Local volume solvers for Earth system data assimilation: implementation in the framework for Joint Effort for Data Assimilation Integration

Sergey Frolov*, Anna Shlyaeva, Wei Huang, Travis Sluka, Clara Draper, Bo Huang, Kriti Bhargava, Jeff Whitaker

Corresponding Author: Sergey Frolov (Sergey.Frolov@noaa.gov), NOAA Physical Sciences Laboratory, 325 Broadway, Boulder, CO 80305-3337.

- Anna Shlyaeva: annash@ucar.edu, Joint Center for Satellite Data Assimilation, Boulder, CO.
- Wei Huang: wei.huang@noaa.gov, Cooperative Institute for Research in Environmental Sciences, NOAA Physical Sciences Laboratory, Boulder, CO.
- Travis Sluka: tsluka@ucar.edu, Joint Center for Satellite Data Assimilation, Boulder, CO.
- Clara Draper: clara.draper@noaa.gov, NOAA Physical Sciences Laboratory, Boulder, CO.
- Bo Huang: bo.huang@noaa.gov, Cooperative Institute for Research in Environmental Sciences, Boulder, CO.
- Kriti Bhargava: kritib@ucar.edu, Joint Center for Satellite Data Assimilation, Boulder, CO.
- Jeff Whitaker: jeffrey.s.whitaker@noaa.gov, NOAA Physical Sciences Laboratory, Boulder, CO.

Key points:

(1) Local volume solvers are developed in the Joint Effort for Data assimilation Integration (JEDI) software.
(2) Localization operators for various components of the Earth system are developed.
(3) Case studies are presented with various components of the Earth system.

Abstract

The Joint Effort for Data assimilation Integration (JEDI) is an international collaboration aimed at developing an open software ecosystem for model agnostic data assimilation. This paper considers implementation of the model-agnostic family of the local volume solvers in the JEDI framework. The implemented solvers include the Local Ensemble Transform Kalman Filter (LETKF), the Gain form Ensemble Transform Kalman Filter (GETKF), and the optimal interpolation variant of the LETKF filter (LETKF-OI). This paper documents the implementation choices and strategies that allow model agnostic implementation. We also document an expansive set of localization approaches that includes generic distance-based localization, localization based on modulated ensemble products, but also localizations specific to ocean (based on the Rossby radius of deformation), and land (based on the terrain difference between observation and model grid point). Finally, we apply the developed solvers in a limited set of experiments, including single-observation experiments in atmosphere and ocean, and cycling experiments for the ocean, land, and aerosol assimilation. We also provide a proof of concept that illustrates how JEDI Ensemble Kalman Filter solvers can be used in a strongly coupled framework providing increments to the ocean based on the combined observations from the ocean and the atmosphere.

Plain Language Summary

The Joint Effort for Data assimilation Integration (JEDI) is an international collaboration aimed at reducing time it takes to transition research on initialization of the Earth system models to operation. JEDI framework is designed to be agnostic of the specific numerical model and, hence, can facilitate collaboration between research

institutions and operational centers. This paper documents implementation of the Ensemble Kalman filtering framework within JEDI. The implementation strategy supports a variety of algorithmic approaches to the Ensemble Kalman Filtering and is appropriate for multiple Earth system applications. Specifically, we demonstrate applications for atmosphere, atmospheric composition, ocean, and land data assimilation.

1 Introduction

Data assimilation for multiple Earth system components simultaneously (coupled data assimilation) is an emerging field, where observations of each Earth system component (such as atmosphere, ocean, ice, land, and aerosols) are combined with a coupled simulation model to produce a statistically consistent estimate of the complete Earth system (Penny et al., 2017). Recent examples of the coupled data assimilation (DA) applications build on a rich history of data assimilation in each sub-component of the system. For example, the National Oceanic and Atmospheric Administration (NOAA) (Saha et al., 2006, 2010), European Centre for Medium-Range Weather Forecasts (ECMWF) (Browne et al., 2019; Laloyaux et al., 2015), and Naval Research Laboratory (NRL) (Barton et al., 2020) operational Earth system forecast models and reanalyses are composed from legacy DA systems (solvers) for the atmosphere, ocean, and ice. These legacy DA solvers are specific to each Earth-system component and present formidable technical barriers for developing strongly coupled DA systems that treat all Earth system components as a joint statistical estimation problem in which observations of one component can correct the state estimate in the coupled components.

The Joint Effort for Data assimilation Integration (JEDI) is an international collaboration aimed at developing an open software ecosystem for model agnostic data

assimilation. Earlier efforts that developed model-agnostic data assimilation infrastructure such as Data Assimilation Research Testbed (DART Anderson et al., 2009) and Parallel Data Assimilation Framework (PDAF Nerger & Hiller, 2013) focused on acceleration of research in DA. In contrast, JEDI is developed by a consortium of operational and research weather forecast centers led by the Joint Center for Satellite Data Assimilation in Boulder, Colorado with the goal of accelerating transition of research results to operations. The JEDI framework achieves this by separating the concerns of the developers into generic blocks (see Figure 1) centered on abstract modeling of the data assimilation problem, observational storage, covariance modeling, observation operator, and interfaces to forecast models (grouped inside of the gray box). These basic building blocks in Figure 1 can be reused and specialized by multiple models (orange boxes in Figure 1). Some of these models include atmospheric models like the Finite Volume cube sphere version 3 atmosphere general circulation model—FV3 (Harris et al., 2021), and the Model for Prediction Across Scales—MPAS (Skamarock et al., 2012); ocean models like the Modular Ocean Model version 6—MOM6 (Adcroft et al., 2019) and the Regional Ocean Model—ROMS (Haidvogel et al., 2000); sea ice models like the Los Alamos sea ice model—CICE (Hunke & Lipscomb, 2015); land surface models like the Noah land surface model with Multi-Parameterization options—Noah-MP (Niu et al., 2011), and atmospheric composition models like the FV3-based Global Ensemble Forecast System – Aerosols (GEFS-Aerosols; Zhang et al., 2021). Consequently, the generic DA algorithms developed using the JEDI framework can be applied to many of the model implementations currently under development within the JEDI consortium. Having this ability to use the same codebase, data structures, and

algorithms for all components of the Earth system model will greatly simplify the transition to strongly coupled data assimilation and testing of the individual components.

This paper focuses on the implementation of local volume solvers within the JEDI framework. By local volume solvers, we mean the large family of algorithms that updates the model forecast at each grid point (e.g. the green square in Figure 2) using a selection of observations in the geographic neighborhood of this grid point (e.g. the orange dots and orange circle in Figure 2). Such updates can be performed independently of each other and can scale well on modern computer architectures. The family of local volume solvers includes the Optimal Interpolation (OI) (Gandin, 1963), the Local Ensemble Transform Kalman Filter (LETKF) (Hunt et al., 2007), the Gain form Ensemble Transform Kalman Filter (GETKF) (Bishop et al., 2017; Lei et al., 2018), and LETKF-OI/GETKF-OI (Frolov et al., 2022) algorithms. Local volume updates are in contrast to global updates that are achieved by inverting global covariance matrices through a gradient descent algorithm employed by variational solvers (Daley, 1991).

Specifically, this paper presents implementation of the local volume solvers in the abstract layer, implementation choices for model data and observational data distribution across processing elements, and a flexible framework for representation of the ensemble localization. We also provide details on the specific implementations of the localization strategies for different components and between the components of the Earth system. We demonstrate our implementation in various components of the Earth system model using a combination of single-observation increments and a limited demonstration of cycling capabilities.

The objective of this paper is to make the framework for local volume solvers in JEDI accessible to end-users and developers. For the initial set of capabilities described in this paper, we implemented the most basic solvers (LETKF and GETKF) and several localization options (observation space for the LETKF and model space for the GETKF). This paper documents these basic capabilities to engage the end-users in rigorous testing and refinement of the existing capabilities. We also aim to document existing interfaces that, we believe, will support implementation of more advanced capabilities by external developers, including hybrid solvers (Kotsuki & Bishop, 2021), local particle filters (Poterjoy, 2016), and iterative versions of the ensemble filters (Bocquet & Sakov, 2014).

2 Mathematical notation for the local volume solver

2.1 Ensemble Kalman filter

A generic implementation of an ensemble KF (EnKF) consists of the time update and the measurement update steps (Evensen, 2003). In the time update step, ensemble initial conditions are propagated forward in time and the mismatches between actual observations and the ensemble observation equivalents are computed, resulting in a prior estimate (the forecast) of the state and observations:

$$\begin{cases} (a): \mathbf{x}_j^{f(k+1)} = \mathcal{M}\left(\mathbf{x}_j^{a(k)}\right) + \boldsymbol{\eta}_j^{(k+1)} \\ (b): \mathbf{y}_j^{f(k+1)} = \mathcal{H}\left(\mathbf{x}_j^{f(k+1)}\right) + \boldsymbol{\zeta}^{(k+1)} \end{cases} \quad (1),$$

Where: $\mathbf{x}_j^{f(k)}$ and $\mathbf{x}_j^{a(k)}$ are the forecast and analysis states for member (j) valid for timestep (k), $\mathcal{M}$ and $\mathcal{H}$ are the forecast and the observation operators; $\boldsymbol{\eta} \leftarrow N(\mathbf{0}, \mathbf{Q})$ and $\xi \leftarrow N(\mathbf{0}, \mathbf{R})$ are the model and the observational noise terms; and $\mathbf{y}_j^{f(k)}$ are the observation equivalents of the ensemble member $\mathbf{x}_j^{f(k)}$.

From the prior forecast of observation equivalents $\mathbf{y}_j^{f(k+1)}$ an innovation vector is computed as:

$$\mathbf{y}^{innov(k+1)} = \mathbf{y}^{observ(k+1)} - \overline{\mathbf{y}_j^{f(k+1)}} \quad (2),$$

where $\mathbf{y}^{observ(k+1)}$, $\overline{\mathbf{y}_j^{f(k+1)}}$, and $\overline{\mathbf{y}^{innov(k+1)}}$ are the observations of nature, the average of ensemble observation equivalents, and the innovations at time (k+1).

From the time update in Eqs. (1-2), we can also compose matrices of ensemble perturbations in the state and observational space.

$$\mathbf{Z}^{f(k+1)} = \mathbf{X}^{f(k+1)} - \overline{\mathbf{x}_j^{f(k+1)}} \quad (3) \text{ and}$$

$$\mathbf{Y}^{f(k+1)} = \left[\mathbf{y}_1^{f(k+1)}, \ldots, \mathbf{y}_{N_{ens}}^{f(k+1)}\right] - \overline{\mathbf{y}_j^{f(k+1)}} \quad (4),$$

where $\mathbf{X}^{f(k)}$, $\mathbf{Z}^{f(k)}$, and $\mathbf{Y}^{f(k)}$ are the matrices holding the ensemble of forecasts, ensemble of forecast perturbations, and ensemble of forecast perturbations in the observational space, and N_{ens} is the ensemble size. The average of the forecast ensemble members in the state space and observational space are defined as $\overline{\mathbf{x}_j^{f(k+1)}}$ and $\overline{\mathbf{y}_j^{f(k+1)}}$ respectively.

In the measurement update step of the ensemble Kalman Filter, a posterior estimate of the ensemble members is computed through a linear combination of forecast (prior) ensemble members:

$$\begin{cases} (a): \overline{\mathbf{x}^{a(k+1)}} = \overline{\mathbf{x}^{f(k+1)}} + \mathbf{Z}^{f(k+1)} \mathbf{W}^{(k+1)}_{\{\mathbf{Y}^{f(k+1)}, \mathbf{R}, \mathbf{y}^{innov(k+1)}\}} \\ (b): \mathbf{X}^{a(k+1)} = \overline{\mathbf{x}^{a(k+1)}} + \mathbf{Z}^{f(k+1)} \mathbf{W}^{(k+1)}_{\{\mathbf{Y}^{f(k+1)}, \mathbf{R}\}} \end{cases} \quad (5)$$

Where $\mathbf{w}^{(k+1)}_{\{\mathbf{Y}^{f(k+1)}, \mathbf{R}, \mathbf{y}^{innov(k+1)}\}}$ and $\mathbf{W}^{(k+1)}_{\{\mathbf{Y}^{f(k+1)}, \mathbf{R}\}}$ are the ensemble transformation vector and matrix for the mean state and the ensemble perturbations, respectively. Specific forms of

the transformation matrices depend on the update algorithm and are discussed in section 3 below. The subscripts $_{\{\mathbf{Y}^{f(k+1)},\mathbf{R},\mathbf{y}^{innov(k+1)}\}}$ and $_{\{\mathbf{Y}^{f(k+1)},\mathbf{R}\}}$ indicate that the weights are functions of the observation perturbations, observation errors, and innovations.

