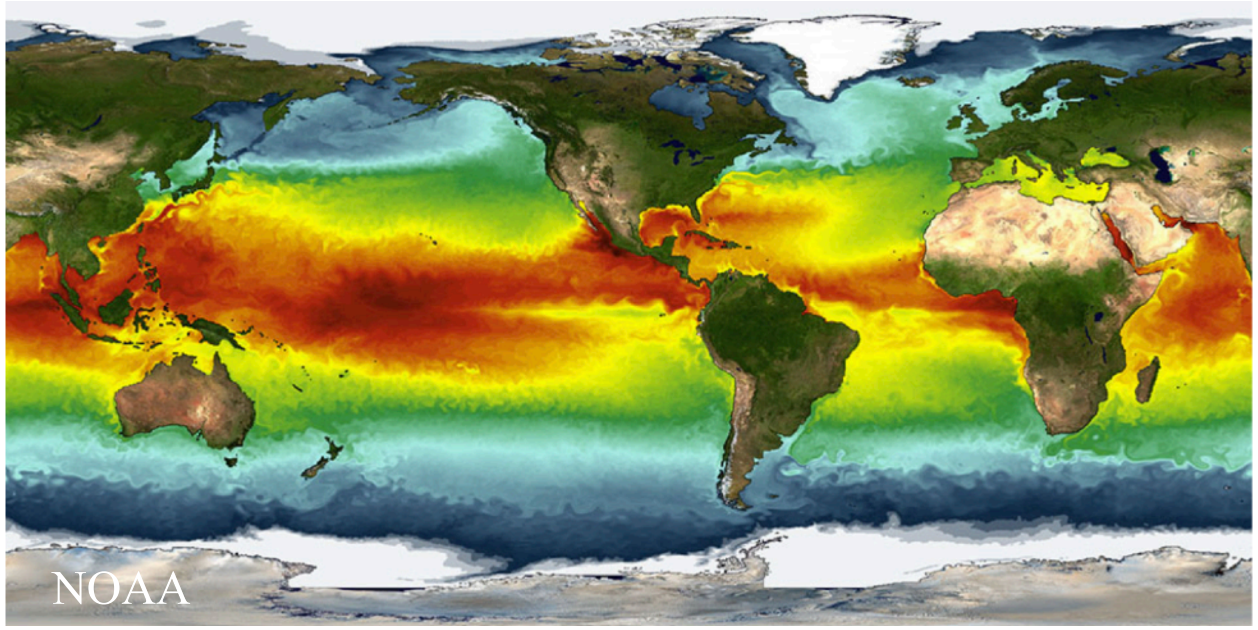
A Multi-Step Data Assimilation Algorithm for Multi-Layer Flow Fields

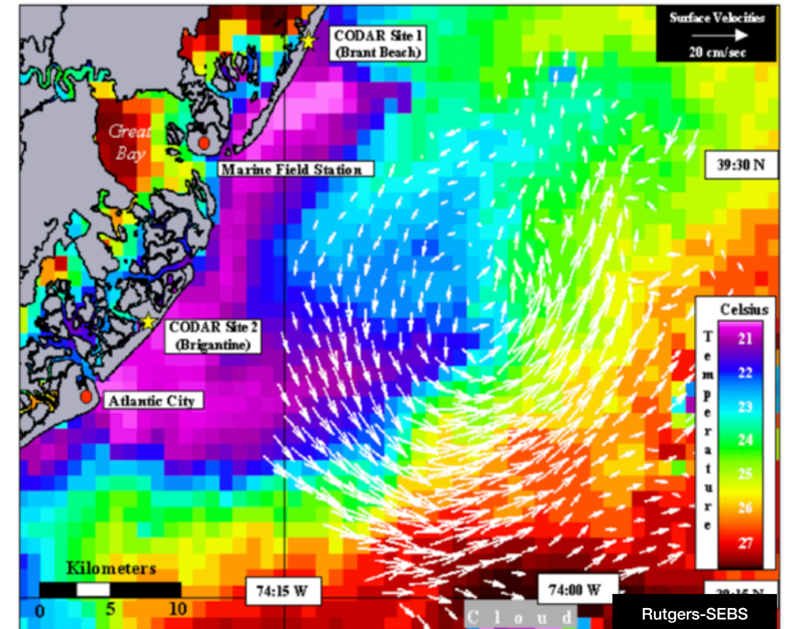
Zhongrui Wang[a], Nan Chen[a*], Di Qi[b]

