A Scalable Real-Time Data Assimilation Framework for Predicting Turbulent Atmosphere Dynamics

Junqi Yin
National Center for Computational Science
Oak Ridge National Laboratory
Oak Ridge, TN, USA
yinj@ornl.gov

Siming Liang
Department of Mathematics
Florida State University
Tallahassee, FL, USA
sliang@fsu.edu

Siyan Liu
Computational Science and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN, USA
lius1@ornl.gov

Feng Bao
Department of Mathematics
Florida State University
Tallahassee, FL, USA
fbao@fsu.edu

Hristo G. Chipilski
Department of Scientific Computing
Florida State University
Tallahassee, FL, USA
hchipilski@fsu.edu

Dan Lu
Computational Science and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN, USA
lud1@ornl.gov

Guannan Zhang*
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN, USA
zhangg@ornl.gov

***Abstract*—AI-based foundation models like FourCastNet, GraphCast, ClimaX, and Pangu-Weather are revolutionizing weather and climate predictions but are not yet ready for operational use. Their limitation lies in the absence of a data assimilation system to incorporate real-time Earth system observations, crucial for accurately forecasting events like tropical cyclones and atmospheric rivers. To overcome these obstacles, we introduce a generic real-time data assimilation framework and demonstrate its end-to-end performance on the Frontier supercomputer. This framework comprises two primary modules: an ensemble score filter (EnSF), which significantly outperforms the state-of-the-art data assimilation method, e.g., the Local Ensemble Transform Kalman Filter (LETKF); and a vision transformer-based surrogate capable of real-time adaptation through the integration of observational data. We demonstrate both the strong and weak scaling of our framework up to 1024 GPUs on the Exascale supercomputer, Frontier. Our results not only illustrate the framework's exceptional scalability on high-performance computing systems, but also demonstrate the importance of supercomputers in real-time data assimilation for weather and climate predictions. Even though the proposed framework is tested only on a benchmark surface quasi-geostrophic (SQG) turbulence system, it has the potential to be combined with existing AI-based foundation models, making it suitable for future operational implementations.**

***Index Terms*—Diffusion models, generative AI, data assimilation, high-performance computing, foundation models, uncertainty quantification**

I. Introduction

The field of meteorology is undergoing a significant transformation owing to rapid advances in artificial intelligence (AI). While existing AI-based foundation models such as FourCastNet [2], GraphCast [3], ClimaX [4] and Pangu-Weather [5] show considerable potential, they are not ready yet for a fully operational implementation since they are decoupled from operational data assimilation (DA) algorithms. This limitation hinders their ability to dynamically incorporate real-time observational data and impacts their effectiveness in predicting complex atmospheric phenomena, such as tropical cyclones and atmospheric rivers. The reliance on physics-based models to provide the initial conditions significantly increases the overall computational costs. In the case of AIFS (the AI model developed by ECMWF), one still needs to combine the physics-based ECMWF model (IFS) with a four-dimensional DA (4D-Var) method in order to initialize the data-driven forecasts every 12h.

Data assimilation is crucial for making reliable weather forecasts because it involves the integration of real-time observational data with weather models, ensuring the models start from the most accurate representation of the current state of the Earth system. Within the Earth sciences, the ensemble Kalman filter (EnKF) of Evensen [6] and its many variants are a state-of-the-art (SOTA) DA method. For example, *one the finalist paper [1] of the 2023 ACM Gordon Bell Prize for Climate Modeling used the Local Ensemble Transform Kalman*

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan.

* Corresponding author

979-8-3503-5554-3/24/$31.00 ©2024 IEEE
DOI 10.1109/SCW63240.2024.00009

Filter (LETKF) that is a specific implementation of EnKF. However, EnKFs suffer from fundamental limitations as they make Gaussian assumptions in their update step, which leads to severe model biases in solving highly nonlinear systems. Previous studies have illustrated the detrimental effects of the resulting analysis biases in high-impact situations such as hurricane prediction [7]. A viable alternative to EnKF is the particle filter (PF) [8]–[10] – a fully non-parametric method which converges to the correct Bayesian solution [11]. Although PFs emerged around the same time as the EnKF, their implementation to large models has been difficult in view of their curse of dimensionality (weight collapse). In practical terms, this means that PFs require prohibitively large ensemble sizes (number of particles) to retain long-term stability.

To overcome these challenges, we propose a generic real-time DA framework and demonstrate its end-to-end performance on the Frontier supercomputer at the Oak Ridge Leadership Computing Facility (OLCF). This framework consists of two primary modules. The first module is an ensemble score filter (EnSF), originally developed in [12], [13]. The EnSF method leverages the generative AI models [14]–[17], and has shown promising accuracy in estimating the state of a high-dimensional Lorenz-96 system with $\mathcal{O}(10^6)$ variables and highly nonlinear observations. The second primary module of our DA framework is a vision transformer (ViT)-based surrogate of the forecast model that could be either a physics-based model or an AI-based foundation model. The surrogate model is needed in our DA framework for two reasons. First, the EnSF requires the gradient of the observation model to update the score function, and the gradient can be efficiently obtained from the surrogate model. Second, due to the complexity of turbulence, the forecast model does not provide sufficient accuracy without incorporating observation data. Nevertheless, the online training of the surrogate model requires the use of supercomputers to perform real-time DA.

