf{y}_k = \{\mathbf{y}_{k0}, \mathbf{y}_{k1}, \cdots, \mathbf{y}_{k(L-1)}\}$ where L denotes the number of moments with observations in the assimilation window and is set as 5. It is crucial to note that hyperparameters also significantly affect the accuracy of traditional methods. We searched for them minutely. See Table 1 for specific search parameters. We performed a grid search on each traditional DA method (such as the variance

Table 2. Evaluation of our NN-based DA model and traditional DA methods on different observation ratios between 75% and 100%. We have tested the analysis and prediction field RMSEs of our NN-based DA model trained with analysis fields generated from different traditional DA methods as well as the SOTA 4DVarNet method and different CNN models of varying complexity trained using our loss function.

Methods	**Fields**	**EnKF**		**LETKF**		**IEnKS**		**4DVar**	
		75%	100%	75%	100%	75%	100%	75%	100%
Traditional (Baseline)	Analysis	0.225 ± 0.009	0.185 ± 0.004	0.220 ± 0.002	0.181 ± 0.004	0.208 ± 0.003	0.177 ± 0.004	0.201 ± 0.002	0.159 ± 0.001
	Prediction	0.246 ± 0.003	0.203 ± 0.005	0.241 ± 0.002	0.198 ± 0.004	0.228 ± 0.003	0.193 ± 0.004	0.215 ± 0.003	0.171 ± 0.001
4DVarNet (supervised)	Analysis	0.209 ± 0.003	0.169 ± 0.001	0.215 ± 0.022	0.165 ± 0.001	0.216 ± 0.026	0.169 ± 0.002	0.207 ± 0.002	1.218 ± 1.563
	Prediction	0.224 ± 0.003	0.181 ± 0.001	0.231 ± 0.023	0.178 ± 0.001	0.233 ± 0.026	0.178 ± 0.001	0.222 ± 0.002	1.249 ± 1.589
TinyNet	Analysis	0.360 ± 0.004	0.287 ± 0.003	0.293 ± 0.005	0.291 ± 0.003	0.301 ± 0.005	0.295 ± 0.003	0.275 ± 0.003	0.299 ± 0.002
	Prediction	0.379 ± 0.005	0.300 ± 0.004	0.309 ± 0.005	0.305 ± 0.004	0.319 ± 0.006	0.309 ± 0.004	0.290 ± 0.004	0.321 ± 0.003
TinyResNet	Analysis	0.247 ± 0.003	0.187 ± 0.002	0.227 ± 0.002	0.192 ± 0.002	0.229 ± 0.003	0.192 ± 0.002	0.217 ± 0.003	0.185 ± 0.002
	Prediction	0.264 ± 0.004	0.199 ± 0.002	0.231 ± 0.004	0.205 ± 0.002	0.246 ± 0.003	0.205 ± 0.002	0.217 ± 0.003	0.201 ± 0.002
ResNet	Analysis	0.206 ± 0.002	0.155 ± 0.002	0.181 ± 0.002	0.157 ± 0.002	0.183 ± 0.002	0.157 ± 0.002	0.192 ± 0.002	$\mathbf{0.149} \pm 0.001$
	Prediction	0.222 ± 0.002	0.166 ± 0.002	0.196 ± 0.002	0.169 ± 0.002	0.199 ± 0.002	0.169 ± 0.002	$\mathbf{0.205} \pm 0.002$	0.161 ± 0.001
Ours	Analysis	$\mathbf{0.196} \pm 0.002$	$\mathbf{0.149} \pm 0.002$	$\mathbf{0.179} \pm 0.002$	$\mathbf{0.150} \pm 0.002$	$\mathbf{0.178} \pm 0.002$	$\mathbf{0.152} \pm 0.002$	$\mathbf{0.191} \pm 0.001$	$\mathbf{0.149} \pm 0.001$
	Prediction	$\mathbf{0.211} \pm 0.002$	$\mathbf{0.160} \pm 0.002$	$\mathbf{0.195} \pm 0.002$	$\mathbf{0.162} \pm 0.002$	$\mathbf{0.194} \pm 0.002$	$\mathbf{0.164} \pm 0.002$	$\mathbf{0.205} \pm 0.002$	$\mathbf{0.160} \pm 0.001$

Note. The first two rows of the table describe the quality of the analysis and forecast fields using traditional data assimilation methods, which are defined as baseline results for comparison. The third and fourth rows describe the performance of the 4DVarNet model trained with true values as labels, which is referred to as supervised training by Fablet et al. (2021). The remaining rows describe the best results obtained from training different CNN models of varying complexity using the loss function proposed in this paper. The best results throughout the experiment are highlighted in bold.

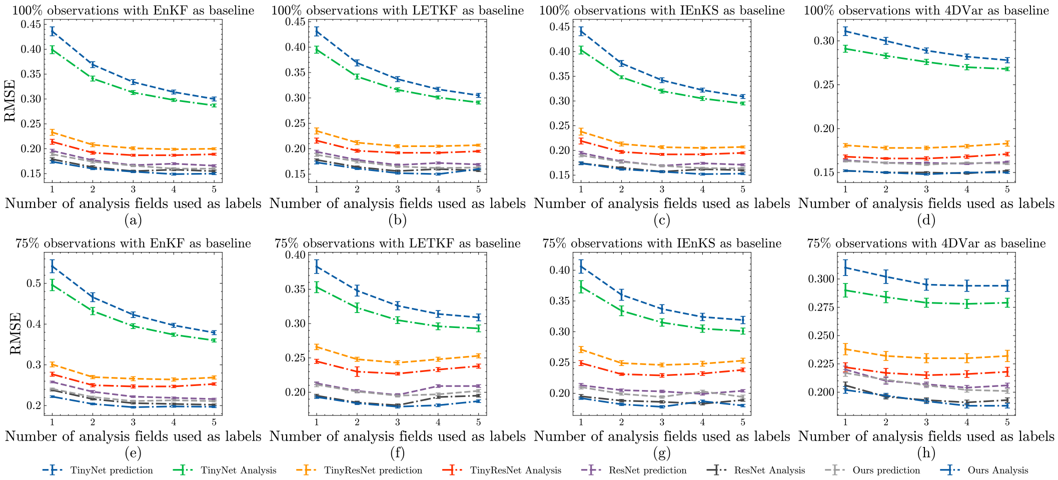

Figure 5. Plot of the error analysis for models trained with different prediction lengths K in the loss function. The first row shows the analysis and prediction errors when the system is fully observed. The second row represents the errors when 75% of grid points are observed.

inflation coefficient, localization radiation, and whether to include rotation). The results of the hyperparameters search are provided in Supporting Information Table S2-S9. After testing, for each tested traditional DA method, the parameter combination with the smallest root mean square error and the most stable results with respect to the truth of the system state is selected. Table 1 lists all the searched parameters.