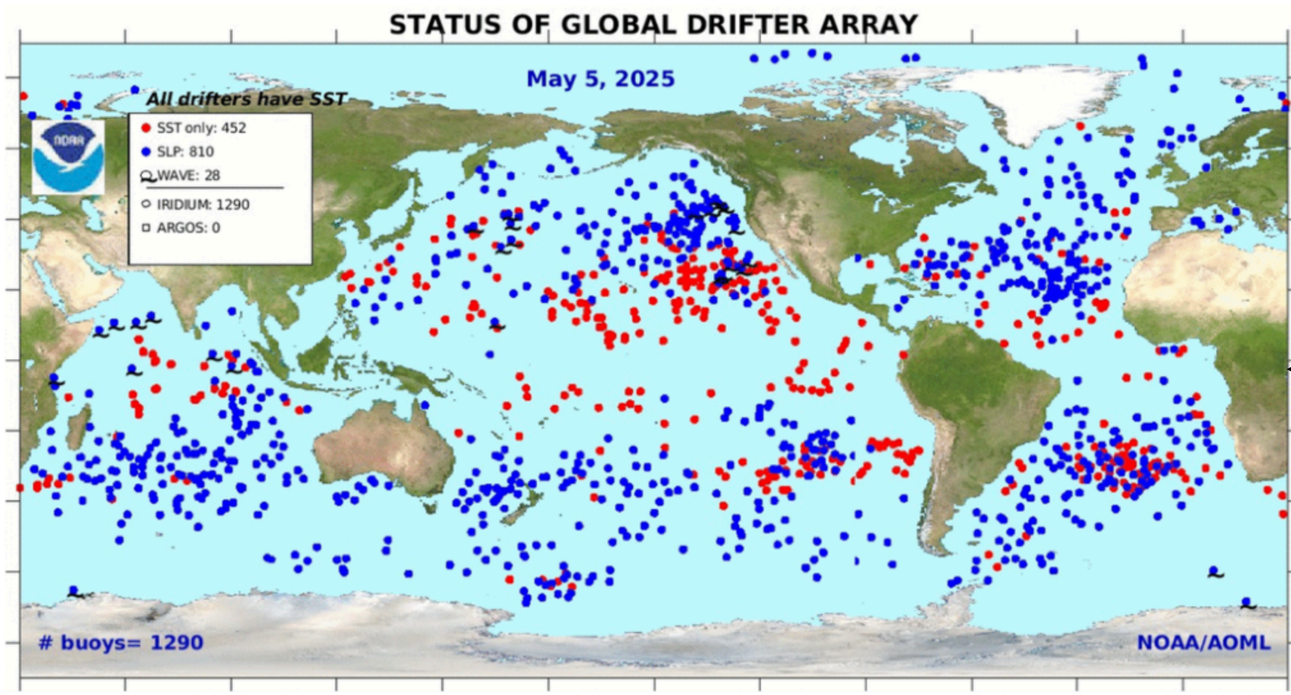
[a]*Department of Mathematics, University of Wisconsin–Madison*