Our results demonstrate the proposed framework's exceptional scalability on high-performance computing systems, which is essential for eventual application to real weather and climate prediction problems. Even though the proposed framework is tested using the benchmark surface quasi-geostrophic turbulence (SQG) model, it has the potential to be combined with existing AI-driven weather models, making it suitable for operational use. Our contributions are listed as follows:

- We introduce a generic real-time data assimilation framework for estimating turbulent dynamics, providing significantly more accurate predictions (Figure 4) than the state-of-the-art LETKF method which was used in one finalist paper [1] of the 2023 Gordon Bell Prize for Climate Modeling.
- We showcase the remarkable strong and weak scaling capabilities of our proposed DA framework on the Frontier supercomputer, which demonstrates the necessity of supercomputers in real-time data assimilation operation.
- We investigate the strategies for large-scale distributed training of ViTs, including compute-efficient kernel sizing on AMD MI250Xs, and memory-efficient data parallelisms for ViTs with billions of parameters.

The rest of this paper is organized as follows. In Section II, we introduce the physical SQG model and setup the data assimilation problem. Section III provides the details of the proposed framework, including the EnSF and the ViT-based surrogate model. The scalability experiments and results are given in Section IV, while Section V summarizes the main findings and and outlines our future plans.

II. Background

A. Data assimilation

Every DA algorithm requires a forecast model to describe how the physical system evolves over time, and a set of observations to reduce the growing forecast errors. Let $k = 0, 1, ..., K$ denote the time index. The general evolution of the system can be written as

$$\textbf{Forecast model:}\quad \boldsymbol{X}_k = \mathbf{f}_{k-1}(\boldsymbol{X}_{k-1}, \boldsymbol{E}^m_{k-1}), \qquad (1)$$

where $\boldsymbol{X}_k$ is the discretized state. Note that this forecast model could be either physics-based like the SQG, or an AI-based foundation model like FourCastNet. We further assume the model predictions are not perfect, and their errors captured by the random vector $\boldsymbol{E}^m_k$. To correct the model predictions, we use observations given by

$$\textbf{Observation model:}\ \boldsymbol{Y}_k = \mathbf{h}_k(\boldsymbol{X}_k) + \boldsymbol{E}^o_k, \qquad (2)$$

where $\mathbf{h}_k$ is the observation operator mapping the state to observation space and $\boldsymbol{E}^o_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is the corresponding observation error. In this case, we have made the simplifying assumption that observations are additive and Gaussian in nature, but more flexible models can be also used [18].

Given the forecast and observation models, a standard way to solve the DA problem is to calculate the filtering probability density function (PDF) $P(\mathbf{x}_k|\mathbf{y}_{1:k})$, in which the state is conditioned on the entire history of observations up to the present (filtering) time. This can be done by iterating through one prediction and one update (analysis) step, as described below.

Prediction: Due to its stochastic nature, the state is evolved forward using the Chapman-Kolmogorov equation such that

$$P(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int P(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})P(\mathbf{x}_k|\mathbf{x}_{k-1})d\mathbf{x}_{k-1}, \quad (3)$$

where $P(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the transition PDF to be determined from the forecast model (1).

Update: After the new measurements $\boldsymbol{Y}_k = \mathbf{y}_k$ are collected, the error-prone forecasts are adjusted using Bayes' theorem:

$$P(\mathbf{x}_k|\mathbf{y}_{1:k}) \propto P(\mathbf{x}_k|\mathbf{y}_{1:k-1})P(\mathbf{y}_k|\mathbf{x}_k), \qquad (4)$$

Accounting for the additive-Gaussian assumption on the observation errors $\boldsymbol{E}^o_k$, the likelihood $P(\mathbf{y}_k|\mathbf{x}_k)$ can be rewritten as

$$P(\mathbf{y}_k|\mathbf{x}_k) \propto \exp\Big[-(\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k))^\top \mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k))\Big]. \tag{5}$$

B. The surface quasi-geostrophic (SQG) model

The new prediction framework is tested on a benchmark model simulating the surface quasi-geostrophic (SQG) dynamics [19]. The numerical implementation follows [20] closely. It is important to emphasize that the proposed DA framework can be combined with any forecasting model, either physics-based or AI-driven, as described in Section III. Nevertheless, our choice to work with the SQG model for this study is motivated by its ability to generate turbulence behavior that is representative of real geophysical flows. In particular, fully developed turbulence in the SQG system follows a kinetic energy (KE) density spectrum with a -5/3 slope, which aligns with reference measurements from field campaigns [21]. Previous studies have shown that such turbulence characteristics set a limit on the ability to make reliable weather predictions [22].

III. Methodology

This section contains the details of the proposed real-time DA framework. The corresponding workflow is summarized in Figure 1. The major challenge of DA for operational use is that there is no existing method that can simultaneously resolve the following three issues: nonlinearity/non-Gaussianity, high-dimensionality, and scalability on HPC. The new DA method described next has demonstrated its ability to resolve all three issues, and has the potential to significantly improve SOTA weather and climate predictions. There are two major scalability tasks, one is the online training of the ViT surrogate using observational data, and the other is the efficient running of the EnSF. Since training ViT and running EnSF occurs sequentially with each filtering iteration, the overall computing time is the summation of the computing times for these two steps.