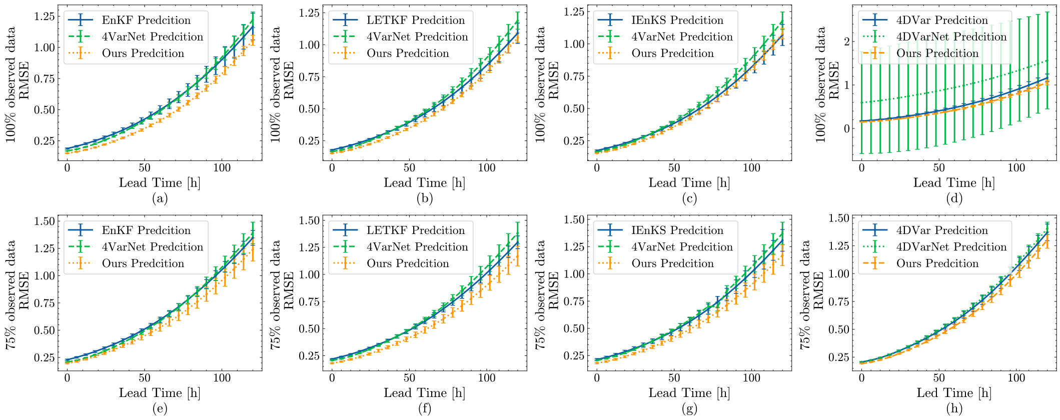

Figure 6. The 6-hour error curve for simulating 5-day forecasts using the analysis field is plotted. The first line is the comparison of our method with baseline and 4DVarNet at 100% observation, and the second line is the comparison at 75% observation.

Table 3. Ratios of the running speed of the analysis and prediction loop processes (our NN-based DA model and 4DVarNet compared to traditional methods).

Method	EnKF	LETKF	IEnKS	4DVar
4DVarNet	0.068 ± 0.003	0.207 ± 0.008	0.446 ± 0.011	1.475 ± 0.111
Ours	0.675 ± 0.033	2.055 ± 0.066	4.430 ± 0.212	14.605 ± 0.702

4 Experimental Results

4.1 Comparision to traditional DA methods and several ML-based DA models

Table 2 presents the RMSEs of the tested DA methods applied to the Lorenz96 physical model with fully and partially spaced direct observations of the state variables. As baselines for the DA experiments, we consider four different traditional DA methods. All NN-based DA models, except 4DVarNet, were trained using analysis fields generated by relevant baseline methods. The 4DVarNet model was implemented based on the publicly available code of Fablet et al. (2021). It was trained using a supervised learning strategy, with the ground truths as training labels. In contrast, the TinyNet, TinyResNet, and ResNet models were implemented based on the publicly available code of Nonnenmacher and Greenberg (2021), which can capture the dynamic features of the Lorenz96 physical model quite well and can learn the solution to our proposed loss function. Comparing these models partly demonstrates that our proposed architecture can learn the solution to the proposed cost function more accurately. Table 2 highlights the best results obtained in bold font. It is evident that 4DVarNet achieves comparable results with the baselines since it is a surrogate optimizer for 4DVar. However, when background fields generated from 4DVar were utilized as training input, the RMSEs of 4DVarNet exhibited large standard deviations. This might be because the small background field error used in training 4DVarNet contributes to inadequate estimation of the background error after training, leading to an unreasonable implicit depiction of the weights for the background and observations by 4DVarNet. On the other hand, our method demonstrated a significant performance improvement compared to all other methods. For instance, our method showed a 14.4% reduction in analysis RMSE when tested on partially observed data, compared to IEnKS. Additionally, our method consistently produced accurate analysis fields, as evidenced by the low standard deviation of errors in all outcomes. The results suggest that the use of analysis fields to constrain the predicted trajectories could potentially impose implicit physical constraints on the output of our NN-based DA model, which could be beneficial in improving the accuracy of the predictions. Furthermore, our method achieves a quality improvement regardless of the baseline method used to provide the training labels. In addition, our 1D-CAM gives our NN-based model the ability to learn the better solution of our loss function than the ResNet model (Nonnenmacher & Greenberg, 2021).

The results of training various machine learning models using the loss function proposed in our study are exhibited in Figure 5. We use analysis fields of multiple time steps as labels and range the length of the prediction constraint. As the number of analysis fields increased, most of the models showed a decreasing trend in analysis and forecasting errors that signified the effectiveness of our proposed loss function. The simplest TinyNet model showed a continuous decrease in error as the number of analysis fields increased. This finding indicates that we can improve the performance of simple models by augmenting the number of analysis fields and prediction length restricted by the loss. Our NN-based DA model structure has higher abilities in extracting system physics information described by the proposed loss function, as indicated by the performance improvement of our NN-based DA model over ResNet. However, the RMSEs of our model fluctuate with different training trajectory lengths. This may be related to the non-linear dynamic nature of the system. Nevertheless, after incorporating the 4DVar-form loss function, our model can produce higher quality analysis fields and more accurate predictions on the Lorenz96 physical model than the compared traditional DA methods. In addition, we can potentially reduce the computational cost of the training process by reducing the constrained trajectory length without sacrificing much performance, which is beneficial in applications where computational resources are limited. Our results demonstrate that the integration of 4DVar-form physical constraints with NNs can significantly improve the quality of ML-based DA.

In order to better assess the impact of the assimilation method on the prediction, we further investigate the variation in the error of the 5-day 6-hour prediction using the aforementioned analysis field. The experimental results are depicted in Figure 6. The yellow curve represents the prediction error of the analysis field obtained through our algorithm, the blue curve represents the outcome of the corresponding traditional DA method, and the green curve represents the result of 4DVarNet trained with ground truth values. In the experiment with a 100% observation ratio, the background field predicted by the 4DVar method is employed as the training input for 4DVarNet, which yields unstable results. This instability may be attributed to the fact that the background field error itself is not significant, thereby leading to a weak error correction capability acquired by 4DVarNet. Eight experiments conducted using the Lorenz96 physical model demonstrate that the analysis field obtained by our method exhibits the lowest predicting error, and the results are sufficiently stable. This is precise because our NN-based DA model incorporates prediction as a constraint to ensure that the output satisfies the objective of minimizing the prediction error of the Lorenz96 physical model.