The original implementation of the EnKF filters in JEDI is based on the square-root form of the EnKF, where the ensemble of analyzed states $\mathbf{X}^{a(k+1)}$ is obtained using a single transformation matrix ($\mathbf{W}^{(k+1)}_{\{\mathbf{Y}^{f(k+1)},\mathbf{R}\}}$) and a single mean innovation $\mathbf{y}^{innov(k)}$. This is in contrast to alternative formulations of the EnKF that use perturbed observations (Evensen, 2004) or use Kalman gain matrices that differ for each ensemble member (Buehner, 2020). Such implementations are possible in the JEDI local volume solver framework; however, they were not implemented at the time this paper was written.

2.2 Local volume solver approximation

In the context of the local volume solver, the measurement update step is performed for each local collection of model states $\mathbf{x}_{(i)}$ as follows. First a local innovation vector is computed by applying the selection operator $\mathbf{S}^{observ}_{(i)}$ (orange circle in Figure 2) that selects innovations close to the grid point (i):

$$\mathbf{y}^{innov(k+1)}_{local(i)} = \mathbf{S}^{observ}_{(i)}\mathbf{y}^{innov(k+1)} \quad (6).$$

Then, if R-localization is used (Hunt et al., 2007), the observation error covariance $\mathbf{R}$ is inflated as a function of the distance between the location of the update point loc(i) and the observation locations loc(y):

$$\mathbf{R}_{local} = \left(\mathbf{\Psi}_{\|loc(i)-loc(y)\|}\mathbf{S}^{observ}_{(i)}\right)\mathbf{R}\left(\mathbf{\Psi}_{\|loc(i)-loc(y)\|}\mathbf{S}^{observ}_{(i)}\right)^{T} \quad (7),$$

where $\mathbf{\Psi}_{\|loc(i)-loc(y)\|}$ is the localization value computed as a function of the distance $\|loc(i)-loc(y)\|$ between observation location *loc(y)* and analysis location *loc(i)*. We discuss

different forms of the localization function in section 5.1. Note that the R-localization in (Hunt et al., 2007) only applies to diagonal matrix R in Eq. (7). An alternative to the R-localization in Eq. (7) is Z-localization (Kotsuki & Bishop, 2022), which attenuates columns of the $\mathbf{Y}$ matrix and can be used in presence of the non-diagonal R matrix. Since Z localization is not yet implemented in JEDI, we will use observation localization and R-localization interchangeably referring to Eq. (7) above.

Lastly, the local update is calculated using local versions of the innovations $\mathbf{y}_{local(i)}^{innov(k+1)}$ and the inflated observational errors $\mathbf{R}_{local}$:

$$\begin{cases} \overline{\mathbf{x}_{(i)}^{a(k+1)}} = \overline{\mathbf{x}_{(i)}^{f(k+1)}} + \mathbf{Z}_{(i)}^{f(k+1)} \mathbf{W}_{\{\mathbf{Y}_{(i)}^{f(k+1)}, \mathbf{R}_{local}, \mathbf{y}_{(i)}^{innov(k+1)}\}}^{(k+1)} \\ \mathbf{X}_{(i)}^{a(k+1)} = \overline{\mathbf{x}_{(i)}^{a(k+1)}} + \mathbf{Z}_{(i)}^{f(k+1)} \mathbf{W}_{\{\mathbf{Y}_{(i)}^{f(k+1)}, \mathbf{R}_{local}\}}^{(k+1)} \end{cases} \quad (8),$$

where $\mathbf{x}_{(i)} = \mathbf{S}_{(i)}^{state} \mathbf{x}$ are the variables local to the grid points (i) selected using model-state selection operator $\mathbf{S}_{(i)}^{state}$. Examples of model-state selection operator $\mathbf{S}_{(i)}^{state}$ include: all model variables associated with a single vertical column (based on the horizontal distance, i.e. the 2D iterator in JEDI notation); or all model variables associated with a single 3D grid point (based on horizontal and vertical distance, i.e. the 3D iterator in JEDI notation). In principle, one may choose to select local points in more complex ways, for example, only temperature variables located within the oceanic mixed layer.

To simplify notation, time indices (k) and (k+1) will be dropped going forward, as these indices are obvious outside of the context of the time update step in Eq. (1).

3 Generic solvers

At the time of writing, there were two basic solvers implemented in JEDI: the LETKF that uses observation space localization (R-localization) and the GETKF that

uses a combination of the model space localization (B-localization) in the vertical direction and observation space localization (R-localization) in the horizontal.

3.1 LETKF

The LETKF filter (Hunt et al., 2007) can be written (and is implemented in JEDI) as a two-step operation. First, the local update weights $\mathbf{w}_{\{\mathbf{Y}_{(i)}^{f},\mathbf{R}_{local},\mathbf{y}_{(i)}^{innov}\}}$ and $\mathbf{W}_{\{\mathbf{Y}_{(i)}^{f},\mathbf{R}_{local}\}}$ are computed and, second, the update weights are applied to rotate, scale, and translate the local prior perturbations $\mathbf{Z}_{(i)}^{f}$. The weight computation requires an inverse of the N_{ens}*N_{ens} matrix. This is performed using eigen solver as follows:

$$\mathbf{A}_{vec}\mathbf{A}_{val}\mathbf{A}_{vec}^{T} \underset{eigen}{\leftarrow} \mathbf{A} = \left(\mathbf{Y}_{local(i)}^{f}\right)^{T} \mathbf{R}_{local(i)}^{-1}\mathbf{Y}_{local(i)}^{f} + \frac{(N_{ens}-1)}{\rho}\mathbf{I} \quad (9).$$

Where $\mathbf{A}_{vec}$ and $\mathbf{A}_{val}$ are the eigen vectors and eigen values, $\mathbf{I}$ is the identity matrix, and ρ is the forgetting (inflation) factor (which will be discussed in section 4.1). We provide two options for this computation: one using a generic C++ Eigen library (Eigen, 2023), and one using direct calls to the LAPACK library (Anderson et al., 1999). After the eigen decomposition is performed, update weights are computed as follows:

$$\begin{aligned}\mathbf{W}_{LETKF} &= \sqrt{N_{ens}}\,\mathbf{A}_{vec}\left(\mathbf{A}_{val}\right)^{-1/2}\mathbf{A}_{vec}^{T} \\ \mathbf{w}_{LETKF} &= \mathbf{A}_{vec}\left(\mathbf{A}_{val}\right)^{-1}\mathbf{A}_{vec}^{T}\left(\mathbf{Y}_{local(i)}^{T}\mathbf{R}_{local(i)}^{-1}\mathbf{y}_{local(i)}^{innov(k+1)}\right)\end{aligned} \quad (10).$$

The update weights are then applied to the ensemble perturbations and the ensemble mean, giving the ensemble mean increment $\mathbf{x}_{(i)}^{inc}$ and the ensemble perturbations $\mathbf{X}_{(i)}^{inc}$:

$$\begin{aligned}\mathbf{x}_{(i)}^{inc} &= \mathbf{Z}_{(i)}^{f}\mathbf{w}_{LETKF} \\ \mathbf{X}_{(i)}^{inc} &= \mathbf{x}_{(i)}^{inc} + \mathbf{Z}_{(i)}^{f}\mathbf{W}_{LETKF}\end{aligned} \quad (11).$$

3.2 GETKF

The JEDI implementation of GETKF follows Lei et.al (2018), with some minor changes as described below. The GETKF solver initializes a vertical localization matrix $\mathbf{L}_{vert}$ consisting of the N_{eig} columns $\mathbf{l}_{j-vert}$ of the square root of the vertical localization $\mathbf{C}_{vert}$.

$$\mathbf{C}_{vert} = \mathbf{L}_{vert}\mathbf{L}_{vert}^{T} \quad (12).$$

Vectors $\mathbf{l}_{j-vert}$ can either be conditioned on the background state or read from an external file.

The localization operator $\mathbf{L}_{vert}$ is used to compute the modulated ensemble $\mathbf{Z}^{\mathrm{mod}}$, by applying the Schur product $\circ$ between each column $\mathbf{l}_{\text{j-vert}}$ in L_{vert}, and each column z_j^f in Z^f:

$$\mathbf{Z}^{\mathrm{mod}} = \mathbf{L}_{vert} \triangle \mathbf{Z}^{f} = \left[\mathbf{l}_1 \circ \mathbf{z}^{f}_{1..n_{ens}}, \ldots, \mathbf{l}_{n_{eig}} \circ \mathbf{z}^{f}_{n_{ens}} \right] \quad (13).$$

The complete modulated ensemble in Eq. (13) is never computed. We either compute N_{eig} modulations of a single original ensemble member $\left[\mathbf{l}_1 \circ \mathbf{z}_j^f, \ldots, \mathbf{l}_{N_{eig}} \circ \mathbf{z}_j^f \right]$ during the oserver step, or all modulated ensemble members corresponding to a local geometry iterator $\mathbf{Z}^{\mathrm{mod}}_{(i)}$ during the measuremnt-update step.

The GETKF computes two observation perturbation matrices. First, identical to Eq. (4), we compute perturbations in the observational space for the original ensemble perturbations $\mathbf{Z}^f$:

$$\mathbf{Y} = \left[\mathbf{y}_1, \ldots, \mathbf{y}^{obs} \right] - \overline{\mathbf{y}_j^f} = \left[\mathscr{H}(\mathbf{x}_1), \ldots, \mathscr{H}(\mathbf{x}_{nens}) \right] - \overline{\mathbf{y}_j^f} \quad (14).$$

Unlike in Lei et.al (2018), the full nonlinear observation operator $\mathscr{H}(\mathbf{x})$ is used here instead of the linearized operator $\mathbf{Hz}$.

Second, we compute observation perturbations to the modulated ensemble $\mathbf{Z}^{\text{mod}}$:

$$\mathbf{Y}^{\text{mod}} = \left[\mathscr{H}(\mathbf{z}_1^{\text{mod}} + \overline{\mathbf{x}_j^f}), \ldots, \mathscr{H}(\mathbf{z}_{n_{ens} n_{eig}}^{\text{mod}} + \overline{\mathbf{x}_j^f}) \right] - \overline{\mathbf{y}_j^f} \quad (15).$$

We compute update weights using Eqs. 8-10 in Lei et.al (2018) as follows. First the eigen decomposition is computed using modulated observation perturbations $\mathbf{Y}^{\text{mod}}$:

$$\mathbf{A}_{vec} \mathbf{A}_{val} \mathbf{A}_{vec}^T \underset{eigen}{\leftarrow} \mathbf{A} = \frac{1}{N_{ens} - 1} \left(\mathbf{Y}_{local(i)}^{\text{mod}} \right)^T \mathbf{R}_{local(i)}^{-1} \mathbf{Y}_{local(i)}^{\text{mod}} \quad (16).$$

Then update weights are computed as:

$$\begin{aligned} \mathbf{W}_{GETKF} &= -\mathbf{A}_{vec} \left[\left(\mathbf{I} - \left(\mathbf{A}_{val} + \frac{1}{\rho} \mathbf{I} \right)^{-1/2} \right) \mathbf{A}_{val}^{-1} \right] \mathbf{A}_{vec}^T \left(\mathbf{Y}_{local(i)}^{\text{mod}} \right)^T \mathbf{R}^{-1/2} \mathbf{Y}_{local(i)} \\ \mathbf{w}_{GETKF} &= \mathbf{A}_{vec} \left(\mathbf{A}_{val} + \frac{1}{\rho} \mathbf{I} \right)^{-1} \mathbf{A}_{vec}^T \left(\left(\mathbf{Y}_{local(i)}^{\text{mod}} \right)^T \mathbf{R}_{local(i)}^{-1} \mathbf{y}_{local(i)}^{innov(k+1)} \right) \end{aligned} \quad (17).$$

The weights are applied to compute analysis perturbations and increments as:

$$\begin{aligned} \mathbf{x}_{(i)}^{inc} &= \mathbf{Z}_{(i)}^{\text{mod}} \mathbf{w}_{GETKF} \\ \mathbf{X}_{(i)}^{inc} &= \mathbf{x}_{(i)}^{inc} + \mathbf{Z}_{(i)}^f + \mathbf{Z}_{(i)}^{\text{mod}} \mathbf{W}_{GETKF} \end{aligned} \quad (18).$$

3.3 LETKF-OI

Frolov et.al (2022) introduced a deterministic form of the LETKF algorithm that uses R-localization to define a parametric error covariance model for univariate variables. In JEDI, this can be implemented by providing the LETKF with two pseudo ensemble members that are generated from a deterministic forecast step x^f, and the standard deviation of the background error σ^f as following:

$$x_i = x^f \pm \frac{\sigma^f}{\sqrt{2}} \quad (19).$$

This gives an ensemble mean of x_i , and an ensemble spread (standard deviation) of σ^f.