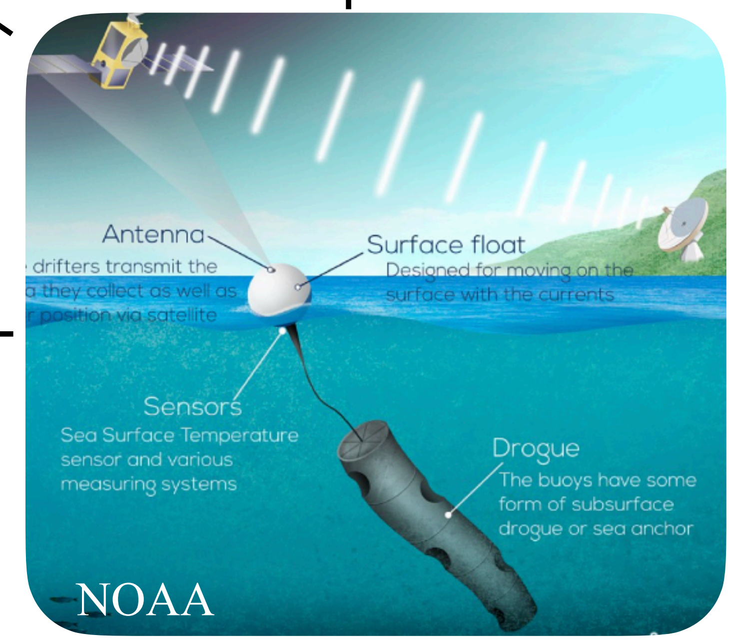
[a]*Department of Mathematics, Purdue University*

**corresponding author*

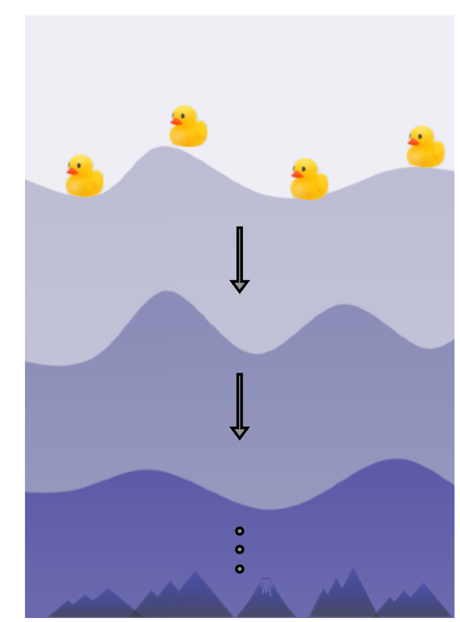






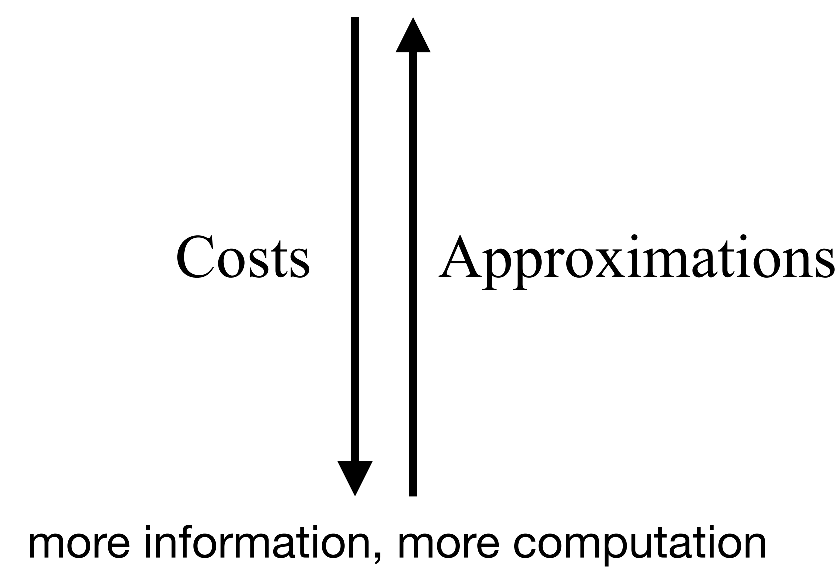


- **Data assimilation for multi-layer flow fields**: infer the states of all layers with only partial layers containing observations.
  - e.g., inferring deep ocean states with surface observations; inferring upper atmospheric states with near-surface observations
- **Challenges:** how to effectively propagate information from the observed layer to other layers in highly nonlinear turbulent flow systems.