A. The ensemble score filter (EnSF)

1) Overview of diffusion models: To describe score-based diffusion models, we need to introduce the following stochastic differential equation (SDE)

$$d\boldsymbol{Z}_t = b(t)\boldsymbol{Z}_t dt + \sigma(t) d\boldsymbol{W}_t, \tag{6}$$

with $\boldsymbol{W}_t$ being the standard Brownian motion. The initial condition $\boldsymbol{Z}_0$ follows some target distribution, which in our case is set to the filtering PDF given by Eq. (4). The following reverse-time SDE can be used to generate samples $\{\boldsymbol{Z}_0^i\}_{i=1}^N$ of the target $\boldsymbol{Z}_0$:

$$d\boldsymbol{Z}_t = \big[b(t)\boldsymbol{Z}_t - \sigma^2(t)\mathbf{s}(\boldsymbol{Z}_t, t)\big]\, dt + \sigma(t) d\overleftarrow{\boldsymbol{W}}_t \tag{7}$$

where we have used the notation $\int \cdot d\overleftarrow{\boldsymbol{W}}_t$ to define a backward Itô stochastic integral [23], [24]. Within this new SDE, the term $\mathbf{s}(\cdot, t)$ is referred to as the score function and is a short-hand for

$$\mathbf{s}(\mathbf{z}_t, t) = \nabla \log(Q(\mathbf{z}_t)). \tag{8}$$

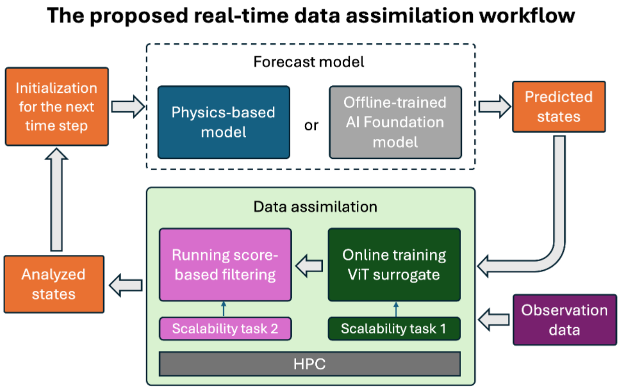
The proposed real-time data assimilation workflow
Forecast model
Initialization for the next time step
Physics-based model
or
Offline-trained AI Foundation model
Predicted states
Data assimilation
Analyzed states
Running score-based filtering
Online training ViT surrogate
Scalability task 2
Scalability task 1
Observation data
HPC

Fig. 1. Illustration of the real-time sequential DA workflow, which needs to be performed very frequently in weather forecast operation. Recent advances in weather and climate modeling focus on developing AI-based foundation models, e.g., FourCastNet, GraphCast, etc., to replace the traditional physics-based forecast models. These data-driven architectures are not yet ready for operational use due to the lack of real-time data assimilation capabilities. The proposed DA framework has two primary modules that need to be scaled on HPC, i.e., the ensemble score filter (EnSF) introduced in Section III-A, which significantly outperforms SOTA methods like LETKF, and a vision transformer (ViT)-based surrogate, introduced in Section III-B, capable of real-time adaptation through the integration of observational data. Our method can be integrated with either physics models or AI-based foundation models. The scalability of our method on HPC is essential to ensure computations can be performed in real time.

The score function $\mathbf{s}(\cdot, t)$ is an essential ingredient for transforming the standard Gaussian distribution of $\boldsymbol{Z}_T$ to the target distribution $Q(\mathbf{z}_0)$. One important technicality is that the drift and diffusion coefficients b and σ need to be properly chosen in order to obtain the desired transformation. Here we follow [14], [15] and define these functions as

$$b(t) = \frac{d \log \alpha_t}{dt}, \quad \sigma^2(t) = \frac{d\beta_t^2}{dt} - 2\frac{d \log \alpha_t}{dt}\beta_t^2, \tag{9}$$

with $\alpha_t = 1 - t$ and $\beta_t = \sqrt{t}$ for $t \in [0, 1]$.

2) The ensemble score filter (EnSF): The main philosophy behind EnSF, our new filtering approach, is to approximate the score functions $\mathbf{s}_{k|k-1}$ and $\mathbf{s}_{k|k}$ corresponding to the prior (forecast) and posterior PDFs in (3) and (4). At the beginning of the k-th iteration, we have the analysis ensemble at time level $k-1$, the EnSF's workflow reduces to the standard iterative application of prediction and update steps, as described next.

Prediction step: This part of the algorithm is identical for all ensemble-based approaches and uses the forecast model (1) on each analysis member $\boldsymbol{X}_{k-1|k-1}^m$, with the integration length determined by the time separation between observations. The resulting sample $\{\boldsymbol{X}_{k|k-1}^m\}_{m=1}^M$ represents an unbiased approximation of the prior PDF $P(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ and will be utilized in the estimation of the prior score $\hat{\mathbf{s}}_{k|k-1}$.