3.4 Coupled solvers

To support strongly coupled data assimilation, the JEDI framework will need to implement abstract containers for coupled model states, increments, and operators. At the time of this writing these capabilities were in the early development stages and were not yet integrated in the framework of the local volume solver. Instead, we implemented strongly coupled data assimilation using the interface solver approximation (Frolov et al., 2016; Sluka et al., 2016). The interface solver uses independent solvers for each component of the Earth system model. However, each of the independent solvers has access to the complete set of Earth system observations (or at least to the relevant sub-set of Earth system observations). In other words, the interface solver is coupled in the observational space.

To illustrate the concept of the interface solver, we consider in Figure 3 an example of assimilation into an atmosphere, ocean, and near surface SST coupled model (similar to Akella et al., 2017 and Frolov et al., 2020). Ensembles of coupled states consist of atmospheric, oceanic, and skin SST variables. The EnKF atmospheric observer processes atmosphere-centric observations like radiosondes and radiances (including surface sensitive radiances). The EnKF ocean observer processes ocean-centric observations (such as Argo profiles or retrievals of the ocean SST). We choose to delegate observations of surface sensitive channels to the atmospheric observer, because the atmospheric states usually have all the required information for these observations already interpolated onto the atmospheric grid. We choose to delegate processing of the

SST retrievals to the ocean solver because SST retrievals are usually consistent with the ocean temperature observed by the ocean drifters at 1 m depth. Ocean temperature at 1m depth is more consistent with the gridded oceanic state than with the skin temperature produced by the near-surface SST model. We remind readers that Figure 3 is used for the illustration of the coupled interface solver concept and how it could be implemented by users in their system. We will provide some illustrations of this implementation in section 7.1. However, we also suggest that other users might choose to configure their interface solver differently (e.g., based on direct radiance assimilation that would include the window channels that are primarily sensitive to the SST instead of relying on the SST retrievals as illustrated in our example presented in Figure 3).

4 Available inflation methods

Several options for prior and posterior inflation were implemented in JEDI at the time of writing.

4.1 Prior inflation

Prior inflation is supported in the form of a forgetting factor (Pham et al., 1998) denoted $\frac{1}{\rho}$ in Eqs. (9) and (17). For ρ greater than one, the prior ensemble is inflated. For values of ρ between 0 and 1, the prior ensemble is contracted. Values of ρ less or equal to zero are not admissible.

4.2 Posterior inflation

Two options for posterior inflation are supported: relaxation to the prior perturbations (RTPP; Zhang et al., 2004); and relaxation to prior spread (RTPS; Whitaker & Hamill, 2012). RTPP is defined as:

$$\mathbf{x}^{a'}_{(j)} = (1-\alpha_{\text{RTPP}})\mathbf{x}^{a}_{(j)} + \alpha_{\text{RTPP}}\mathbf{x}^{f}_{(j)} \quad (20),$$

where $\alpha_{\text{RTPP}} \in (0,1]$ is the scalar inflation coefficient, and $\mathbf{x}^{a}_{(j)}$ and $\mathbf{x}^{a'}_{(j)}$ are the analyzed ensemble members (j) before and after RTPP inflation.

RTPS is defined as a grid point operation:

$$\mathbf{x}^{a'}_{(i)(j)} = \alpha_{\text{RTPS}} \frac{(\sigma^{f}_{(i)} - \sigma^{a}_{(i)})}{\sigma^{a}_{(i)}} \mathbf{x}^{a}_{(i)(j)} \quad (21),$$

Where $\alpha_{\text{RTPS}} \in (0,1]$ is the scalar inflation coefficient, $\sigma^{f}_{(i)}$ and $\sigma^{a}_{(i)}$ are the standard deviations (spread) of the prior and posterior ensemble, and $\mathbf{x}^{a}_{(i)(j)}$ and $\mathbf{x}^{a'}_{(i)(j)}$ are the analyzed ensemble members (j) at the (i)$^{\text{th}}$ grid point before and after RTPS inflation.

5 Localization

5.1 Observation space localization for Earth system components

Observation space localization $\mathbf{\Psi}(\|loc(i) - loc(y)\|, \psi)$ in Eq. (7) inflates observations errors for local observations as a function of some distance between the location of the local volume $loc(i)$, the location of the observation $loc(y)$, and the localization length scale ψ. It is customary to use the inverse of a correlation-like function, such as the Gaspari-Cohn (Gaspari & Cohn, 1999), that is equal to 1 at the location of the local volume and decays to zero (infinite inflation) over some support radius ψ.

When localization in more than one dimension is required (e.g., horizontal and vertical), different localization scales (e.g., ψ_{hor} and ψ_{vert}) in each dimension might be

required. The total localization function can then be composed as a function of multidimensional distance in one of the two ways.

One could add all normalized distances and then compute the localization function:

$$\boldsymbol{\Psi}(\|loc(i)-loc(y)\|,\psi_{hor},\psi_{vert}) = \boldsymbol{\Psi}(\sqrt{\frac{\|loc(i)-loc(y)\|^2_{hor}}{\psi^2_{hor}}+\frac{\|loc(i)-loc(y)\|^2_{vert}}{\psi^2_{vert}}}) \quad (22).$$

Alternately, one could assume that the localization functions are separable and compose the total localization function as a product of each individual localization:

$$\boldsymbol{\Psi}(\|loc(i)-loc(y)\|,\psi_{hor},\psi_{vert}) = \boldsymbol{\Psi}\left(\frac{\|loc(i)-loc(y)\|^2_{hor}}{\psi^2_{hor}}\right)\boldsymbol{\Psi}\left(\frac{\|loc(i)-loc(y)\|^2_{vert}}{\psi^2_{vert}}\right) \quad (23).$$

The JEDI implementation uses the second option for two reasons. First, using separable localizations allows us to keep track of distances in each dimension within the object that implements each of the localization blocks. And second, each localization block can use a different correlation function; for example, horizontal localization could use the Gaspari-Cohn function while vertical localization could use the Gaussian function.

Specifying observation localization for each observation stream independently allows the JEDI local volume solver to be flexible. For example, one could specify wider localization for radiosonde observations than for radar reflectivity observations. In the case of the coupled data assimilation example given earlier, one could specify different localization scales for the SST, the ocean in-situ temperature, and for the atmospheric wind observations.

Finally, the observation space localization (as implemented in JEDI) allows users to limit the number of observations used in a local volume analysis (by selecting those

353 closest to the analysis grid point). Hamrud et al. (2015) showed that selecting the closest
354 observations allows for a significant speed up of the LETKF algorithm with no loss in the
355 fidelity of the LETKF update.

356 5.1.1 Horizontal, distance-based localization

357 Several options for horizontal R-localization are supported by JEDI, including
358 Box Car, Gaspari-Cohn, and the second order auto regressive (SOAR) function (see
359 Figure 4). Computing great-circle distances between two points on a sphere can be
360 computationally expensive. Instead of the direct computation of the distances, we use
361 KD-tree lookups. Horizontal observation localization also selects the nearest N
362 observations closest to the updated volume.

363 5.1.2 Vertical localization

364 The Box Car, Gaspari-Cohn, and SOAR functions are also supported for vertical
365 localization. The vertical distance between the local volume and the observation is
366 computed in the vertical coordinate of choice (e.g., pressure, depth, or height). An
367 optional log transformation to the vertical coordinate is also supported before the vertical
368 distance is computed.

369 To implement R-localization in the vertical, the vertical location of each
370 observation must be included in the observation file, and the model interface must then
371 return a matching vertical location for each analysis grid point. For some observation
372 types and model grids this might not be possible. For example, satellite radiances observe
373 integrated vertical properties, which can be difficult to relate to a specific observation
374 height. Alternatively, many ocean-specific implementations of the Kalman filter analyze
375 the entire vertical column of grid points, again making it difficult to assign a specific

height to the analysis volume. In such cases, model space localization discussed in section 5.2 might be a more appropriate vertical localization strategy.

5.1.3 Component-specific examples

5.1.3.1 Brasnett localization for land DA

To enable localization of land-specific observations like snow, JEDI includes the Brasnett localization (Brasnett, 1999), which attenuates the localization as a function of horizontal distance and the vertical difference in the orography of the observation location and the analyzed grid point:

$$L_{brasnett} = (1 + c_x l_x)\exp\left[-\left(c_x l_x\right)\right]\exp\left[-\left(\frac{l_z}{c_z}\right)^2\right] \quad (24),$$

Where $L_{brasnett}$ is the Brasnett correlation function, l_x is the horizontal distance between the two points in meters, l_z is the vertical (terrain) difference between the two points in meters, c_x is the horizontal lengthscale in m^{-1} (default value recommended in Brasnett, (1999) is 0.000018 m^{-1}, which is equivalent to a 120km e-folding scale), and c_z is the horizontal coefficient in meters (default is 800 m). Figure 4 in Brasnett (1999) provides a nice illustration how steep terrain in the Western United States affects correlation structures modeled by Eq. (24).

5.1.3.2 Rossby-radius based localization for ocean DA

Because water has a higher density than air, the first Rossby radius of deformation for baroclinic instabilities varies latitudinally in the ocean from ~300 km near the equator to 3 km in the Arctic ocean (Chelton et al., 1998). This variability affects error correlation scales in eddy resolving models. To accommodate this variability, we introduced a

variant of the Gaspari-Cohn function with the support radius varying as a function of the local Rossby radius as following:

$$L_{rossby} = \min\left\{\max\left[\max\left(L_0 + m_r L_r, m_g\sqrt{a_{(i)}}\right), L_{\min}\right], L_{\max}\right\} \quad (25)$$

where L_{rossby} is the support radius provided to the Gaspari-Cohn function; L_{max} and L_{min} are the user-specified maximum and minimum support radii; L_0 is the user-specified base value of the localization radius; L_r is the Rossby radius of first baroclinic deformation from (Chelton et al., 1998); $\sqrt{a_{(i)}}$ is the square root of the grid cell area at the location of the analysis volume; and m_r and m_g are the user-specified multiples. Eq. (25) provides a lot of flexibility to model localization radius L_{rossby}. The default values for Eq. (25) are $L_0=0$, $m_r=1.0$ and $m_g=1.0$, $L_{max}=inf$, and $L_{min}=0.0$.

5.2 Model space localization

The GETKF solver described in section 3.2 supports model-space localization in the vertical direction, similar to the original description in Lei et.al (2018). We expect that future implementations might also support model space localization in the horizontal direction, similar to the MLGETKF solver introduced by Wang et al. (2021).

Currently several options are supported for specifying the vertical localization matrix. One option is for the model interface to return the vertical coordinate of the model; then a correlation matrix can be computed based on the localization scale specified in this vertical coordinate. Examples of the vertical coordinates include model level numbers and log pressure. Another option is for the GETKF solver to read the square root of the horizontally-varying vertical localization matrix from disk (one model increment file for each column of the square root). These files can be generated in several ways. For example, one can compute the correlation matrix from a random sample drawn

from the parametrized form of the vertical localization operator used in the ensemble variational solver (EnVAR). Alternatively, a user can implement their own vertical localization operator that can output the square root of the vertical correlation as a collection of increment files.

6 Object-oriented abstraction for the local volume solver in JEDI

To enable a generic implementation of the local volume solver with different update formulas and model implementations, JEDI uses an object-oriented design. Specifically, the key elements of the local solver framework are abstracted as objects that can provide internal storage for variables and standard methods that can be overloaded based on the solver formalism. The design includes basic abstraction classes provided by the Object-Oriented Prediction System (OOPS) library (Figure 1) and spatialized classes that were developed to implement local volume solvers.

6.1 Parallel data distribution

6.1.1 Model-space data distribution

Model implementations (lower left block in Figure 1) implement the Message Passing Interface (MPI) distribution of the model states in JEDI. These distributions can vary depending on the model implementation.

6.1.2 Observation-space distribution

Several options for parallel distribution of observations are available in JEDI at the time of this writing:

- Round robin distribution, which randomly distributes observations across all available processing elements (PEs) to optimize the balance of observations across PEs.
- Inefficient distribution, which replicates all observations on each processor.
- Halo distribution, which stores overlapping sets of observations on each PE. This distribution assumes that the model grid is distributed across PEs in tiles (blue and black grids in Figure 2). The overlapping observations sets are described by a user-specified circle centered on the geometric center of the local model tile (large blue circle centered on the blue tile in Figure 2 or blue triangle in Figure 5). The default halo radius (halo size) is set large enough to encompass all observations on Earth, which reduces the Halo distribution to the Inefficient distribution. Figure 5 shows the details of how Halo distribution is computed. In practice, the Halo size is selected to be just large enough to contain observations that are necessary to update all grid points on a PE.

The solver step of the local volume solvers (Eq. (5)) assumes that all observations that are required to make an analysis update at a particular model point are available on the PE owning this model point, and no communications are needed. Thus, only Inefficient or Halo distributions can be used for the solver step. Because the solver has to store the ensemble of observations, Inefficient distribution can easily run out of memory available to each PE on the computational node. Using Halo distribution is highly recommended for global observational counts above 1 million.