DA for multi-layer flow fields

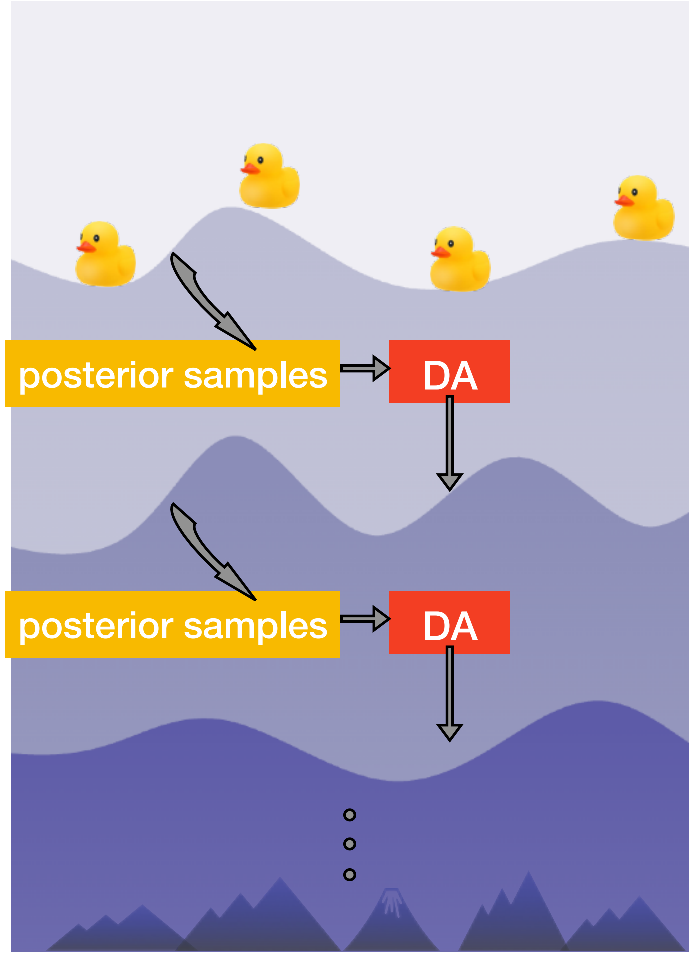
- **Linearly projects corrections to other layers using regression:**
  - Optimal interpolation (OI) methods (Molcard et al, 2005).
  - Ensemble Kalman Filter (EnKF)-based methods (Apte et al, 2008; Sun & Penny, 2019; Chen et al, 2022): corrections from the observed layer are projected to other layers linearly by the Kalman filter, but then corrections can propagate across layers through the nonlinear forecast model.
- **Nonlinear DA methods:**
  - Rank Histogram filter (Anderson, 2020)
  - Particle filters (van Leeuwen, 2009)
  - MCMC (Apte et al., 2008)
  - Conditional Gaussian Filter (Chen and Majda, 2016)



- **Goal**: a DA scheme that can characterize the nonlinear coupling across layers for effective information propagate, while maintaining computational efficiency.

Ideas: multi-step nonlinear DA for multi-layer flow

- **Multi-step**
  - We have an imperfect but useful DA scheme — not fully nonlinear but capable of propagating corrections from the observed layer to some adjacent layers.
  - After filtering the observed layer, sample from its posterior to generate pseudo-observations. Perform DA again using these pseudo-observations to improve the estimate of adjacent layers (as a single step).
  - Recursively apply this procedure to all layers (multi-step).



- **Nonlinear DA scheme for a single step**
  - In this work, we consider the conditional Gaussian nonlinear systems (CGNSs), in which
    - Nonlinearity of observed variables are allowed and characterized analytically in DA.
    - Analytic formulae for posterior mean and covariance exists.

Review of discrete-time DA and continuous-time DA

Kalman Filter vs Kalman-Bucy Filter

Kalman Filter

$$\mathbf{x}_{n+1} = \mathbf{M}\mathbf{x}_n + \Sigma_{\mathbf{x}}\varepsilon_{\mathbf{x},n}$$
$$\mathbf{y}_{n+1} = \mathbf{H}\mathbf{x}_{n+1} + \Sigma_{\mathbf{y}}\varepsilon_{\mathbf{y},n+1},$$
$$\Delta t = t_{n+1} - t_n$$