In addition, the running time ratio of the analysis and prediction loop processes (our NN-based DA model and 4DVarNet compared to traditional methods) was also reported in Table 3. From the table, we can see that our NN-based DA model was faster than IEnKS, LETKF, and 4DVar. It proves that our NN-based DA model could accelerate the DA process. In particular, compared to the 4DVar method on the Lorentz96 physical model, our method achieves a speedup ratio of 14.

We further illustrate randomly selected time series with 200 assimilation cycles in Figure 7. The simulations are based on random initial fields and assume that 75% of the grid points are observed. The difference between ground truths and predictions is shown in Figure 7(c) and 7(e). The results demonstrate that our NN-based DA model can produce more accurate predictions than baseline methods, as the distance of the prediction to the true system state is lower, both spatially and temporally. Figure 7(f)-(j) presents the RMSEs of predictions at five randomly chosen grid points, with the overall RMSEs of the predictions with our NN-based DA model being lower than those of IEnKS. Comparisons of our NN-based DA model with 4DVar, EnKF, and LETKF are shown in Supporting Information Figures S1 to S3. These results demonstrate that our NN-based DA model can produce more accurate predictions than the compared traditional DA methods.

4.2 The Ability to Absorb Different Error Observations

The quality control (QC) of observations is a critical factor in the quality of DA results (Sakov & Sandery, 2017; Jin et al., 2019). Inaccurate observations can lead to suboptimal initial fields, necessitating the development of DA methods that can effectively assimilate observations with higher errors. To evaluate the robustness of our pretrained model in Section 4.1 without retraining, we conducted experiments with 5 different observation error variations from 1 to 5. Further, these values were input into the DA system to adjust the corresponding observation error covariances to achieve the prescribed standard deviation. The results, shown in Figure 8, demonstrate that our NN-based DA model can assimilate observations with higher errors more efficiently than the compared traditional DA methods, as evidenced by the slower increase in RMSEs of the analysis and prediction results. Furthermore, our NN-based DA model exhibits much smaller standard deviations than traditional DA methods. This feature of our NN-based DA model makes the QC process easier. From Figure 8e, with an increase in observation error, the assimilation results given by EnKF fluctuate significantly, attributable to the instability of the gain matrix calculation after an increase in observation error. Furthermore, the same results can be seen in the experiments when assimilating partial observations using the LETKF and IEnKS methods. In contrast to these methods, our NN-based DA model consistently gives stable assimilation results. This also demonstrates that our method can effectively assimilate observations with higher errors.

4.3 Experiments on Lorenz96 Physical Models with Larger Variable Numbers

A key characteristic of the data assimilation method is its scalability with the increase of system variables. In this section, we demonstrate the effectiveness of our method on an extended version of the Lorenz96 physical model with more variables. We apply the NN method based on 4DVar constraints and validate it on Lorenz96 models with different numbers of variables ranging from 100 to 500. We compare our method with the LETKF method, which is able to handle variable expansion well, as well as the 4DVarNet method, and report the error growth curves for a 5-day forecast. In this experiment, we still use the LETKF method with 20 ensemble members as the baseline, and we tuned the parameters according to Table 1, the tuning results can be found in Supporting Information Table S1. The results of the forecast cycle experiments are shown in Table 4. In the simulation experiments with a 75% observation ratio, our method achieved at least a 14% reduction in background field error and a 12.7% reduction in analysis error compared to LETKF. It also improved performance compared to 4DVarNet, which was trained using the ground truth as labels. In addition, the results of the 5-day forecast experiments are shown in Figure 9, which demonstrates stable error reduction for our method compared to LETKF and 4DVarNet. These results indicate the scalability of our method in the extended Lorenz96 physical model with more variables.

5 Related Work

In this section, we further discuss how the proposed method relates to and is different from previous works, as well as the 4DVar method. To the best of our knowledge, our work—while drawing on these earlier works—is the first ML-based approach to learn from and provide superior initial fields over traditional DA methods.