In contrast to the solver step, the observation step of the local volume solvers (Eq. (1)) uses any of the distributions listed above. For the observation step, the round robin distribution provides the best load balancing between different PEs, and has the best time-to-completion because it uses non-overlapping distributions of observations. Inefficient distribution provides the longest time to completion, doesn't scale with the number of PEs, and is impractically slow for observation counts on the order of 1 million. Halo distribution provides a practical balance between the two options.

The current implementation of the local ensemble solver in JEDI also allows the use of different observation distributions during the observer (Eq. (1)) and the solver (Eq. (5)) steps. This is accomplished by executing the EnKF application in the observer-only mode (e.g., using the most efficient round robin distribution for the observer step), followed by the execution of the EnKF application in the solver-only mode (e.g., using the most efficient Halo distribution for the solver step). The redistribution of the observations between the two runs is accomplished by writing and reading of the intermediate observer files. Future implementations of JEDI might support this observation space redistribution in memory without the need to dump intermediate files to disk.

6.1.3 Ensemble transpose

Ensemble DA algorithms benefit from optimized storage of ensembles of model states and observations, with the ensemble dimension as the fastest varying dimension in multidimensional arrays. At the time of writing, such optimized storage has not yet been implemented in JEDI. Instead, model states, observations, and all ensemble containers provide the '`packEigen()`' method that can pack local observations and local model states

into consecutive memory that can be passed to linear algebra libraries like LAPACK or Eigen. A companion 'setEigen()' method is also provided that can assign data from a packed array for local observations and model increments back to JEDI objects.

Figure 6 illustrates how the 'packEigen()' and 'setEigen()' methods operate on the ensemble of observations stored in the JEDI native format. JEDI stores each ensemble of observations as a list of lists. In this storage configuration, only the observations from the same observation type for a given ensemble member are stored consecutively in memory. Performing 'packEigen()' and 'setEigen()' operations requires costly non-sequential memory operations. Optimization of the ensemble storage for the local volume solver in JEDI is an avenue for future optimization.

7 Case studies

To illustrate usage of the developed local volume solvers in realistic applications, we provide a summary of case studies that involve assimilation of the atmospheric, oceanic, land, and atmospheric composition measurements. These case studies include assimilation of single observations that illustrate how information is spread from observation location to grid points. We also present limited cycling experiments that compare forecasts from the local volume analysis against a combination of other reference systems (e.g. variational or systems with no data assimilation).

7.1 Single observation experiments

Figures 7 and 8 illustrate the impact of model space localization on spreading of the innovation vertically in the atmosphere from assimilation of a single surface pressure observation (Figure 7) and a lower-troposphere temperature observation (Figure 8).

In Figure 7, the surface pressure observation is localized by modulating the 3D pressure field, instead of prescribing localization directly between surface pressure and the rest of the 3D state. Figure 7 shows that the surface pressure is more correlated with the mid- and high- tropospheric temperatures and layer pressures than with the temperature and layer pressure in the boundary layer directly above the surface. This finding is intuitive as surface pressure measures the integral of the vertical column and is sensitive to the location of the large-scale weather systems that tend to locally displace the tropopause, resulting in areas of high- and low-pressure systems.

To date, the applications of JEDI in atmosphere and ocean have been developed in a weakly-coupled DA framework, in which observations of different fluids cannot directly impact other fluids. Applications in the strongly coupled framework are still in the early proof-of-concept stage. For example, Figure 9 illustrates the impact on the ocean temperature profile from assimilating a single sea surface temperature (SST—blue line) and a single atmospheric surface temperature (AST—red line) observation. No vertical localization is used for either the within-fluid (SST-ocean) or the across-fluid (AST-ocean) assimilation. Both SST and AST assimilation have a large impact on the ocean temperatures in the top 100 meters of the water column. Overall, both increments warm and deepen the ocean mixed layer.

The increments presented in Figure 9 illustrate the cross-fluid assimilation capabilities of the JEDI local volume solver, but we caution readers from over-interpreting the scientific merit of these increments. For example, unlike the SST increment which is confined to the top 100 meters, the AST increment has a strong negative increment between 300 and 400 meters deep. We speculate that this deep

increment is likely due to spurious correlations between the ocean and the atmosphere and should be attenuated with an appropriate cross-fluid localization. We also note that within the mixed layer, the magnitude of the AST increment is larger than the SST increment. We attribute this to the poorly calibrated ensemble spread in the coupled ensemble, where ocean temperatures are under spread compared to the atmospheric temperatures.

7.2 Cycling experiments

This section presents limited results that illustrate cycling of the local volume solvers in three distinct application areas: ocean, land, and aerosol assimilation. When appropriate, we compare cycling results for the local volume solvers against other systems (such as variational solvers or runs without data assimilation). Several results presented in this section were obtained from early engineering studies focused on demonstrating the feasibility of cycling with a family of JEDI solvers. We expect that, as these systems mature, separate scientific evaluations will be published by the respective research groups fully detailing scientific findings from their cycling experiments.

7.2.1 Ocean

In this section, we use the LETKF with a small representative subset of marine observations and compare its performance to the 3DVar data assimilation scheme using the JCSDA's Sea-Ice, Ocean and Coupled Assimilation (SOCA) framework for the MOM6 Hurricane North Atlantic (HAT10) domain. This regional study uses the SOCA-SCIENCE code repository (JCSDA, 2023e) to cycle the experiments. The experiments are cycled every 24 hours. The horizontal resolution of the model is 1/4°, and it has 75 vertical levels. The DA schemes assimilate absolute dynamic topography (ADT) and sea

surface temperatures (SST) only. The assimilated ADT observations include data from CRYOSAT-2, JASON-3, Saral/ALTIKA, Sentinel-3A, and Sentinel-3B. The level 4 AVISO ADT from Copernicus Marine Services is used to estimate forecast errors but is not assimilated. The SST observations are from the Visible Infrared Imaging Radiometer Suite (VIIRS) instrument onboard the Suomi National Polar-Orbiting Partnership (Suomi NPP) spacecraft. The LETKF here uses Rossby radius-based localization scales, which were found to be better (not shown) compared to the Gaspari-Cohn method.

The results indicate that the LETKF is performing as expected, and even with minimal tuning it is able to outperform the existing 3DVAR (Figure 10). Both data assimilation schemes lead to better performance overall than the no DA case, however, the 3DVAR does show degradations in certain areas near the western boundary currents that the LETKF does not suffer from (Figure 11). This is particularly noticeable in the SST observation departure (O-B) RMSD along the Equatorial currents (Figure 11 a/b) and in the ADT O-B RMSD along the Gulf Stream (Figure 11 c/d). This is likely due to the fact that the LETKF is able to apply the velocity increments while 3DVAR currently does not, leading to overall better-balanced state estimation by LETKF. These findings demonstrate that the JEDI-LETKF is both technically and scientifically capable of further development for ensemble ocean data assimilation.

7.2.2 Land

Figure 12 demonstrates the application of the LETKF-OI to assimilate snow depth observations into NOAA's land-surface model, Noah-MP. Point-based snow depth observations from NOAA's Global Historical Climatology Network Daily (GHCN; Menne et al., 2012) are assimilated once daily into an offline (land-only) version of the

Noah-MP model, from 1 September 2019 to 1 May 2020. The model is run at 1-degree resolution, and forced with archived output from NOAA's GDAS/GFS system. As expected, the LETKF-OI effectively improves the fit between the forecasts and observations, reducing the mean daily O-F standard deviation from 72.0 mm to 51.3 mm, and reducing the mean daily O-F mean from -14.1 mm to 0.2 mm. The snow depth LETKF-OI is being prepared for implementation in the NOAA operational system. This system is based closely on the OI-based snow analyses in use at other weather prediction centers (Brasnett, 1999; Gichamo & Draper, 2022; de Rosnay et al., 2012), and will assimilate both station snow depth observations and remotely sensed snow cover observations.

7.2.3 Atmospheric composition

A global aerosol data assimilation system for the NOAA Global Ensemble Forecast System - Aerosols (GEFS-Aerosols; Zhang et al., 2022) was recently developed based on the three-dimensional ensemble-variational (3D-EnVar) framework in JEDI. In this system, the aerosol optical depth (AOD) that represents the total amount of aerosols in a column was assimilated to constrain aerosol concentrations in the GEFS-Aerosols initial conditions. Briefly, the one-member control aerosol analysis was obtained by the 3D-EnVar solver in JEDI that used the pure ensemble-estimated background error covariance for aerosols. The background ensemble was updated by the LETKF or GETKF solver as described in this manuscript. Detailed system descriptions and evaluations in the retrospective and near-real-time experiments were documented in Huang et al. (n.d.). In this section, we present the initial AOD assimilation application using the 3D-EnVar solver in JEDI in combination of the LETKF and GETKF solvers

(hereafter referred to as EnVar-L and EnVar-G, respectively). Two six-hourly cycled AOD assimilation experiments that employed EnVar-L and EnVar-G, respectively, were performed in June 2016 to assimilate within GEFS-Aerosols the Visible Infrared Imaging Radiometer Suite (VIIRS) 550 nm AOD retrievals produced by the National Environmental Satellite, Data, and Information Service (NESDIS) at NOAA (Jackson et al., 2013). The AOD assimilation impacts were investigated by comparing with another cycled experiment without AOD assimilation (hereafter referred to as NODA) in the same period.

Figure 13 compares simulated AOD from the NODA six-hour forecast, the one-member control and ensemble mean analyses in the EnVar-L and EnVar-G experiments against assimilated VIIRS AOD retrievals. As expected, compared to the NODA six-hour forecast, both the control and ensemble mean analyses in the two AOD assimilation experiments showed significantly improved agreement with VIIRS AOD retrievals in terms of AOD value, biases and root-mean-square errors (RMSEs), suggesting the substantial benefits of assimilating VIIRS AOD retrievals within GEFS-Aerosols. In these two AOD assimilation experiments, the one-member control analyses obtained from EnVar showed slightly lower biases and RMSEs than their corresponding ensemble mean analyses produced by LETKF and GETKF, respectively. In addition, the biases and RMSEs in the one-member control analysis in the EnVar-L experiment were marginally lower than those in the EnVar-G experiment, while their ensemble mean analyses were comparable. The differences of the resulting control and ensemble mean analyses in these two AOD assimilation experiments could be attributed to their specific localization strategies as discussed in this manuscript and their sensitivities to localization length

scales and inflation factors. Further tuning of localization length scales and inflation factors in EnVar-L and EnVar-G will be performed to optimize their assimilation performance in the future.

8 Summary, conclusions, and future work

This paper describes a model agnostic implementation strategy for the local volume solvers in the JEDI framework. We found that EnKF implementations that target Earth system assimilation need to support a wide variety of ensemble localization strategies, as appropriate for the different components of the Earth system. For example, model space localization (GETKF) was requested by the atmospheric community, observation-space localization (LETKF) with localization distance condition on the Rossby radius of deformation was requested by the oceanic community, and terrain-dependent localization was requested by the land-modeling community. Our preliminary demonstration of a strongly coupled DA example (Figure 9) demonstrates that further work is needed to develop, implement, and test methods for cross-fluid localization operators.

Another challenge specific to a generic DA framework such as JEDI, is efficiently distributing computations and memory across PEs. To our knowledge, EnKF-specific implementations like DART have engineered their PE distributions to be optimized specifically for the EnKF computations, by distributing both the model grid points and observations randomly across the PEs (method 4.a of (Anderson & Collins, 2007)). This ensures good load balancing at the expense of inter-processor communication. By contrast, the JEDI framework supports a number of applications with different load balancing requirements, such as forward model simulations, observation operators,

variational solvers, and EnKF solvers. We found that in a generic framework like JEDI, it is more challenging to develop grid and observation containers that were optimized specifically for the EnKF computation. At the time of writing, our implementation inherited grid point distributions from the model interfaces (e.g., cubed-sphere for the FV3 model) and implemented a new ("Halo") observation distribution that stores only observations required for the local volume computation on each PE (Figure 2). This is similar to method 4.b of Anderson and Collins (2007). We are currently exploring ways to improve load and memory balancing for local volume computations.

During the development of the framework for local volume solvers, we benefited greatly from the large number of contributors to the JEDI framework, who tested the generic local volume solvers that we developed in this paper using atmospheric, oceanic, land, and aerosol test cases. Our brief summary of the cycling results showed that different variants of the local volume solver (GETKF, LETKF, LETKF-OI) all compared well to a variety of reference solutions including a free running model (land and aerosols) and variational solvers (ocean and aerosols).

Testing of the developed solvers in the cycling system also revealed that further work is needed to improve scalability, computational performance, and memory management of the local volume solver implementation in JEDI. We chose not to highlight specific computational results in this paper as the JEDI code optimization is rapidly progressing and any results with regard to the computational performance presented in this paper are likely to become quickly obsolete.