Update step: The main goal here is to obtain an approximation for the posterior score $\mathbf{s}_{k|k}$ (i.e., $\hat{\mathbf{s}}_{k|k}$). The Bayesian update formula in Eq. (4) inspired us to propose the following form for the posterior score $\mathbf{s}_{k|k}(\mathbf{z}, t)$:

$$\mathbf{s}_{k|k}(\mathbf{z}, t) := \mathbf{s}_{k|k-1}(\mathbf{z}, t) + h(t)\nabla_{\mathbf{x}} \log p(\mathbf{y}_k|\mathbf{z}). \tag{10}$$

Note that the coefficient $h(t)$ multiplying the likelihood score represents a damping factor that decreases over the pseudo-time interval $[0, T]$ such that $h(0) = 1$ and $h(T) = 0$. In our numerical experiments, we define $h(t) = T - t$, although other options are also possible and will be explored in future work. Unlike existing diffusion models that use neural networks to learn the score function, we first derive the closed-form representation of the score function as follows:

$$\begin{aligned}\mathbf{s}(\mathbf{z}_t, t) =& \nabla_{\mathbf{z}} \log Q(\mathbf{z}_t) = \nabla_{\mathbf{z}} \log \left(\int Q(\mathbf{z}_t|\mathbf{z}_0)Q(\mathbf{z}_0)d\mathbf{z}_0 \right) \\ =& \frac{1}{\int Q(\mathbf{z}_t|\mathbf{z}_0')Q(\mathbf{z}_0')d\mathbf{z}_0'} \int -\frac{\mathbf{z}_t - \alpha_t \mathbf{z}_0}{\beta_t^2} Q(\mathbf{z}_t|\mathbf{z}_0)Q(\mathbf{z}_0)d\,\mathbf{z}_0 \\ =& -\int \frac{\mathbf{z}_t - \alpha_t \mathbf{z}_0}{\beta_t^2} \mathbf{w}_t(\mathbf{z}_t, \mathbf{z}_0)Q(\mathbf{z}_0)d\mathbf{z}_0.\end{aligned} \tag{11}$$

Notice that the weight function $\mathbf{w}_t(\mathbf{z}_t, \mathbf{z}_0)$ follows the definition

$$\mathbf{w}_t(\mathbf{z}_t, \mathbf{z}_0) = \frac{Q(\mathbf{z}_t|\mathbf{z}_0)}{\int Q(\mathbf{z}_t|\mathbf{z}_0')Q(\mathbf{z}_0')d\mathbf{z}_0'}, \tag{12}$$

and satisfies the condition $\int \mathbf{w}_t(\mathbf{z}_t, \mathbf{z}_0)Q(\mathbf{z}_0)d\mathbf{z}_0 = 1$. In our implementation, we use Monte Carlo estimator to approximate the integrals in Eq. (11), such that we can completely avoid the training process for existing diffusion models.

3) Scalable implementation of EnSF on HPC: We have implemented the EnSF method in PyTorch, making the code base compatible with both CPU-based platforms and those equipped with accelerators. The computational workload scales with various factors, including the number of ensembles, problem dimensions, and the total number of filtering cycles. The most efficient factor for parallelization are the ensembles, as it incurs minimal communication overhead. Considering the large memory capacity of GPUs on Frontier, straightforward parallelization can already support EnSF with dimensions up to 100 million, which is more than sufficient for our application. Since the training of the ViT surrogate is the bottleneck of the overall scaling, we will focus on the optimization of distributed training in the following.

B. *ViT surrogate for the SQG model*

a) **Compute-efficient architecture:** We have developed a Vision Transformer (ViT) surrogate tailored specifically for the surface quasi-geostrophic (SQG) model, utilizing a standard ViT backbone. Figure 2 illustrates the architecture of SQG-ViT, which consists of multi-head self-attention and multi-layer perceptron (MLP) components, augmented by normalization layers before and after the attention mechanism. To address overfitting, we have incorporated Dropout and DropPath regularization techniques. It is worth noting that the MLP component typically dominates the parameter count, making matrix-matrix multiplication (GEMM) the most computationally intensive operation.

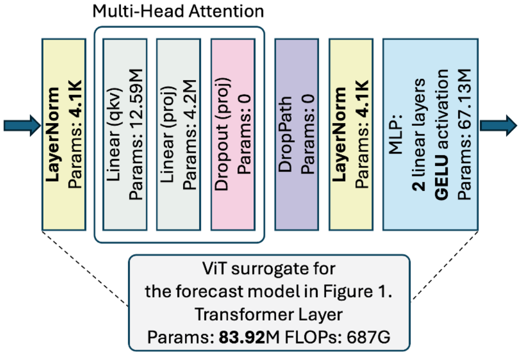

Fig. 2. Building block of ViT surrogate model for the forecast model in Figure 1. The number of parameters and floating point operations (FLOPs) are exemplified with 8-head attention, an embedding dimension of 2048, and a MLP to attention ratio of 8.

The performance of GEMM is significantly influenced by the shapes of the matrices [25], [26], thereby impacting the overall training efficiency of ViT. This dependency underscores the importance of appropriately sizing kernels, a task determined by factors such as embedding dimension, number of attention heads, and the ratio of MLP to attention. Adhering to the scaling law for Transformer architecture, where model capacity scales with the number of parameters, optimizing kernel sizes for computational efficiency becomes imperative for large-scale training on HPC systems. In the following section, we describe our distributed training strategies.