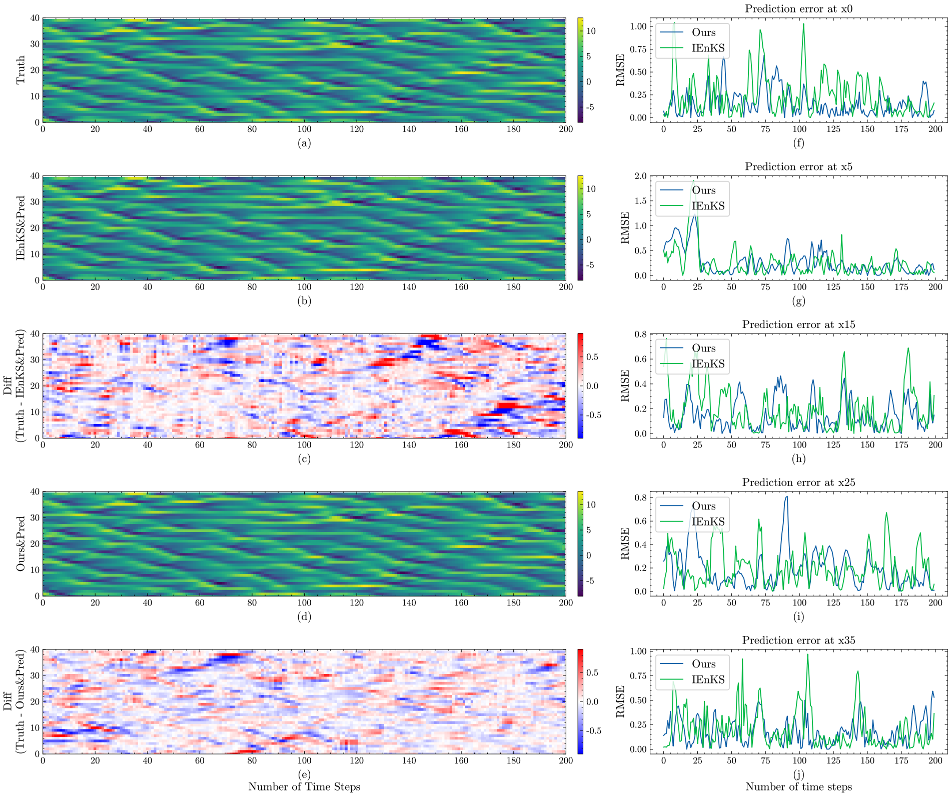

Figure 7. Visualization of the comparison between our NN-based DA model and IEnKS for assimilation cycles. We randomly display a time series of 200 assimilation windows. Both simulations observe 75% of the grid points as observation data. Figure 7(a) represents the truths of the Lorenz96 physical model provided by pure numerical prediction from a random initial field. Figure 7 (b) shows the system states are generated by numerical prediction with IEnKS as the DA method. The IEnKS method is used at each assimilation window and the simulation starts from a random initial field. Figure 7 (c) represents the difference between (a) and (b). Figure 7 (d) shows the system states generated by numerical prediction with our NN-based DA model as the DA method. our NN-based DA model is used at each assimilation window and the simulation starts from a random initial field. Figure 7 (e) represents the difference between (a) and (d). Figure 7(f)-(j) represents the RMSEs of predictions with our NN-based DA model as the DA method compared to predictions with IEnKS as the DA method. The green line is the RMSEs of the predictions using IEnKS as the DA method. The blue line is the RMSEs of the predictions with our NN-based DA model as the DA method. Five randomly chosen grid points are represented.

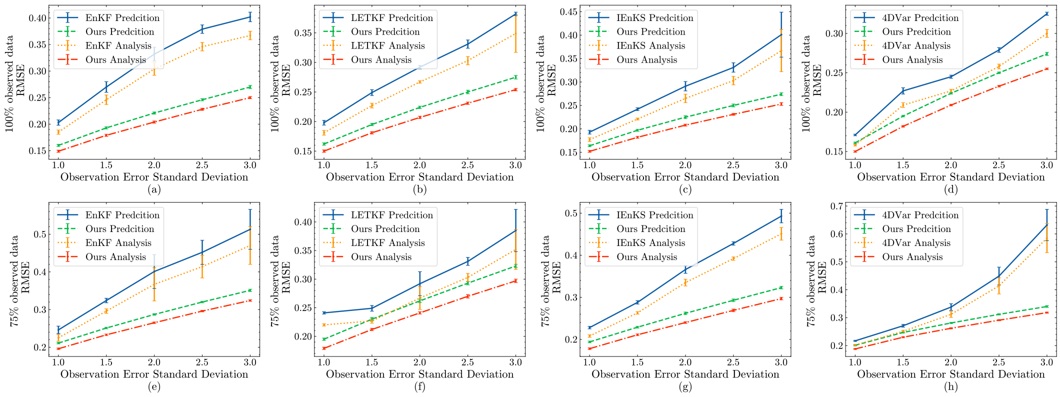

Figure 8. Plot of the error analysis for increasing observational error standard derivation. The first row shows the analysis and prediction errors when the system is fully observed. The second row represents the errors when 75% of grid points are observed. The solid blue line represents the error in the prediction from the analysis field generated by the traditional DA method, and the green dashed line represents the error in the prediction from the analysis field generated by our NN-based DA model. The orange dashed line represents the error in the analysis field generated by the traditional DA method, and the red dashed line represents the error in the analysis field generated by our NN-based DA model. From left to right, the results of our NN-based DA model are compared with EnKF, LETKF, IEnKS, and 4DVar.

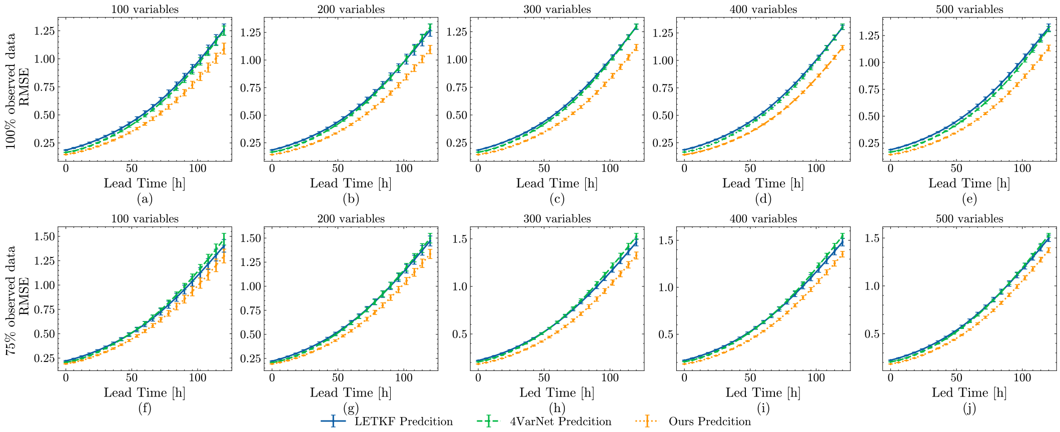

Figure 9. Variations in the 5-day simulation forecast error using different assimilation methods on an expanded variable Lorenz96 physical model. The number of variables ranges from 100 to 500. The first line represents the results with a 100% observation ratio, while the second line represents the results with a 75% observation ratio. The blue line represents the error for LETKF, the green line represents the error for 4DVarNet, and the yellow line represents the error for our method.

Table 4. The performance of the proposed method was compared with LETKF and 4DVarNet on the Lorenz96 physical model with more variables. We have tested the analysis and prediction field RMSEs of our NN-based DA model trained with analysis fields generated from LETKF as well as the SOTA 4DVarNet method and LETKF.