Finally, we believe that this local volume framework lays the ground work for future algorithmic developments. Specifically, JEDI can benefit from implementation of

the serial solver (Anderson, 2001), which replaces the matrix inverse in Eqs. (10) and (17) with a sequence of scalar inverse problems. The serial solver is a pre-requisite to the implementation of non-linear filters, such as the local particle filters (Poterjoy, 2016), or the Gamma, Inverse-Gamma, and Gaussian (GIGG) filter (Bishop, 2016). The most natural implementation of serial filters in JEDI would involve replacing the Ensemble Transform computation for each of the local volumes with the serial EnKF update (similar to method 4.b in Anderson & Collins, 2007). Another intriguing direction would be to implement iterative Kalman Filters, which can exploit the existing library of linearized observation operators in JEDI (Bocquet & Sakov, 2014). Implementation of the multi-scale localization operators similar to (Wang et al., 2021) would be yet another beneficial development.

Acknowledgments

S. Frolov, J. Whitaker, and W. Huang acknowledge support from various grants provided by the NOAA Weather Program Office, Earth Prediction and Innovation Center, and the Joint Technology Transfer Initiative. A. Shlyaeva, T. Sluka, and K. Bhargava acknowledge support from the Joint Center for Satellite Data assimilation. B. Huang would like to acknowledge the funding support from NOAA/OAR/WPO/Air Quality Program (NA19OAR4590080) and NOAA/OAR/CPO/MAPP program (NA18OAR4310281) for the aerosol data assimilation capability development in JEDI.

All authors are grateful for computational capabilities provided by the NOAA Research and Development HPC System.

Open Research

The software code for the generic JEDI solvers is available in the OOPS code repository (JCSDA, 2023d). Software implementations for land, atmosphere, and composition are available in the FV3-JEDI code repository (JCSDA, 2023a). Software implementation for the ocean and ice components is available in the SOCA code repository (JCSDA, 2023f). The Unified Forecast System model is available in the UFS code repository (UFS, 2023). Training and documentation for the JEDI software are available in JCSDA (2023b) and JCSDA (2023c), respectively.

Figures:

Figure 1: Separation of concerns in JEDI. Image adopted from JEDI academy slides (http://academy.jcsda.org/2021-10/slides/2021-06-21-WhyOOPSJEDI.pptx). Model implementation acronyms correspond to different model implementations and are defined in the main text of the paper. Other acronyms include: OOPS—Object Oriented Prediction System; SABER—System Agnostic Background Error Representation; UFO—Unified Forward Operator; VADER—Variable Derivation Repository; and IODA--Interface for Observational Data Access.

Figure 2: Local volume solver domain decomposition. Blue grid: grid points stored on the current (this) processing element (PE). Black grid: grid points on the neighboring PE. Green square indicates the updated, analysis point x(i). Dots indicate observations: (orange) observations used in the analysis; (red) observations that are needed to update all points on this PE; (gray) other observations that are not needed to update grid points on this PE. Circles indicate: (orange) the area used to collect local observations for the current local analysis, (blue) halo region that contains all possible

observation locations needed to update grid points on this PE, (gray) observations halo region that contains all possible observation locations needed to update grid points on neighboring PE.

Figure 3: Example of interface solver implementation in JEDI framework. A pair of LETKF observers are run first obtaining ensemble observations of all atmospheric and oceanic variables. In this illustration, the near surface SST forecast (nsst) from the diurnal layer model is used as a first guess by the atmospheric observer for observations of the surface-sensitive radiances. Ocean state is used by the ocean observer to estimate the forecast of SST retrievals. Then a pair of LETKF solvers is executed one producing analysis on the atmospheric and the second on the oceanic grids. Each solver can use a combination of atmospheric and oceanic observations. In this example atmospheric solver is tasked to produce the analysis of the nsst variable as it has more complete picture of atmospheric observations that can potentially influence the nsst estimate.

Figure 4: Localization as a function of normalized distance and localization type. For illustration purposes, distance was scaled by a factor of 8 before the SOAR function was calculated to achieve a near zero value for normalized distance of 1 (e.g. the SOAR function value is 0.003 for the distance of 8).

Figure 5: Halo distribution of observations. Shown are (blue grid) the grid points stored on this PE. (Blue triangle) halo center computed as a center of the grid points. (Blue circle) patch radius R_{patch} that encloses all the grid points on this PE; (Orange circle) halo size radius $R_{halo\ size}$ that is added to the patch circle. In practice, $R_{halo\ size}$ is related to the localization radius. (Green circle) the total halo circle that is computed as a summation of $R_{patch} + R_{halo\ size}$. Observations can then be divided in to patch observations

738 that exclusively belong to this PE based on the shortest distance to the patch center (the
739 blue dot). Halo observations (green dots) that are within the green total halo radius. And
740 observations that are not stored on this PE because they are outside of the total halo
741 radius (gray dots).

742 Figure 6: packEigen and setEigen methods operating on observational data in
743 OOPS. Top row present data stored in generic containers for observational data in OOPS.
744 Observational data stored as vectors of vectors, with each blue rectangle stored as
745 contiguous data but with potential extraneous memory between blocks. Bottom row
746 shows contiguous storage of ensembles of local observational data. The entire record is
747 contiguous with the ensemble dimension as a leading index.

748 Figure 7: Increment to the vertical column of atmospheric temperature (a) and
749 layer pressure (b) from assimilation of single surface pressure observation at 40.5°N,
750 160.5°E and innovation of 6 hPa. Background ensemble valid for 2015-12-05 18:00Z.
751 GETKF vertical localization was used with localization scale of 30 levels and 10 eigen
752 vectors capturing 96% of the variance.

753 Figure 8: Increment to the vertical column of atmospheric temperature (a) and
754 layer pressure (b) from assimilation of single temperature observations at 40.5°N,
755 160.5°E, 950 hPa, and innovation of 1 K. Background ensemble valid for 2015-12-05
756 18:00Z. GETKF vertical localization was used with localization scale of 30 levels and 10
757 eigen vectors capturing 96% of the variance.

758 Figure 9: Increment to the vertical column of ocean temperature from assimilation
759 of a single atmospheric temperature observations at 27.5°S, 154.5°W, 950 hPa (red) and
760 sea surface temperature observation at the same horizontal location (blue). Both with the

761 innovation of 1 K. Background ensemble valid for 2015-12-05 18:00Z. No vertical
762 localization was used.

763 Figure 10: RMSE for the 24-hour forecast of (a) SST and (b) ADT observations.
764 Colors indicate: (black) no data assimilation, (green) 3DVAR algorithm, and (red)
765 LETKF.

766 Figure 11: RMSE for the difference between run with no data assimilation and
767 3DVAR (left column, panels a and c) and LETKF (right column, panels b and d). Top
768 row shows differences for SST (panels a and b) and the bottom row shows differences for
769 ADT (panels c, d).

770 Figure 12: RMSE for the 6-hour forecast of the snow depth (GHCN network).
771 Colors indicate: (black) no data assimilation, and (red) LETKF-OI assimilation of snow
772 depth measurements. Line style indicates: (solid) standard deviation of error (dashed)
773 mean error.

774 Figure 13: (a) Time-series of 550 nm AOD means from VIIRS (gray), no data
775 assimilation six-hour forecast (black), the one-member control and ensemble mean
776 analyses in the EnVar-L (blue and green, respectively) and EnVar-G (red and orange,
777 respectively) experiments in a six-hourly interval in June 2016, and their corresponding
778 (b) biases and (c) root-square-mean errors (RMSEs) against VIIRS AOD retrievals. (d)-
779 (f) are the temporally means of the corresponding (a)-(c) over the last three weeks in June
780 2016.

781

782 References:

783 Adcroft, A., Anderson, W., Balaji, V., Blanton, C., Bushuk, M., Dufour, C. O., et al.
784 (2019). The GFDL Global Ocean and Sea Ice Model OM4.0: Model Description and
785 Simulation Features. *Journal of Advances in Modeling Earth Systems*, *11*(10), 3167–

3211. https://doi.org/10.1029/2019MS001726
Akella, S., Todling, R., & Suarez, M. (2017). Assimilation for skin SST in the NASA GEOS atmospheric data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, *143*(703), 1032–1046. https://doi.org/10.1002/qj.2988
Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., et al. (1999). *LAPACK Users' Guide, Third Edition*. Philadelphia: SIAM .
Anderson, J., Hoar, T., Raeder, K., Liu, H., Collins, N., Torn, R., & Avellano, A. (2009). The Data Assimilation Research Testbed: A Community Facility. *Bulletin of the American Meteorological Society*, *90*(9), 1283–1296. https://doi.org/10.1175/2009BAMS2618.1
Anderson, J. L. (2001). An Ensemble Adjustment Kalman Filter for Data Assimilation. *Monthly Weather Review*, *129*(12), 2884–2903. Retrieved from http://dx.doi.org/10.1175/1520-0493(2001)129%3C2884:AEAKFF%3E2.0.CO;2
Anderson, J. L., & Collins, N. (2007). Scalable Implementations of Ensemble Filter Algorithms for Data Assimilation. *Journal of Atmospheric and Oceanic Technology*, *24*(8), 1452–1463. https://doi.org/10.1175/JTECH2049.1
Barton, N., Metzger, E. J., Reynolds, C. A., Ruston, B., Rowley, C., Smedstad, O. M., et al. (2020). The Navy's Earth System Prediction Capability: a new global coupled atmosphere-ocean-sea ice prediction system designed for daily to subseasonal forecasting. *Earth and Space Science*. https://doi.org/10.1029/2020EA001199
Bishop, C. H. (2016). Ensemble Kalman Filtering for highly skewed non-negative uncertainty distributions. *Quarterly Journal of the Royal Meteorological Society*, *143*(703), 1395–1412.
Bishop, C. H., Whitaker, J. S., & Lei, L. (2017). Gain form of the ensemble transform Kalman Filter and its relevance to satellite data assimilation with model space ensemble covariance localization. *Monthly Weather Review*, *145*(11), 4575–4592. https://doi.org/10.1175/MWR-D-17-0102.1
Bocquet, M., & Sakov, P. (2014). An iterative ensemble Kalman smoother. *Quarterly Journal of the Royal Meteorological Society*, *140*(682), 1521–1535. https://doi.org/10.1002/qj.2236
Brasnett, B. (1999). A Global Analysis of Snow Depth for Numerical Weather Prediction. *Journal of Applied Meteorology*, *38*(6), 726–740.
Browne, P. A., de Rosnay, P., Zuo, H., Bennett, A., & Dawson, A. (2019). Weakly coupled ocean-atmosphere data assimilation in the ECMWF NWP system. *Remote Sensing*. https://doi.org/10.3390/rs11030234
Buehner, M. (2020). Local Ensemble Transform Kalman Filter with Cross Validation. *Monthly Weather Review*, *148*(6), 2265–2282. https://doi.org/10.1175/MWR-D-19-0402.1
Chelton, D. B., deSzoeke, R. A., Schlax, M. G., El Naggar, K., & Siwertz, N. (1998). Geographical Variability of the First Baroclinic Rossby Radius of Deformation. *Journal of Physical Oceanography*. https://doi.org/10.1175/1520-0485(1998)028<0433:GVOTFB>2.0.CO;2
Daley, R. (1991). *Atmospheric Data Analysis*. Cambridge University Press.
Eigen. (2023). Eigen: C++ template library for linear algebra. Retrieved from https://eigen.tuxfamily.org/
Evensen, G. (2003). The Ensemble Kalman Filter: theoretical formulation and practical