Method	optimizer	optimizer gradient	optimizer gradient weight	hierarchical
FSDP	n/a	shard_grad_op	full_shard	hybrid_shard
ZeRO	stage 1	stage 2	stage 3	n/a

TABLE I
THE DISTRIBUTED TRAINING METHODS WITH DIFFERENT MEMORY PARTITION STRATEGIES.

b) **Fully sharded data parallel (FSDP):** In addition to conventional data parallelism, where each device hosts a duplicate of the model, recent advancements in memory-efficient data parallelism, such as FSDP, have emerged as more suitable options for training large models due to their reduced memory footprint. Even when utilizing half precision, Vision Transformer (ViT) training necessitates approximately 12 times the model parameter size in memory storage, encompassing model weights (1X), optimizer states (2X for Adam optimizer), gradients (1X), and intermediate storage (2X) like FSDP units. FSDP offers distributed partitioning of various memory components through three strategies outlined in Table I. Specifically, `shard_grad_op` distributes gradients and optimizer states across all devices, `full_shard` partitions all memory components, and `hybrid_shard` represents a blend of data parallelism and FSDP.

c) **ZeRO data parallel:** Besides PyTorch built-in FSDP, another widely utilized memory-efficient data parallel implementation is DeepSpeed ZeRO. These two strategies exhibit an almost one-to-one correspondence (refer to Table I). However, ZeRO offers a broader array of tuning parameters for performance optimization compared to FSDP. These include adjusting the message bucket size for operations like `AllGather` and `Reduce`, enabling continuous memory allocation for gradients, and other similar optimizations.

d) **Computational budget estimation:** The total number, T, of floating-point operations (FLOPs) required for training ViT is directly proportional to the number of tokens, which depends on factors such as input size (L), patch size (P), number of epochs (E), and the number of model parameters (M). Specifically, this relationship follows $T = 6\prod_{i=1}^{d}\frac{L_i}{P_i} * E * M$, where d represents the dimension of the input image. The number of tokens per input image is given by the product, and hence T is essentially proportional to the total number of tokens during the training and the number of model parameters. The factor 6 comes from the fact that every token is processed with a multiply-accumulate (MAC) and two MACs during the forward and backward propagation, respectively. In Figure 3, we present the total number of FLOPs and the computation hours (in the unit of Frontier node hours) needed to train three representative sizes of ViT. Without loss of generalizability, we assume training over 100 epochs with a dataset containing 1 million images.

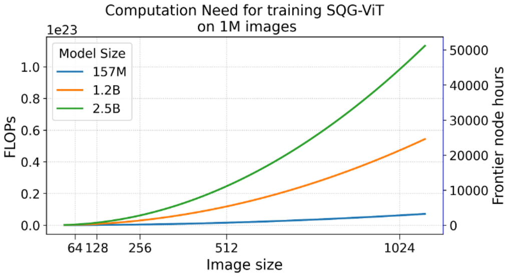

Fig. 3. Computation need in terms of FLOPs and Frontier node hours for training ViT surrogate model for the SQG model on 1M images.

IV. Results

We perform the experiments on the first Exascale supercomputer, Frontier. Each Frontier node is equipped with four AMD Instinct MI250X GPUs with dual Graphics Compute Dies (GCDs). A GCD is viewed as an effective GPU, and we use GCD and GPU interchangeably in the following discussion. All four MI250Xs (eight effective GPUs) are connected using 100 GB/s Infinity Fabric (200 GB/s between 2 GCDs of MI250X). We report the following two sets of experimental results (the code is publicly available[1]):

- **Accuracy tests**: Comparing our method with the state-of-the-art LETKF method to demonstrate the superior accuracy of our method in predicting highly nonlinear turbulent dynamics.
- **Scalability tests**: Demonstrating the scalability of the proposed real-time DA workflow in Figure 1, including the online ViT training and the online EnSF execution.

A. Accuracy tests

a) **Experimental setup:** For our numerical tests, we discretize the SQG model on a 64x64x2 mesh and evaluate the errors of different DA systems in the setting where the entire SQG state is directly observed; that is, the observation operation $\mathbf{h}_k$ in (2) becomes the identity matrix $\mathbf{I}$. For simplicity, the error covariance matrix $\mathbf{R}$ is also set to $\mathbf{I}$. Observations are generated synthetically every 12h within a standard observation system simulation (OSSE) framework [27]. We compared with accuracy of our method with the state-of-the-art LETKF method, which was originally proposed by Bishop et al. [28] and further developed in Hunt et al. [29].

The ensemble size for both DA algorithms (LETKF and EnSF) is set to 20. Initial ensembles are created through the random selection of model states from a long-term integration of the SQG model. Since the external model errors discussed earlier are unpredictable, LETKF's inflation and localization parameters are tuned in an error-free twin experiment. We find that the optimal RTPS factor and cut-off localization scales are 0.3 and 2000 km, respectively. One significant advantage of the EnSF algorithm used in our new DA framework is that it can maintain stable performance without any special tuning. For the numerical tests presented in this study, localization is not applied and the variance (spread) of the analysis ensemble is simply relaxed to the prior (forecast) values in order to guarantee the long-term filtering stability.