Variables	**Methods**	**Fields**	**LETKF**	
			75%	100%
100	LETKF	Analysis	0.219 ± 0.003	0.185 ± 0.001
		Prediction	0.240 ± 0.003	0.202 ± 0.001
	4DVarNet (supervised)	Analysis	0.205 ± 0.002	0.166 ± 0.001
		Prediction	0.221 ± 0.003	0.178 ± 0.002
	Ours	Analysis	**0.191** ± 0.002	**0.146** ± 0.001
		Prediction	**0.206** ± 0.002	**0.159** ± 0.001
200	LETKF	Analysis	0.222 ± 0.003	0.185 ± 0.001
		Prediction	0.244 ± 0.003	0.203 ± 0.001
	4DVarNet (supervised)	Analysis	0.207 ± 0.001	0.166 ± 0.001
		Prediction	0.223 ± 0.002	0.179 ± 0.001
	Ours	Analysis	**0.193** ± 0.001	**0.143** ± 0.001
		Prediction	**0.209** ± 0.001	**0.179** ± 0.001
300	LETKF	Analysis	0.223 ± 0.002	0.187 ± 0.005
		Prediction	0.245 ± 0.002	0.205 ± 0.005
	4DVarNet (supervised)	Analysis	0.207 ± 0.001	0.165 ± 0.001
		Prediction	0.223 ± 0.001	0.178 ± 0.001
	Ours	Analysis	**0.184** ± 0.001	**0.142** ± 0.000
		Prediction	**0.199** ± 0.001	**0.154** ± 0.000
400	LETKF	Analysis	0.237 ± 0.033	0.185 ± 0.001
		Prediction	0.259 ± 0.033	0.204 ± 0.001
	4DVarNet (supervised)	Analysis	0.207 ± 0.001	0.165 ± 0.000
		Prediction	0.223 ± 0.001	0.178 ± 0.000
	Ours	Analysis	**0.173** ± 0.000	**0.141** ± 0.000
		Prediction	**0.188** ± 0.000	**0.153** ± 0.001
500	LETKF	Analysis	0.206 ± 0.001	0.187 ± 0.001
		Prediction	0.273 ± 0.060	0.206 ± 0.001
	4DVarNet (supervised)	Analysis	0.206 ± 0.001	0.166 ± 0.000
		Prediction	0.223 ± 0.001	0.179 ± 0.000
	Ours	Analysis	**0.184** ± 0.001	**0.142** ± 0.000
		Prediction	**0.200** ± 0.001	**0.154** ± 0.000

5.1 ML-based method for DA

In recent years, the integration of ML, DA, and uncertainty quantification has demonstrated promising outcomes in enhancing the accuracy and comprehension of models in diverse fields (Cheng et al., 2023). Our work builds on the growing literature describing ML-based methods for learning and aiding the DA processes. These efforts fell into the following three groups: 1) ML-based tangent linear and adjoint models; 2) ML-based surrogate models for ensemble DA; and 3) ML-based models for directly dealing with DA tasks.

5.1.1 ML-based Tangent Linear and Adjoint Models

A variety of ML-based surrogate models have been proposed for replacing tangent linear and adjoint models of 4DVar methods (Nonnenmacher & Greenberg, 2021; Kotamarthi, 2022; Dong et al., 2022). These studies generally learn the numerical model by relying on an NN. The minimization of the 4DVar cost function is achieved by using the NN's backpropagation and some kind of gradient descent methods. For instance, in Nonnenmacher and Greenberg (2021), a differentiable emulator was trained on the Lorenz96 physical model and applied to the 4DVar assimilation method. This work proved that the Jacobians of the differentiable emulator and the numerical system show close agreement, and the differentiable emulator can provide missing derivatives for the 4D-Var method without greatly degrading forecast accuracy. Furthermore, Dong et al. (2022) also proved that the auto differentiable function of the DL framework could provide a simple adjoint model for the 4DVar method. Additionally, in Kotamarthi (2022), the differentiable reduced-order surrogate model is merged into an optimization strategy where observations of the genuine state are used to enhance the forecast of the surrogate. This work assessed the long short-term memory model on a real-world forecasting task for geopotential height and obtained competitive results to climatology and persistence baselines for mean absolute error. Although most of the hybrid ML-4DVar methods focus on the efficient adjoint process, they also require iteratively optimizing the cost function. It still consumes more computational resources than the end-to-end process of our NN-based DA model. Furthermore, the quality of the initial fields generated by these hybrid ML-4DVar methods was similar to that of 4DVar, which is lower than our methods on the Lorenz96 physical model. This may contribute to the longer-term and more comprehensive information provided by our loss than 4DVar's cost function. Thus, our NN-based DA model is potentially an alternative DA method for accurate end-to-end assimilation.

5.1.2 ML-based Surrogate Models for Ensemble DA

Some works seek to build data-driven surrogate models combined with ensemble DA methods to predict the future(Brajard et al., 2020; Chattopadhyay et al., 2021, 2023), e.g., Brajard et al. (2020), who built an iterative algorithm with the EnKF (Evensen et al., 2009) to generate the initial field and then alternate with an NN to learn the Lorenz96 physical model. By tuning certain parameters of the algorithm (the number of forecast steps of the NN and the standard deviation of the model noise in DA), it was possible to favor the prediction skill over the long-term dynamics reconstruction. Furthermore, a sigma-point ensemble Kalman algorithm and the U-STN model were also integrated in Chattopadhyay et al. (2021) to provide stable, accurate DA cycles for geopotential height prediction. It showed that the gain from applying DA to an ML-based surrogate model would be most significant when the observations are noisy and sparse. Additionally, Chattopadhyay et al. (2023) employs a pretrained ML-based surrogate model that generates and evolves a large ensemble of states cheaply to compute the background error covariance matrix with smaller sampling errors. This work estimates a better initial condition without the need for any ad-hoc localization strategies. Recently, some works investigate the possibility of learning both the state and dynamics of a physical system online, to update their estimates when new observations are acquired, using sequential DA techniques such as the EnKF and a simple representation for the surrogate model and state augmentation (Bocquet, Farchi, &

Malartic, 2020; Malartic et al., 2022). Malartic et al. (2022) investigated the possibility of integrating a local EnKF with a data-driven surrogate dynamical core to jointly estimate the state and parameters of the system. Peyron et al. (2021) proposed an ETKF-Q-L method that learned the latent structure of the dynamic using an autoencoder to reduce the computational cost and memory storage. These interdisciplinary approaches, which combine ML and DA, have shown promising results in improving the accuracy and interpretability of models across various domains (Bocquet, 2023). However, the demand for huge ensemble members requires more external storage and computational resources than the proposed end-to-end model. The online calculation for a large background covariance matrix and its inversion is another term leading to unavoidable computational cost. Thus, our NN-based DA model can provide a better trade-off between the computational cost and assimilation quality.