implementation. *Ocean Dynamics*, *53*(4), 343–367.
Evensen, G. (2004). Sampling strategies and square root analysis schemes for the EnKF. *Ocean Dynamics*, *54*, 539–560.
Frolov, Sergey;, Campbel, W., Ruston, B., Bishop, C. H., Kuhl, D. D., Flatau, M. K., & McLay, J. G. (2020). Assimilation of Low-Peaking Satellite Observations Using the Coupled Interface Framework. *Monthly Weather Review*, *148*(2), 637–654. https://doi.org/https://doi.org/10.1175/MWR-D-19-0029.1
Frolov, Sergey, Bishop, C. H., Holt, T. R., Cummings, J. A., & Kuhl, D. D. (2016). Facilitating strongly-coupled ocean-atmosphere data assimilation with an interface solver. *Monthly Weather Review*, *144*, 3–20. https://doi.org/http://dx.doi.org/10.1175/MWR-D-15-0041.1
Frolov, Sergey, Whitaker, J. S., & Draper, C. (2022). Including parameterized error covariance in local ensemble solvers: experiments in a 1D model with balance constraints. *Quarterly Journal of the Royal Meteorological Society*. https://doi.org/10.1002/QJ.4289
Gandin, L. (1963). *Objective Analysis of Meteorological Fields*. Leningrad: Hydrometeoizdat.
Gaspari, G., & Cohn, S. E. (1999). Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, (April 1998), 723–757.
Gichamo, T. Z., & Draper, C. S. (2022). An Optimal Interpolation–Based Snow Data Assimilation for NOAA's Unified Forecast System (UFS). *Weather and Forecasting*, *37*(12), 2209–2221. https://doi.org/10.1175/WAF-D-22-0061.1
Haidvogel, D. B., Arango, H. G., Hedstrom, K., Beckmann, A., Malanotte-Rizzoli, P., & Shchepetkin, A. F. (2000). Model evaluation experiments in the North Atlantic Basin: simulations in nonlinear terrain-following coordinates. *Dynamics of Atmosphere and Oceans*, *32*, 239–281.
Hamrud, M., Bonavita, M., & Isaksen, L. (2015). EnKF and hybrid gain ensemble data assimilation. Part I: EnKF implementation. *Monthly Weather Review*, *143*(12), 4847–4864. https://doi.org/10.1175/MWR-D-14-00333.1
Harris, L., Chen, X., Putman, W., Zhou, L., & Chen, J.-H. (2021). A Scientific Description of the GFDL Finite-Volume Cubed-Sphere Dynamical Core. https://doi.org/10.25923/6NHS-5897
Huang, B., Pagowski, M., Trahan, S., Martin, C. R., Tangborn, A., Kongdragunta, S., & Kleist, D. T. (n.d.). JEDI-Based Three-Dimensional Ensemble-Variational Data Assimilation System for Global Aerosol Forecasting at NCEP. *Journal of Advances in Modeling Earth Systems*.
Hunke, E. C., & Lipscomb, W. H. (2015). *CICE : the Los Alamos Sea Ice Model Documentation and Software User's Manual Version 4.0. Los Alamos National Laboratory Tech. Rep. LA-CC-06-012*.
Hunt, B. R., Kostelich, E. J., & Szunyogh, I. (2007). Efficient Data Assimilation for Spatiotemporal Chaos : a Local Ensemble Transform Kalman Filter. *Physica D*, *230*, 112–126.
Jackson, J. M., Liu, H., Laszlo, I., Kondragunta, S., Remer, L. A., Huang, J., & Huang, H. C. (2013). Suomi-NPP VIIRS aerosol algorithms and data products. *Journal of Geophysical Research: Atmospheres*, *118*(22), 12,673-12,689.

https://doi.org/10.1002/2013JD020449
JCSDA. (2023a). FV3-JEDI. Boulder, CO: Joint Center for Satellite Data Assimilation. Retrieved from https://github.com/JCSDA/fv3-jedi
JCSDA. (2023b). JCSDA Academy. Retrieved from http://academy.jcsda.org/
JCSDA. (2023c). JEDI Documentation. Retrieved from https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/
JCSDA. (2023d). OOPS. Boulder, CO: Joint Center for Satellite Data Assimilation. Retrieved from https://github.com/JCSDA/oops/
JCSDA. (2023e). SOCA-Science. Boulder, CO: Joint Center for Satellite Data Assimilation. Retrieved from https://github.com/JCSDA-internal/soca-science
JCSDA. (2023f). SOCA Tag 1.0.0. Boulder, CO: Joint Center for Satellite Data Assimilation. Retrieved from https://github.com/JCSDA/soca/releases/tag/1.0.0
Kotsuki, S., & Bishop, C. H. (2021). Implementing Hybrid Background Error Covariance into the LETKF with Attenuation-based Localization: Experiments with a Simplified AGCM. *Monthly Weather Review*, *1*(aop). https://doi.org/10.1175/MWR-D-21-0174.1
Kotsuki, S., & Bishop, C. H. (2022). Implementing Hybrid Background Error Covariance into the LETKF with Attenuation-Based Localization: Experiments with a Simplified AGCM. *Monthly Weather Review*, *150*(1), 283–302. https://doi.org/10.1175/MWR-D-21-0174.1
Laloyaux, P., Balmaseda, M., Dee, D., Mogensen, K., & Janssen, P. (2015). A coupled data assimilation system for climate reanalysis. *Quarterly Journal of the Royal Meteorological Society*. https://doi.org/doi:10.1002/qj.2629
Lei, L., Whitaker, J. S., & Bishop, C. (2018). Improving Assimilation of Radiance Observations by Implementing Model Space Localization in an Ensemble Kalman Filter. *Journal of Advances in Modeling Earth Systems*, *10*(12), 3221–3232. https://doi.org/10.1029/2018MS001468
Menne, M. J., Durre, I., Korzeniewski, B., McNeal, S., Thomas, K., Yin, X., et al. (2012). Global Historical Climatology Network Daily (GHCN-Daily), Version 3.26. NOAA National Climatic Data Center. https://doi.org/10.7289/V5D21VHZ
Nerger, L., & Hiller, W. (2013). Software for ensemble-based data assimilation systems—Implementation strategies and scalability. *Computers & Geosciences*, *55*, 110–118. https://doi.org/10.1016/J.CAGEO.2012.03.026
Niu, G. Y., Yang, Z. L., Mitchell, K. E., Chen, F., Ek, M. B., Barlage, M., et al. (2011). The community Noah land surface model with multiparameterization options (Noah-MP): 1. Model description and evaluation with local-scale measurements. *Journal of Geophysical Research: Atmospheres*, *116*(D12), 12109. https://doi.org/10.1029/2010JD015139
Penny, S. G., Akella, S., Alves, O., Bishop, C. H., Buehner, M., Chevallier, M., et al. (2017). *Coupled Data Assimilation for Integrated Earth System Analysis and Prediction: Goals, Challenges and Recommendations*. Geneva. Retrieved from https://www.wmo.int/pages/prog/arep/wwrp/new/documents/Final_WWRP_2017_3_27_July.pdf
Pham, D. T., Verron, J., & Roubaud, M. C. (1998). A singular evolutive extended Kalman filter for data assimilation in oceanography. *Journal of Marine Systems*, *16*,

323–340.
Poterjoy, J. (2016). A Localized Particle Filter for High-Dimensional Nonlinear Systems. *Monthly Weather Review*, *144*(1), 59–76. https://doi.org/10.1175/MWR-D-15-0163.1
de Rosnay, P., Balsamo, G., Albergel, C., Muñoz-Sabater, J., & Isaksen, L. (2012). Initialisation of Land Surface Variables for Numerical Weather Prediction. *Surveys in Geophysics 2012 35:3*, *35*(3), 607–621. https://doi.org/10.1007/S10712-012-9207-X
Saha, S., Nadiga, S., Thiaw, C., Wang, J., Wang, W., Zhang, Q., et al. (2006). The NCEP Climate Forecast System. *Journal of Climate*, *19*, 3483–3517. https://doi.org/10.1175/JCLI3812.1
Saha, S., Moorthi, S., Pan, H. L., Wu, X., Wang, J., Nadiga, S., et al. (2010). The NCEP climate forecast system reanalysis. *Bulletin of the American Meteorological Society*, *91*, 1015–1057. https://doi.org/10.1175/2010BAMS3001.1
Skamarock, W. C., Klemp, J. B., Duda, M. G., Fowler, L. D., Park, S. H., & Ringler, T. D. (2012). A Multiscale Nonhydrostatic Atmospheric Model Using Centroidal Voronoi Tesselations and C-Grid Staggering. *Monthly Weather Review*, *140*(9), 3090–3105. https://doi.org/10.1175/MWR-D-11-00215.1
Sluka, T. C., Penny, S. G., Kalnay, E., & Miyoshi, T. (2016). Assimilating atmospheric observations into the ocean using strongly coupled ensemble data assimilation. *Geophysical Research Letters*, *43*(2), 752–759. https://doi.org/10.1002/2015GL067238
UFS. (2023). Unified Forecast System. National Oceanic and Atmospheric Administration. Retrieved from https://github.com/ufs-community/
Wang, X., Chipilski, H. G., Bishop, C. H., Satterfield, E., Baker, N., & Whitaker, J. S. (2021). A Multiscale Local Gain Form Ensemble Transform Kalman Filter (MLGETKF). *Monthly Weather Review*, *149*(3), 605–622. https://doi.org/10.1175/MWR-D-20-0290.1
Whitaker, J. S., & Hamill, T. M. (2012). Evaluating methods to account for system errors in ensemble data assimilation. *Monthly Weather Review*, *140*(9), 3078–3089. https://doi.org/10.1175/MWR-D-11-00276.1
Zhang, F., Snyder, C., & Sun, J. (2004). Impacts of initial estimate and observation availability on convective-scale data assimilation with an ensemble Kalman filter. *Monthly Weather Review*, *132*(5), 1238–1253. https://doi.org/https://doi.org/10.1175/1520-0493(2004)132<1238:IOIEAO>2.0.CO;2
Zhang, L., Montuoro, R., McKeen, S. A., Baker, B., Bhattacharjee, P. S., Grell, G. A., et al. (2022). Development and evaluation of the Aerosol Forecast Member in the National Center for Environment Prediction (NCEP)'s Global Ensemble Forecast System (GEFS-Aerosols v1). *Geoscientific Model Development*, *15*(13), 5337–5369. https://doi.org/10.5194/GMD-15-5337-2022

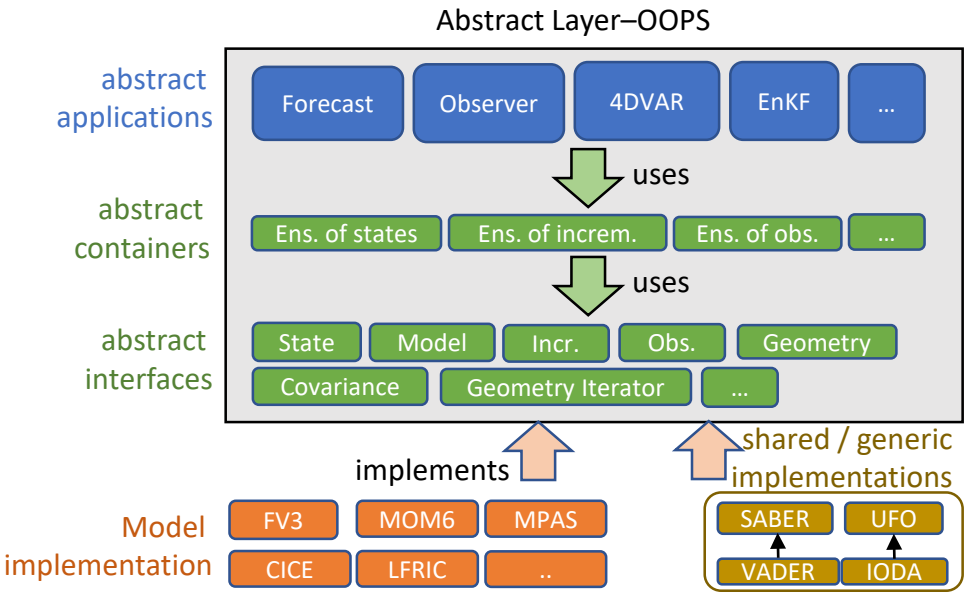

Figure 1: Separation of concerns in JEDI. Image adopted from JEDI academy slides (http://academy.jcsda.org/2021-10/slides/2021-06-21-WhyOOPSJEDI.pptx). Model implementation acronyms correspond to different model implementations and are defined in the main text of the paper. Other acronyms include: OOPS—Object Oriented Prediction System; SABER—System Agnostic Background Error Representation; UFO—Unified Forward Operator; VADER—Variable Derivation Repository; and IODA--Interface for Observational Data Access.