We compare the performance over the time period $t \in [0, 3600]$ and consider the four different architectures:

- **SQG only**: Run the SQG model iteratively from $t = 0$ to $t = 3600$ without incorporating observations.
- **ViT only**: Run the offline trained ViT surrogate iteratively from $t = 0$ to $t = 3600$ without using observations.
- **SQG + LETKF**: Apply LETKF (a SOTA method in the DA community) to assimilate observations and correct the SQG forecasts.
- **ViT + EnSF**: This is the proposed framework in this study – use the more accurate EnSF method to adjust the forecasts from the pre-trained ViT surrogate of the true SQG dynamics.

Figure 4 shows the root mean squared error (RMSE) of the above four experiments. We can make several important observations. First, DA is a necessary component to ensure accurate long-term reconstruction of the SQG state. This to be contrasted with the SQG-only and ViT-only experiments where the RMSEs experience a rapid growth as a result of the developing SQG turbulence. This is caused by the chaotic dynamics and the rapid amplification of IC errors. Second, the LETKF RMSEs gradually increase as we add model errors to true SQG state. Eventually, the LETKF's performance is comparable to the SQG-only and ViT-only simulations in which DA is not carried out. The latter implies that the SOTA LETKF method is sensitive to model imperfections even when

[1] https://github.com/jqyin/sqg_vit

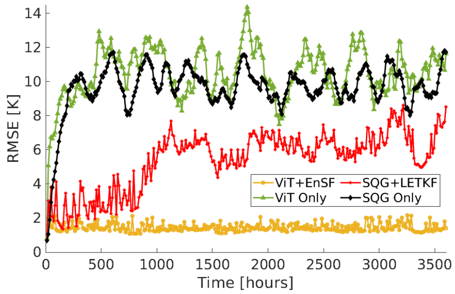

Fig. 4. The root mean squared errors (RMSEs) of the four test cases. We observe that data assimilation is a necessary component to ensure accurate reconstruction of the SQG state. On the other hand, the RMSE of experiments that only use SQG or ViT without a DA component grows very fast in time. Moreover, LETKF diverges from the ground truth as model errors accumulate in time, suggesting that the LETKF method is sensitive to model imperfections. The proposed EnSF+ViT framework provides superior performance since we observe stable performance throughout all analysis cycles even in the absence of fine tuning.

the inflation and localization parameters are optimally tuned. Third, EnSF+ViT provides superior performance – we observe stable results throughout the entire integration period without any special fine tuning.

B. Scalability tests

We investigate the scaling of the proposed DA framework, i.e., the ViT+EnSF workflow, on Frontier from the compute-efficient architecture search on single node, to performance analysis and profiling, and optimization at scale.

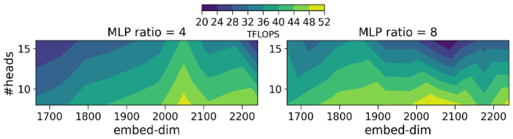

Fig. 5. Computation performance (TFLOPS) heatmap for the ViT surrogate's architecture on Frontier.

a) **Compute-efficient architecture:** As shown in Figure 5, the single-node training performance of 256^2 inputs varies from 20 TFLOPS to 52 TFLOPS, mostly depending on the embedding dimensions, the number of attention heads, and the MLP ratio (i.e., the percentage of MLP parameters of a ViT layer). Typically, higher number of attention heads reduce the performance, and a embedding dimension of 2048 provides the best performance. Increasing the weight of MLP operations will improve the performance overall.

We design our scaling experiments for three input and model sizes, with detailed architectures listed in Table II. The number of parameters ranges from 157M to 2.5B. While the number of attention heads is fixed at 8, the embedding dimension increases from 1024 to 2048, to provide more capacity for larger inputs. The number of layers is doubled from each size as well.

To study the performance bottleneck, we profile the runtime of the ViT training at 1024 GPUs on Frontier for all three model and input sizes. As shown in Figure 6, the runtime breakdown indicates the training is dominated by computation and communication, with negligible IO, although the IO portion increases slightly from small input (64^2) to large input (256^2). Specifically, for 64^2, the computation is less intensive (hence takes longer runtime) compared to larger models due to the 1024 embedding size, and yet the portion of communication is still larger than that of 128^2, indicating a slower training performance. On the other hand, for 256^2, the computation workload is twice of 128^2, but the communication takes a larger portion because the message volume also doubles. Our results show that ViT training is mostly communication bound at scale, especially for large inputs (i.e., longer sequences).

input	patch	#layers	#heads	#embed dim	#mlp ratio	#params
64^2	4	12	8	1024	4	157M
128^2	4	24	8	2048	4	1.2B
256^2	4	48	8	2048	4	2.5B

TABLE II
THE ARCHITECTURE OF THE VIT SURROGATE MODELS.

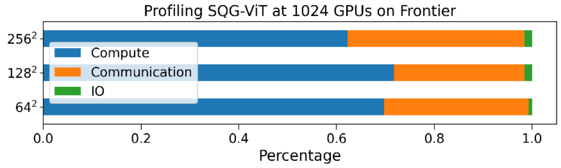

Fig. 6. The runtime percentage of computation, communication and IO for training the ViT surrogate model with input size of 64^2, 128^2, and 256^2, respectively.

b) **Scaling on Frontier:** With the profiling analysis and baselines established, we are ready to compare different distribution strategies and scale the ViT surrogate up to 1024 GPUs on Frontier. In Figure 7, we first compare the scaling of different model and input sizes. 128^2 performs the best with a scaling efficiency of 86%, while 64^2 and 256^2 performs comparably. This is consistent with the profiling analysis (see Figure 6), which indicates a trade-off between the computation intensity and communication volume, and 128^2 input with a 1.2B model size seem to be optimal on Frontier.