5.1.3 ML-based Models for Directly Dealing with DA tasks

Many researchers aim to introduce the applications of NN design to approximate the mapping from the background fields and observations to the analysis fields (Cintra et al., 2016; Pawar et al., 2020; Wu et al., 2021; Arcucci et al., 2021). Cintra et al. (2016) presents the ML-based approach to emulate the LETKF method. With greater computing performance and comparable quality to LETKF analyses, the DA procedure is carried out by employing the NN to obtain the initial conditions for the atmospheric global model. In Pawar et al. (2020), an LSTM embedding model is recommended to estimate the nudging term, which not only drives the state trajectories to the observations but also acts as a stabilizer. Wu et al. (2021) introduced a fast DA (FDA) method that replaces the DA process by training an NN with 4DVar results as target outputs. When tested on the Lorenz63 system, FDA outperforms 4DVar in terms of computational performance under the premise of similar quality. Furthermore, in Arcucci et al. (2021), a recurrent NN trained with the state of the dynamical system and the results of the 3DVar process is applied for DA purposes. Fablet et al. (2021) utilized the automatic differentiation tools embedded in DL frameworks to learn a variational model and a gradient-based solver both implemented as NNs. Lafon (2023) proposed an algorithm that jointly learns a parametric distribution of the state, the dynamics governing the evolution of the parameters, and a solver. These works successfully accelerated the process of DA but were not explicitly grounded in physics, making it challenging to produce initial fields consistent with the kinetic features of the system. Our NN-based DA model is also an end-to-end solution for DA tasks. We can not only take advantage of the low computational cost but also provide higher-quality initial fields. Thus, our study provides a new idea for building accurate ML-based DA methods.

5.2 Relationship with PINNs and 4DVar

To make the NN satisfy the basic physical laws described by PDEs, a class of physics-informed machine learning methods (Raissi et al., 2019; Sirignano & Spiliopoulos, 2018) is introduced to solve the forward and inverse problems involving PDEs. These approaches use auto differentials to compute spatial or temporal derivatives and use PDEs as the training loss. These approaches provide new ideas for combinatorial physics and data-driven approaches. Moreover, in the traditional DA area, the 4DVar method (Peng et al., 2017) minimizes cost functions to optimize 1) the fit of the initial field to the background field and 2) the mapping from the state of the model to the observations. The initial field is the one that leads to an accurate numerical prediction that fits the observations well. The success of solving the PDE-based variational problem in both 4DVar and PINNs indicates the suitability of 4DVar-form physical constraint loss for training ML-based DA models, especially when no direct pixel-wise ground truth exists. This further suggests that training with 4DVar-form loss functions may enable NNs to generate initial fields that can drive accurate predictions, as the 4DVar-form physical constraint loss can provide an accurate representation of the system's kinetic features. Thus, we use a series of analysis fields to constrain the prediction trajectories starting from our NN-based DA model's output. This is because the observed data are usually sparse and irregular, whereas the analysis field is complete and distributed over the grid points. Using analysis fields as constraint targets can make the cost function converge more easily. At the same time, the analysis field is usually physically consistent with the numerical model, which also allows our NN-based DA model to learn a more physically stable result.

6 Discussion and Conclusion

6.1 Contributions

In this paper, we introduce a novel 4DVar-constrained ML-based DA method for efficient and high-quality DA. This method combines the computational efficiency of NNs with the physical constraints of the 4DVar method. As the full-field "state" of Earth is unavailable due to the sparsity of observations, a comparison of an NN's output and pixel-by-pixel ground truth pairs is not possible. To address this issue, we constructed a 4DVar-form loss function using analysis fields as fitting targets. Numerical experiments on the Lorenz96 physical model show that the ability of the 4DVar-form constrained NN can improve the ML-based DA method's accuracy while giving an approximately 14-fold speedup ratio over the 4DVar method. When put to the test on the 500-variables Lorenz96 physical model, the experimental results show that our approach can achieve at least a 13% reduction in RMSEs compared to baseline traditional methods. The main advantage of our approach is that it does not require system truths as training labels and inherently incorporates the 4DVar-form physical consistency of the system.

6.2 Limitations and Future Work

While this and other works have successfully trained ML-based DA models on simple systems such as Lorenz96, it is less certain whether they can scale to more complex systems with additional spatial variables and additional interaction variables. Here, we highlight some limitations of the current method. Such issues will be subject to future research.

6.2.1 Nonlinear and weakly constrained problems

In real-world DA applications, the numerical prediction models are always strongly nonlinear, and the models all have errors. In order to make the proposed method suitable for these situations multiple approaches may be helpful. For example, we find that the spatial organization and localization of the system aid in reducing the size of the function space where we search for the ML-based DA model. Nonlinear and weakly constrained problems can be overcome by learning to improve the parameterized schemes or to correct model errors (Farchi et al., 2021; Bonavita & Laloyaux, 2020).

6.2.2 Scalability

The input to the NN may not resemble the training data, which is a concern when fusing unexpected observations with a trained DA model. This issue may affect the performance of the NN-based model. One might resolve this problem by using regularization techniques (Sanchez-Gonzalez et al., 2020) that introduce noise into the inputs during training. However, these potential solutions require additional experimentation and research before they are likely to solve the corresponding problems. Furthermore, in complex systems, considering the error covariance matrices may allow us to describe the spatial and physical correlations between variables, thereby enhancing the method's adaptability to the system.

6.2.3 Computational cost

The existing method is employed for estimating initial fields in the Lorenz96 physical model. However, implementing a large numerical prediction model using deep learning frameworks like PyTorch proves to be time-consuming. Furthermore, the iterative running of the numerical model for training DA models results in substantial costs. Consequently, the immense size of the Earth system presents a significant challenge in applying the methods outlined in this paper to future real-world NWP processes. One possible solution to this problem is to explore the use of reduced-order modeling techniques as a replacement for the physical prediction model used in the 4DVar-form constraint discussed in this paper. With the advancements in artificial intelligence technology, ML has demonstrated its potential in the field of medium-term forecasting. This is evident in recent works such as FourCastNet (Kurth et al., 2023), PanGu-Weather (Bi et al., 2023), GraphCast (Lam et al., 2023), ClimaX (Nguyen et al., 2023), and et al. However, it is essential to address the training cost when incorporating an ML prediction model as a constraint.