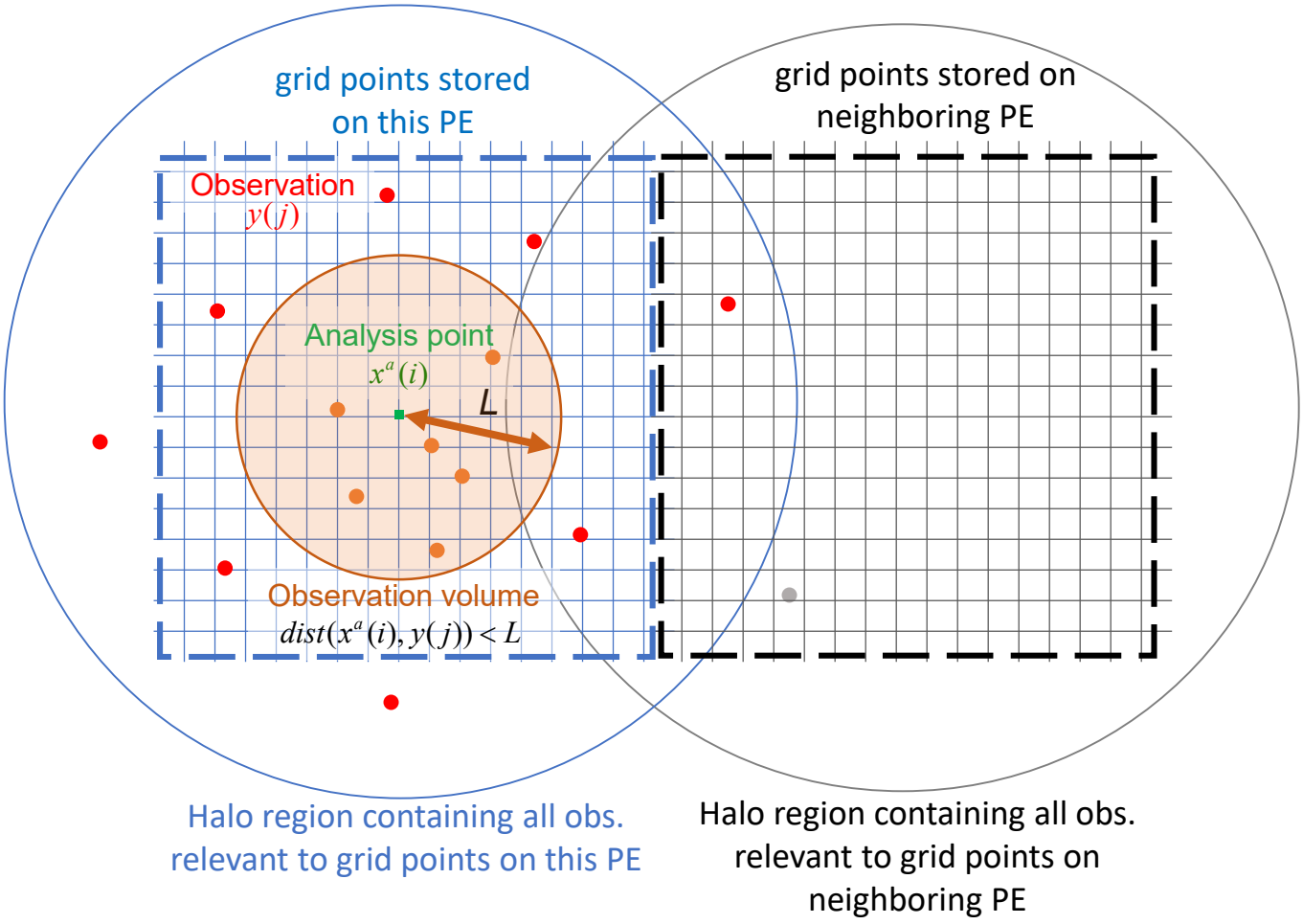

Figure 2: Local volume solver domain decomposition. Blue grid: grid points stored on the current (this) processing element (PE). Black grid: grid points on the neighboring PE. Green square indicates the updated, analysis point x(i). Dots indicate observations: (orange) observations used in the analysis; (red) observations that are needed to update all points on this PE; (gray) other observations that are not needed to update grid points on this PE. Circles indicate: (orange) the area used to collect local observations for the current local analysis, (blue) halo region that contains all possible observation locations needed to update grid points on this PE, (gray) observations halo region that contains all possible observation locations needed to update grid points on neighboring PE.

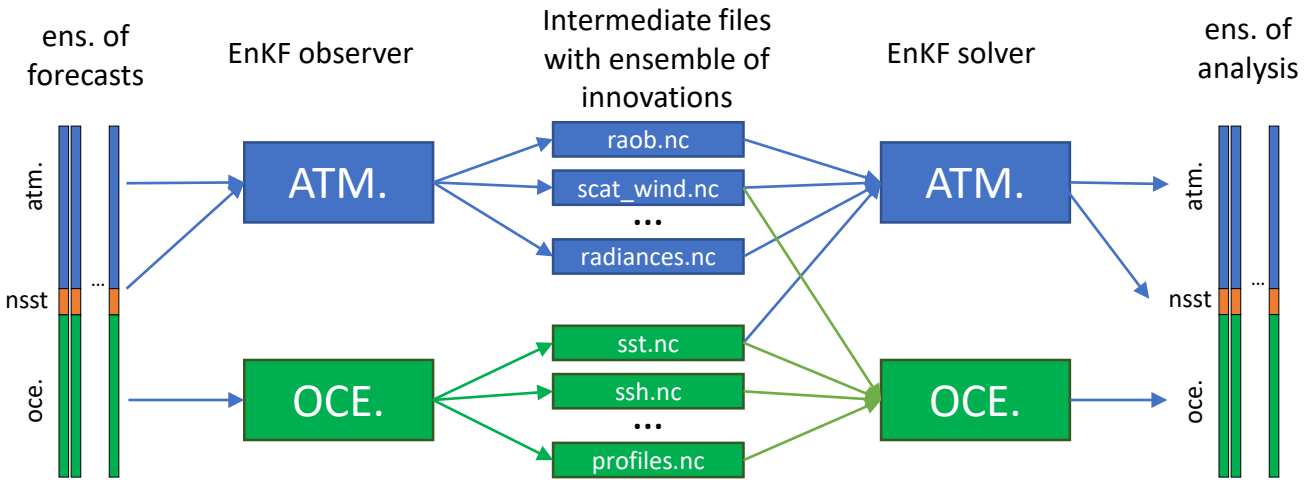

Figure 3: Example of interface solver implementation in JEDI framework. A pair of LETKF observers are run first obtaining ensemble observations of all atmospheric and oceanic variables. In this illustration, the near surface SST forecast (nsst) from the diurnal layer model is used as a first guess by the atmospheric observer for observations of the surface-sensitive radiances. Ocean state is used by the ocean observer to estimate the forecast of SST retrievals. Then a pair of LETKF solvers is executed one producing analysis on the atmospheric and the second on the oceanic grids. Each solver can use a combination of atmospheric and oceanic observations. In this example atmospheric solver is tasked to produce the analysis of the nsst variable as it has more complete picture of atmospheric observations that can potentially influence the nsst estimate.

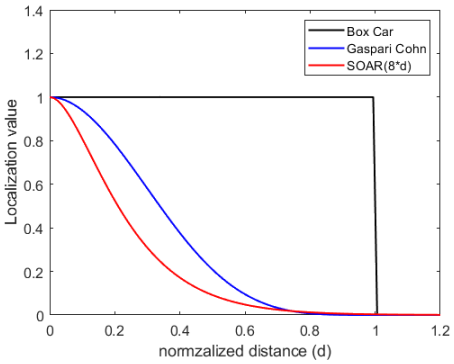

Figure 4: Localization as a function of normalized distance and localization type. For illustration purposes, distance was scaled by a factor of 8 before the SOAR function was calculated to achieve a near zero value for normalized distance of 1 (e.g. the SOAR function value is 0.003 for the distance of 8).

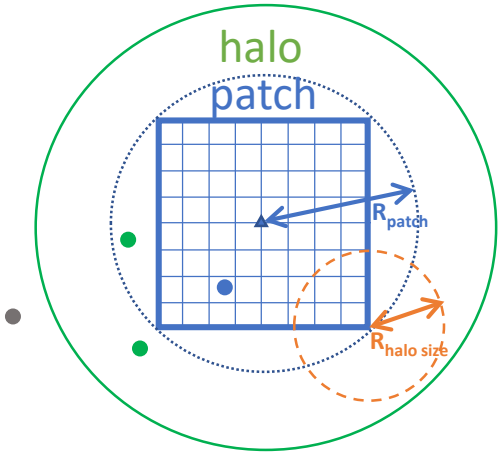

Figure 5: Halo distribution of observations. Shown are (blue grid) the grid points stored on this PE. (Blue triangle) halo center computed as a center of the grid points. (Blue circle) patch radius R_{patch} that encloses all the grid points on this PE; (Orange circle) halo size radius $R_{halo\ size}$ that is added to the patch circle. In practice, $R_{halo\ size}$ is related to the localization radius. (Green circle) the total halo circle that is computed as a summation of $R_{patch} + R_{halo\ size}$. Observations can then be divided in to patch observations that exclusively belong to this PE based on the shortest distance to the patch center (the blue dot). Halo observations (green dots) that are within the green total halo radius. And observations that are not stored on this PE because they are outside of the total halo radius (gray dots).

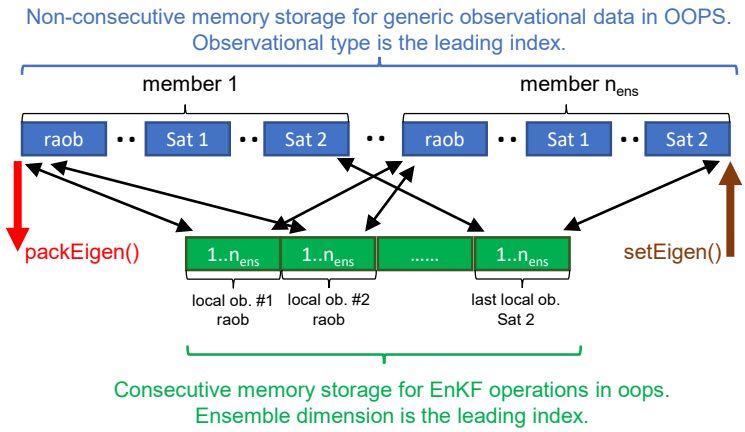

Figure 6: packEigen and setEigen methods operating on observational data in OOPS. Top row present data stored in generic containers for observational data in OOPS. Observational data stored as vectors of vectors, with each blue rectangle stored as contiguous data but with potential extraneous memory between blocks. Bottom row shows contiguous storage of ensembles of local observational data. The entire record is contiguous with the ensemble dimension as a leading index.

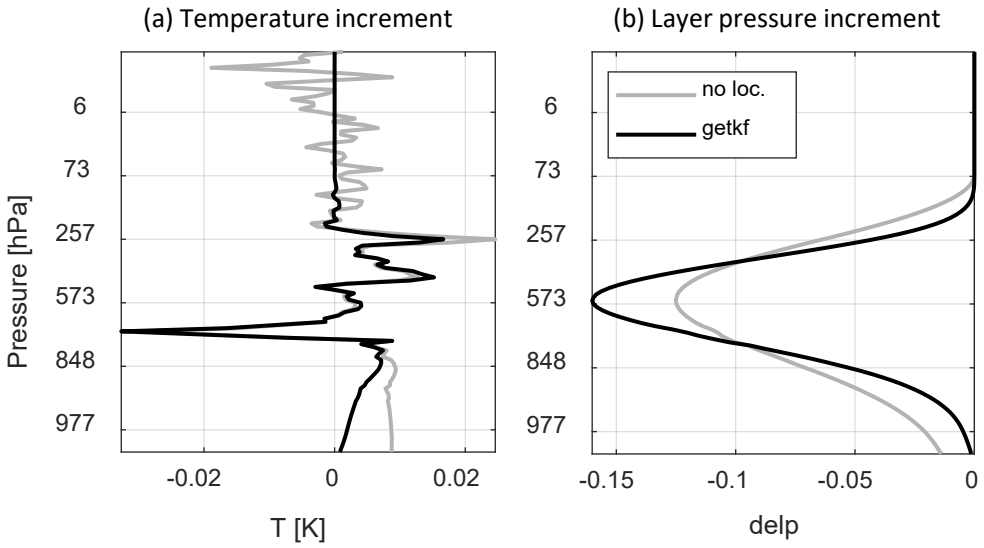

Figure 7: Increment to the vertical column of atmospheric temperature (a) and layer pressure (b) from assimilation of single surface pressure observation at 40.5°N, 160.5°E and innovation of 6 hPa. Background ensemble valid for 2015-12-05 18:00Z. GETKF vertical localization was used with localization scale of 30 levels and 10 eigen vectors capturing 96% of the variance.

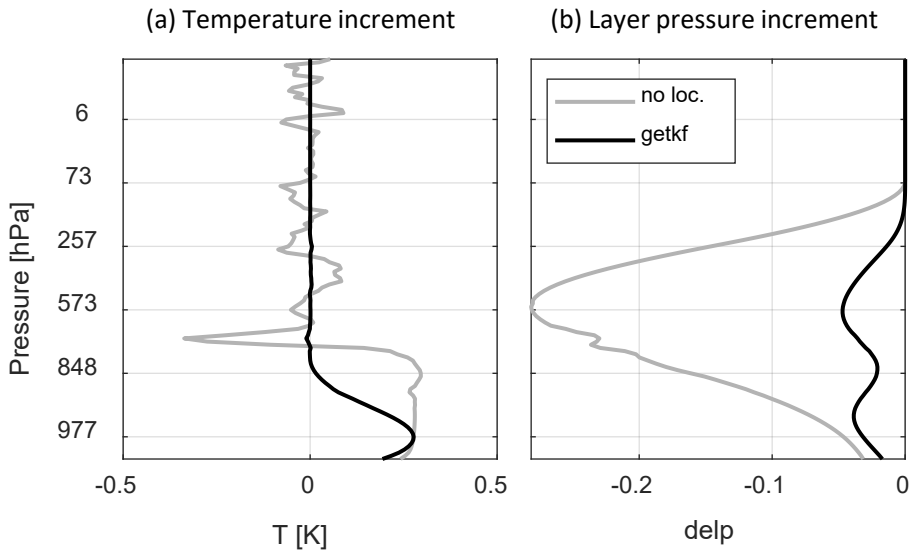

Figure 8: Increment to the vertical column of atmospheric temperature (a) and layer pressure (b) from assimilation of single temperature observations at 40.5°N, 160.5°E, 950 hPa, and innovation of 1 K. Background ensemble valid for 2015-12-05 18:00Z. GETKF vertical localization was used with localization scale of 30 levels and 10 eigen vectors capturing 96% of the variance.