However, for our scientific application, a larger input is desired. To improve the performance of 256^2, we further study different memory-efficient data-parallel strategies. As shown in Figure 7, the DeepSpeed stage 1 with default setting (message bucket size 200MB) in PyTorch lightning does not perform well because the communication bandwidth of `AllReduce` deteriorates around this message size. On the other hand, a very large message size will not work well either due to less opportunities to overlap communication with computation. We find a message size around 500MB works the best, and resulted scaling efficiency improves to 85%. Overall, with more optimization knobs, DeeSpeed ZeRO data-parallel outperforms FSDP for training SQG-ViT on Frontier.

c) **EnSF scaling:** With the training of the forward model optimized, we study the scaling behavior of EnSF on

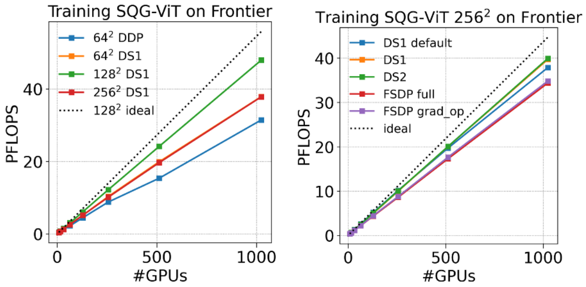

Fig. 7. Scaling ViT surrogate up to 1024 GPUs on Frontier with distributed data parallel (DDP), DeepSpeed (DS) stage 1 and 2, and fully sharded data parallel (FSDP) with full and grad_op strategies. The model size for 64^2, 128^2, and 256^2 input is 157M, 1.2B, and 2.5B, respectively.

Frontier. The MPI parallelization is along the dimension of the ensemble, so the ranks are straightforwardly parallel and the outputs are MPI reduced in the end. As shown in Figure 8, EnSF weak scales perfectly up to 1024 GPUs on Frontier. The time per step is about 0.4s for 1M dimension, and 28s for 100M.

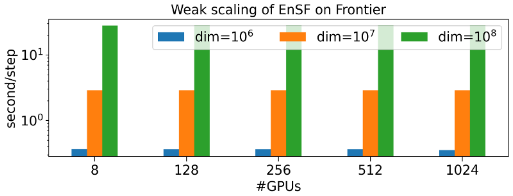

Fig. 8. Weak scaling of EnSF on Frontier up to 1024 GPUs for dimension size of 10^6, 10^7, and 10^8, respectively.

V. Conclusion

In this study, we introduce a generic sequential data assimilation framework for estimating turbulent dynamics and demonstrate its end-to-end performance on the Frontier supercomputer. By investigating compute-efficient kernel sizing and comparing various parallelization strategies, we achieve a 85% strong scaling efficiency and linear weak scaling up to 1024 GPUs, respectively, on the Frontier supercomputer. Our results demonstrate the framework's exceptional scalability on high-performance computing systems, which is essential for improving the medium-range forecasts of high-dimensional Earth system applications. As shown in the numerical experiment, e.g., Figure 4, physics-based or AI-based weather/climate models cannot predict turbulent dynamics without an efficient DA workflow. We emphasize that the proposed workflow can be combined with any physics-based or AI-based foundation weather models because of using the ViT surrogate.

Acknowledgement

This work is supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program, under the contract ERKJ388, and by Dan Lu's Early Career Project, sponsored by the Office of Biological and Environmental Research. ORNL is operated by UT-Battelle, LLC., for the U.S. Department of Energy under Contract DE-AC05-00OR22725. Feng Bao would also like to acknowledge the support from U.S. National Science Foundation through project DMS-2142672 and the support from the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Grant DE-SC0022297. This research used resources of the Oak Ridge Leadership Computing Facility (OLCF), which is a DOE Office of Science User Facility at the Oak Ridge National Laboratory supported by the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

[1] T. Miyoshi, A. Amemiya, S. Otsuka, Y. Maejima, J. Taylor, T. Honda, H. Tomita, S. Nishizawa, K. Sueki, T. Yamaura, Y. Ishikawa, S. Satoh, T. Ushio, K. Koike, and A. Uno, "Big data assimilation: Real-time 30-second-refresh heavy rain forecast using fugaku during tokyo olympics and paralympics," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '23, (New York, NY, USA), Association for Computing Machinery, 2023.

[2] T. Kurth, S. Subramanian, P. Harrington, J. Pathak, M. Mardani, D. Hall, A. Miele, K. Kashinath, and A. Anandkumar, "FourCastNet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators," in *Proceedings of the Platform for Advanced Scientific Computing Conference*, PASC '23, (New York, NY, USA), Association for Computing Machinery, 2023.

[3] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu, A. Merose, S. Hoyer, G. Holland, O. Vinyals, J. Stott, A. Pritzel, S. Mohamed, and P. Battaglia, "Learning skillful medium-range global weather forecasting," *Science*, vol. 382, no. 6677, pp. 1416–1421, 2023.