6.3 Summary

In summary, our approach combines the physical constraints of the 4DVar method with the computational efficiency of NNs. Users can solve costly assimilations much faster with our NN-based 4DVar method. It potentially exemplifies how ML methods can be leveraged to improve both the efficiency and quality of DA techniques without system truths as training labels. By applying the 4DVar-form loss function for model training, NNs can also improve the quality of the initial field. These improvements are due to the combined effect of physical laws and NNs, which are still undergoing rapid improvement: modern physics-informed ML methods allow accelerating numerical methods with much more compact representations by following fundamental physical laws. We expect the trend toward physics-informed ML to continue for the foreseeable future and eventually improve our predictive skills for Earth.

Open Research Section

The software DAPPER used to produce the data presented in this manuscript is available in Raanes et al. (2018). The code used to do experiments in this manuscript is available in wuxinwang (2023).

Acknowledgments

This work was supported by the science and technology innovation Program of Hunan Province (2022RC3070) and the National Natural Science Foundation of China (Grant Nos. 42205161 and 42275170). The authors would like to thank Juan Zhao, Renze Dong, and Shuo Ma for providing valuable suggestions for this article. The authors are grateful to the editor and three anonymous reviewers who helped improve the quality of this paper.

References

Arcucci, R., Zhu, J., Hu, S., & Guo, Y.-K. (2021). Deep data assimilation: integrating deep learning with data assimilation. *Applied Sciences*, *11*(3), 1114.

Bannister, R. (2017). A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, *143*(703), 607–633.

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

Bocque, M., Raanes, P. N., & Hannart, A. (2015). Expanding the validity of the ensemble kalman filter without the intrinsic need for inflation. *Nonlinear Processes in Geophysics Discussions*.

Bocquet, M. (2016). Localization and the iterative ensemble kalman smoother. *Quarterly Journal of the Royal Meteorological Society*, *142*(695), 1075–1089.

Bocquet, M. (2023). Surrogate modelling for the climate sciences dynamics with machine learning and data assimilation. *Frontiers in Applied Mathematics and Statistics*, *9*, 22.

Bocquet, M., Brajard, J., Carrassi, A., & Bertino, L. (2019). Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models. *Nonlinear Processes in Geophysics*, *26*(3), 143–162.

Bocquet, M., Brajard, J., Carrassi, A., & Bertino, L. (2020). Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization. *Foundations of Data Science*, *2*(1), 55–80.

Bocquet, M., Farchi, A., & Malartic, Q. (2020). Online learning of both state and dynamics using ensemble kalman filters. *Foundations of Data Science*, *3*(3), 305–330.

Bocquet, M., & Sakov, P. (2014). An iterative ensemble kalman smoother. *Quarterly Journal of the Royal Meteorological Society*, *140*(682), 1521–1535.

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

Boukabara, S.-A., Krasnopolsky, V., Penny, S. G., Stewart, J. Q., McGovern, A., Hall, D., . . . others (2020). Outlook for exploiting artificial intelligence in the earth and environmental sciences. *Bulletin of the American Meteorological Society*, 1–53.

Brajard, J., Carrassi, A., Bocquet, M., & Bertino, L. (2020). Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the lorenz 96 model. *Journal of Computational Science*, *44*, 101171.

Carrassi, A., Bocquet, M., Bertino, L., & Evensen, G. (2018). Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *Wiley Interdisciplinary Reviews: Climate Change*, *9*(5), e535.

Chattopadhyay, A., Mustafa, M., Hassanzadeh, P., Bach, E., & Kashinath, K. (2021). Towards physically consistent data-driven weather forecasting: Integrating data assimilation with equivariance-preserving deep spatial transformers. *arXiv preprint arXiv:2103.09360*.

Chattopadhyay, A., Nabizadeh, E., Bach, E., & Hassanzadeh, P. (2023). Deep learning-enhanced ensemble-based data assimilation for high-dimensional nonlinear dynamical systems. *Journal of Computational Physics*, 111918.

Cheng, S., Quilodrán-Casas, C., Ouala, S., Farchi, A., Liu, C., Tandeo, P., . . . others (2023). Machine learning with data assimilation and uncertainty quantification for dynamical systems: a review. *IEEE/CAA Journal of Automatica Sinica*, *10*(6), 1361–1387.

Cintra, R., de Campos Velho, H., & Cocke, S. (2016). Tracking the model: Data assimilation by artificial neural network. In *2016 international joint conference on neural networks (ijcnn)* (pp. 403–410).

Clayton, A. M., Lorenc, A. C., & Barker, D. M. (2013). Operational implementation of a hybrid ensemble/4d-var global data assimilation system at the met office. *Quarterly Journal of the Royal Meteorological Society*, *139*(675), 1445–1461.

Coddington, E. A., & Levinson, N. (1984). Theory of ordinary differential equations. *Physics Today*, *9*(2).

Compo, G. P., Whitaker, J. S., Sardeshmukh, P. D., Matsui, N., Allan, R. J., Yin, X., . . . others (2011). The twentieth century reanalysis project. *Quarterly Journal of the Royal Meteorological Society*, *137*(654), 1–28.

Courtier, P., Thépaut, J.-N., & Hollingsworth, A. (1994). A strategy for operational implementation of 4d-var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, *120*(519), 1367–1387.

Dong, R., Leng, H., Zhao, J., Song, J., & Liang, S. (2022). A framework for four-dimensional variational data assimilation based on machine learning. *Entropy*, *24*(2). Retrieved from `https://www.mdpi.com/1099-4300/24/2/264` doi: 10.3390/e24020264

Düben, P., Modigliani, U., Geer, A., Siemen, S., Pappenberger, F., Bauer, P., . . . others

(2021). Machine learning at ecmwf: A roadmap for the next 10 years. *European Centre for Medium-Range Weather Forecasts, Tech. Rep*, *878*.

Evensen, G., et al. (2009). *Data assimilation: the ensemble kalman filter* (Vol. 2). Springer.

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

Farchi, A., Laloyaux, P., Bonavita, M., & Bocquet, M. (2021). Using machine learning to correct model error in data assimilation and forecast applications. *Quarterly Journal of the Royal Meteorological Society*, *147*(739), 3067–3084.

Frei, M., & Künsch, H. R. (2013). Mixture ensemble kalman filters. *Computational Statistics & Data Analysis*, *58*, 127–138.