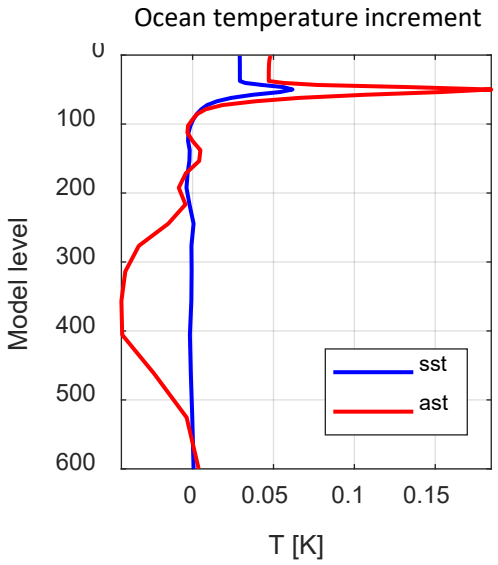

Figure 9: Increment to the vertical column of ocean temperature from assimilation of a single atmospheric temperature observations at 27.5°S, 154.5°W, 950 hPa (red) and sea surface temperature observation at the same horizontal location (blue). Both with the innovation of 1 K. Background ensemble valid for 2015-12-05 18:00Z. No vertical localization was used.

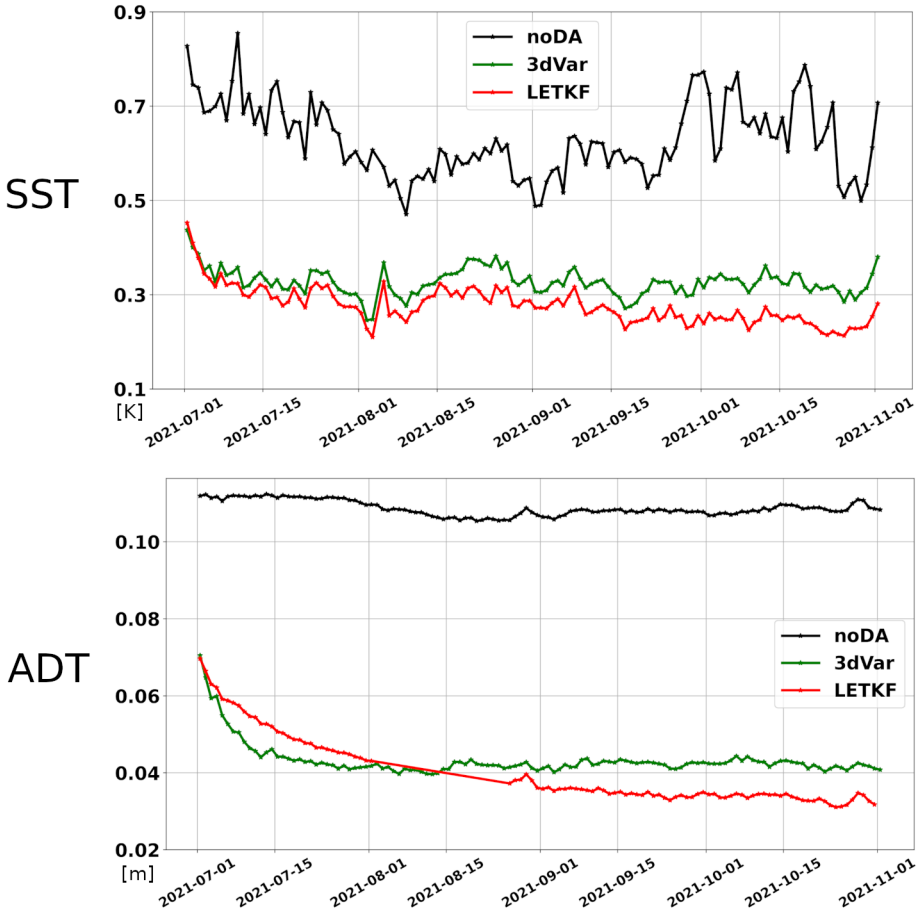

Figure 10: RMSE for the 24-hour forecast of (a) SST and (b) ADT observations. Colors indicate: (black) no data assimilation, (green) 3DVAR algorithm, and (red) LETKF.

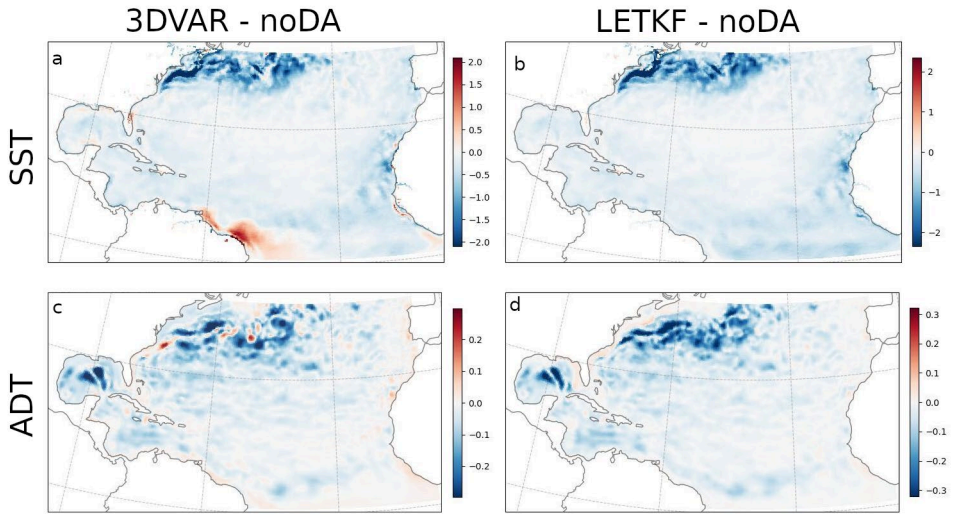

Figure 11: RMSE for the difference between run with no data assimilation and 3DVAR (left column, panels a and c) and LETKF (right column, panels b and d). Top row shows differences for SST (panels a and b) and the bottom row shows differences for ADT (panels c, d).

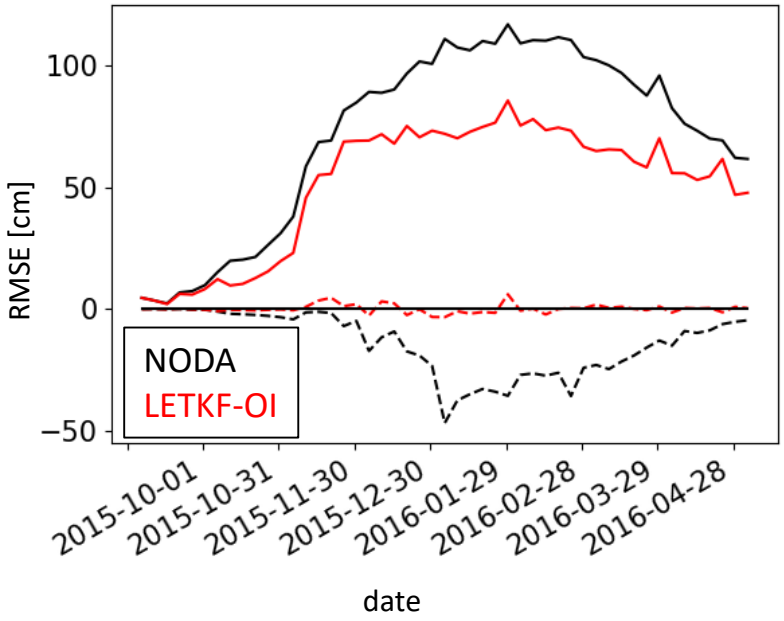

Figure 12: RMSE for the 6-hour forecast of the snow depth (GHCN network). Colors indicate: (black) no data assimilation, and (red) LETKF-OI assimilation of snow depth measurements. Line style indicates: (solid) standard deviation of error (dashed) mean error.

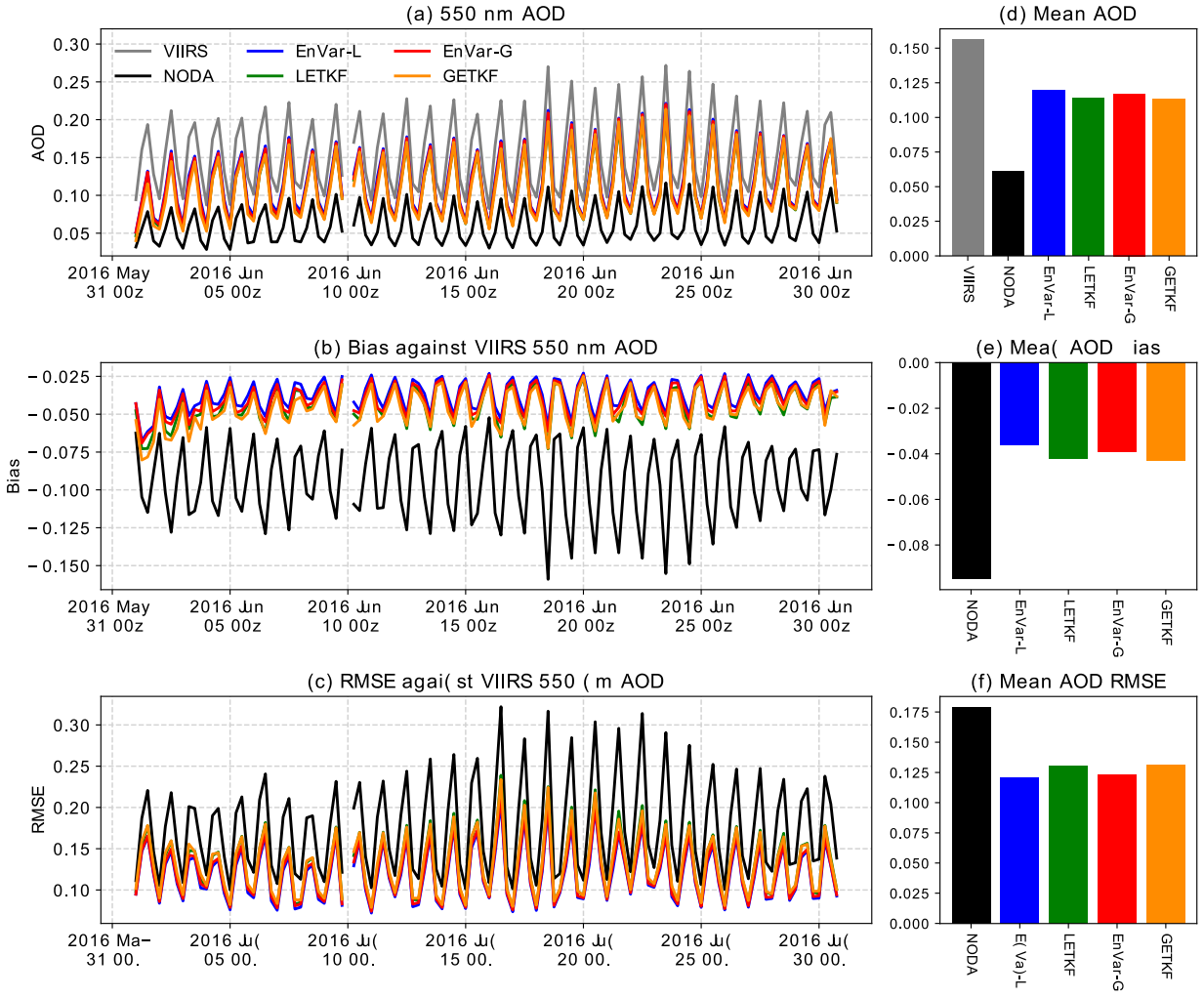

Figure 13: (a) Time-series of 550 nm AOD means from VIIRS (gray), no data assimilation six-hour forecast (black), the one-member control and ensemble mean analyses in the EnVar-L (blue and green, respectively) and EnVar-G (red and orange, respectively) experiments in a six-hourly interval in June 2016, and their corresponding (b) biases and (c) root-square-mean errors (RMSEs) against VIIRS AOD retrievals. (d)-(f) are the temporally means of the corresponding (a)-(c) over the last three weeks in June 2016.