[4] T. Nguyen, J. Brandstetter, A. Kapoor, J. K. Gupta, and A. Grover, "Climax: A foundation model for weather and climate," *arXiv preprint arXiv:2301.10343*, 2023.

[5] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, "Accurate medium-range global weather forecasting with 3d neural networks," *Nature*, vol. 619, pp. 533–538, July 2023.

[6] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *Journal of Geophysical Research*, vol. 99, pp. 10143–10162, 1994.

[7] J. Poterjoy, "Implications of multivariate non-Gaussian data assimilation for multi-scale weather prediction," *Monthly Weather Review*, vol. 150, pp. 1475–1493, 2022.

[8] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Proc. Inst. Elect. Eng. F*, vol. 1400, p. 107–113, 1993.

[9] P. J. van Leeuwen, "Particle filtering in geophysical systems," *Monthly Weather Review*, vol. 137, p. 4089–4114, 2009.

[10] P. J. van Leeuwen, H. R. Künsch, L. Nerger, R. Potthast, and S. Reich, "Particle filters for high-dimensional geoscience applications: a review," *Quarterly Journal of the Royal Meteorological Society*, vol. 145, pp. 2335–2365, 2019.

[11] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Transactions on signal processing*, vol. 50, pp. 736–746, 2002.

[12] F. Bao, Z. Zhang, and G. Zhang, "A score-based nonlinear filter for data assimilation," *Journal of Computational Physics*, vol. 514, p. 113207, 2024.

[13] F. Bao, Z. Zhang, and G. Zhang, "An ensemble score filter for tracking high-dimensional nonlinear dynamical systems," *ArXiv:2309.00983*, pp. 1–17, 2023.

[14] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.

[15] Y. Liu, M. Yang, Z. Zhang, F. Bao, Y. Cao, and G. Zhang, “Diffusion-model-assisted supervised learning of generative models for density estimation,” *Journal of Machine Learning for Modeling and Computing*, vol. 5, no. 1, pp. 25–38, 2024.

[16] M. Yang, P. Wang, D. Del-Castillo-Negrete, Y. Cao, and G. Zhang, “A pseudo-reversible normalizing flow for stochastic dynamical systems with various initial conditions,” *SIAM Journal on Scientific Computing*, vol. 46, no. 4, pp. C508–C533, 2023.

[17] M. Yang, P. Wang, M. Fan, D. Lu, Y. Cao, and G. Zhang, “Conditional pseudo-reversible normalizing flow for surrogate modeling in quantifying uncertainty propagation,” *ArXiv:2404.00502*, 2024.

[18] H. G. Chipilski, “Exact nonlinear state estimation,” *arXiv*, p. 1–31, 2023.

[19] R. Tulloch and K. S. Smith, “Quasigeostrophic turbulence with explicit surface dynamics: Application to the atmospheric energy spectrum,” *Journal of the Atmospheric Sciences*, vol. 66, pp. 450–467, 2009.

[20] R. Tulloch and K. S. Smith, “A note on the numerical presentation of surface dynamics in quasigeostrophic turbulence,” *Journal of the Atmospheric Sciences*, vol. 66, pp. 1063–1068, 2009.

[21] G. D. Nastrom and K. S. Gage, “A climatology of atmo- spheric wavenumber spectra of wind and temperature ob- served by comercial aircraft,” *Journal of the Atmospheric Sciences*, vol. 42, pp. 950–960, 1985.

[22] D. R. Durran and M. Gingrich, “Atmospheric predictability: why butterflies are not of practical importance,” *Journal of the Atmospheric Sciences*, vol. 71, p. 2476–2488, 2014.

[23] P. E. Kloeden and E. Platen, *Numerical solution of stochastic differential equations*, vol. 23 of *Applications of Mathematics (New York)*. Berlin: Springer-Verlag, 1992.

[24] F. Bao, Y. Cao, A. Meir, and W. Zhao, “A first order scheme for backward doubly stochastic differential equations,” *SIAM/ASA J. Uncertain. Quantif.*, vol. 4, no. 1, pp. 413–445, 2016.

[25] J. Yin, A. Tsaris, S. Dash, R. Miller, F. Wang, and M. A. Shankar, “Comparative evaluation of deep learning workloads for leadership-class systems,” *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, vol. 1, no. 1, p. 100005, 2021.

[26] Q. Anthony, J. Hatef, D. Narayanan, S. Biderman, S. Bekman, J. Yin, A. Shafi, H. Subramoni, and D. Panda, “The case for co-designing model architectures with hardware,” 2024.

[27] R. N. Hoffman and R. Atlas, “Future observing system simulation experiments,” *Bulletin of the American Meteorological Society*, vol. 97, p. 1601–1616, 2016.

[28] C. H. Bishop, B. J. Etherton, and S. J. Majumdar, “Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects,” *Monthly Weather Review*, vol. 129, p. 420–436, 2001.

[29] B. R. Hunt, E. J. Kostelich, and I. Szunyogh, “Efficient data assimilation for spatiotemporal chaos: A local ensemble transform kalman filter,” *Physica D: Nonlinear Phenomena*, vol. 230, pp. 112–126, 2007.