Gettelman, A., Geer, A. J., Forbes, R. M., Carmichael, G. R., Feingold, G., Posselt, D. J., … Zuidema, P. (2022). The future of earth system prediction: Advances in model-data fusion. *Science Advances*, *8*(14), eabn3488.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 315–323).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT Press. (`http://www.deeplearningbook.org`)

Gustafsson, N., Janjić, T., Schraff, C., Leuenberger, D., Weissmann, M., Reich, H., … others (2018). Survey of data assimilation methods for convective-scale numerical weather prediction at operational centres. *Quarterly Journal of the Royal Meteorological Society*, *144*(713), 1218–1256.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., … others (2022). A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, *45*(1), 87–110.

Hassanzadeh, P., Chattopadhyay, A., Palem, K., & Subramanian, D. (2019). Data-driven prediction of a multi-scale lorenz 96 chaotic system using a hierarchy of deep learning methods: Reservoir computing, ann, and rnn-lstm. In *Aps division of fluid dynamics meeting abstracts* (pp. C17–009).

He, K., Zhang, X., Ren, S., & Sun, J. (2016, June). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr).*

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., … others (2020). The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, *146*(730), 1999–2049.

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 7132–7141).

Huang, L., Leng, H., Li, X., Ren, K., Song, J., & Wang, D. (2021, February). A Data-Driven Method for Hybrid Data Assimilation with Multilayer Perceptron. *Big Data Research*, *23*, 100179. doi: 10.1016/j.bdr.2020.100179

Huang, L., Leng, H., Song, J., Zhao, J., Chen, R., & Wang, D. (2020). A hybrid 3dvar-enkf data assimilation approach based on multilayer perceptron. In *2020 international joint conference on neural networks (ijcnn)* (pp. 1–10).

Hunt, B. R., Kostelich, E. J., & Szunyogh, I. (2007). Efficient data assimilation for spatiotemporal chaos: A local ensemble transform kalman filter. *Physica D: Nonlinear Phenomena*, *230*(1-2), 112–126.

Jin, J., Lin, H. X., Segers, A., Xie, Y., & Heemink, A. (2019). Machine learning for observation bias correction with application to dust storm data assimilation. *Atmospheric Chemistry and Physics*, *19*(15), 10009–10026.

Kenton, J. D. M.-W. C., & Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacl-hlt* (pp. 4171–4186).

Kotamarthi, R. M. R. W. M. C. L. B. F. (2022). Efficient high-dimensional variational data assimilation with machine-learned reduced-order models. *Geoscientific Model Development*, 3433–3445.

Kurth, T., Subramanian, S., Harrington, P., Pathak, J., Mardani, M., Hall, D., … Anandkumar, A. (2023). Fourcastnet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators. In *Proceedings of the platform for advanced scientific computing conference* (pp. 1–11).

Lafon, N. (2023). Uncertainty quantification when learning dynamical models and solvers with variational methods. In *103rd ams annual meeting.*

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., … others (2023). Learning skillful medium-range global weather forecasting. *Science*, eadi2336.

Le Dimet, F.-X., & Talagrand, O. (1986). Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A: Dynamic Meteorology and Oceanography*, *38*(2), 97–110.

Lorenz, E. (1996). Predictability-a problem partly solved. In *Proc seminar on predictability, reading, uk, ecmwf.*

Loshchilov, I., & Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Loshchilov, I., & Hutter, F. (2017). Fixing weight decay regularization in adam. *CoRR*, *abs/1711.05101*. Retrieved from `http://arxiv.org/abs/1711.05101`

Malartic, Q., Farchi, A., & Bocquet, M. (2022). State, global, and local parameter estimation using local ensemble kalman filters: Applications to online machine learning of chaotic dynamics. *Quarterly Journal of the Royal Meteorological Society*, *148*(746), 2167–2193.

Nguyen, T., Brandstetter, J., Kapoor, A., Gupta, J. K., & Grover, A. (2023). Climax: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*.

Nonnenmacher, M., & Greenberg, D. S. (2021). Deep emulators for differentiation, forecasting, and parametrization in earth science simulators. *Journal of Advances in Modeling Earth Systems*, *13*(7), e2021MS002554.

Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J. A., Orts-Escolano, S., Garcia-Rodriguez, J., & Argyros, A. (2020). A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(6), 2806–2826.

Pawar, S., Ahmed, S. E., San, O., Rasheed, A., & Navon, I. M. (2020). Long short-term memory embedded nudging schemes for nonlinear data assimilation of geophysical flows. *Physics of Fluids*, *32*(7), 076606.

Peng, W., Liang, X., Zhang, X., Huang, X., Lu, B., & Fu, Q. (2017). Application of physical filter initialization in 4dvar. *Monthly Weather Review*, *145*(6), 2201 - 2216. Retrieved from `https://journals.ametsoc.org/view/journals/mwre/145/6/mwr-d-16-0274.1.xml` doi: 10.1175/MWR-D-16-0274.1

Peyron, M., Fillion, A., Gürol, S., Marchais, V., Gratton, S., Boudier, P., & Goret, G. (2021). Latent space data assimilation by using deep learning. *Quarterly Journal of the Royal Meteorological Society*, *147*(740), 3759–3777.

Raanes, P. N., Grudzien, C., & 14tondeu. (2018, December). *nansencenter/dapper: Version 0.8.* (Version 0.8) [Computer software]. Zenodo. Retrieved from `https://doi.org/10.5281/zenodo.2029296` doi: 10.5281/zenodo.2029296

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, *378*, 686–707.

Sakov, P., & Sandery, P. (2017). An adaptive quality control procedure for data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, *69*(1), 1318031.

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., & Battaglia, P. (2020). Learning to simulate complex physics with graph networks. In *International conference on machine learning* (pp. 8459–8468).

Sirignano, J., & Spiliopoulos, K. (2018). Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, *375*, 1339–1364.

Wu, P., Chang, X., Yuan, W., Sun, J., Zhang, W., Arcucci, R., & Guo, Y. (2021). Fast data assimilation (fda): Data assimilation by machine learning for faster optimize model state. *Journal of Computational Science*, *51*, 101323.

wuxinwang. (2023, September). *wuxinwang1997/nn-4dvar-james: v1.0.0.* (Version 1.0.0) [Computer software]. Zenodo. Retrieved from `https://doi.org/10.5281/zenodo.8328977` doi: 10.5281/zenodo.8328977