Posterior $\mathbf{x}_n|\mathbf{y}_n \sim \mathcal{N}(\mathbf{x}^a, \mathbf{P}^a)$

$$\mathbf{x}_n^a = \mathbf{x}_n^f + \mathbf{K}_n(\mathbf{y}_n - \mathbf{H}\mathbf{x}_n^f)$$
$$\mathbf{P}_n^a = (\mathbf{I} - \mathbf{K}_n\mathbf{H})\mathbf{P}_n^f,$$
$$\mathbf{K}_n = \mathbf{P}_n^f\mathbf{H}^\top(\mathbf{H}\mathbf{P}_n^f\mathbf{H}^\top + \mathbf{R}).$$

Kalman-Bucy Filter

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{M}\mathbf{x} + \Sigma_{\mathbf{x}}\dot{\mathbf{W}}_{\mathbf{x}}$$
$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{H}\mathbf{x} + \Sigma_{\mathbf{y}}\dot{\mathbf{W}}_{\mathbf{y}},$$

Posterior $\mathbf{x}(t)|\mathbf{y}(s), 0 \le s \le t \sim \mathcal{N}(\chi(t), \mathbf{P}(t))$.

$$\frac{\mathrm{d}\chi}{\mathrm{d}t} = \mathbf{M}\chi + \mathbf{K}(t)\left(\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} - \mathbf{H}\chi\right)$$
$$\frac{\mathrm{d}\mathbf{P}}{\mathrm{d}t} = \mathbf{M}\mathbf{P} + \mathbf{P}\mathbf{M}^\top - \mathbf{P}\mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H}\mathbf{P} + \mathbf{Q},$$
$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}^\top\mathbf{R}^{-1}$$

- What's the connection/difference between these two formulations? Can we simply take $\Delta t \to 0$ to get a KBF from KF, or discretize the KBF to get a KF?

Review of discrete-time DA and continuous-time DA

Kalman Filter vs Kalman-Bucy Filter

Kalman Filter

$$\mathbf{x}_{n+1} = \mathbf{M}\mathbf{x}_n + \Sigma_{\mathbf{x}}\varepsilon_{\mathbf{x},n}$$

$$\boxed{\mathbf{y}_{n+1}} = \mathbf{H}\mathbf{x}_{n+1} + \boxed{\Sigma_{\mathbf{y}}\varepsilon_{\mathbf{y},n+1}},$$

$$\Delta t = t_{n+1} - t_n$$

$$\Big\downarrow$$

$$\mathbf{y}_{n+1} - \mathbf{y}_n = \mathbf{H}(\mathbf{M} - \mathbf{I})\mathbf{x}_n + \mathbf{H}\Sigma_{\mathbf{x}}\varepsilon_{\mathbf{x},n} + \boxed{\Sigma_{\mathbf{y}}(\varepsilon_{\mathbf{y},n+1} - \varepsilon_{\mathbf{y},n})}.$$

Kalman-Bucy Filter

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{M}\mathbf{x} + \Sigma_{\mathbf{x}}\dot{\mathbf{W}}_{\mathbf{x}}$$

$$\boxed{\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t}} = \mathbf{H}\mathbf{x} + \boxed{\Sigma_{\mathbf{y}}\dot{\mathbf{W}}_{\mathbf{y}}},$$

$$\Big\downarrow$$

$$\mathbf{y}_{n+1} - \mathbf{y}_n = \mathbf{H}\mathbf{x}_n\Delta t + \boxed{\Sigma_{\mathbf{y}}\varepsilon_{\mathbf{y},n}\sqrt{\Delta t}},$$

- The observation noise of KF does not goes to zero as $\Delta t \to 0$, unlike the KBF. In fact, the KBF only looks like KF in the integral sense:

$$\boxed{\mathbf{y}(t0 + \Delta t) - \mathbf{y}(t0)} = \int_{t_0}^{t_0+\Delta t} \mathbf{H}\mathbf{x}(s)ds + \int_{t_0}^{t_0+\Delta t} \Sigma_{\mathbf{y}}\mathrm{d}\mathbf{W}_{\mathbf{y}}$$

  where the **observation increments of KBF** are the **"observations" of KF**.

- By solely looking at $\mathbf{y}$, the $\mathbf{y}$ in KF is very different from the $\mathbf{y}$ in KBF. Because the observation noise in **KF does not accumulate over time,** thus **independent** of the underlying dynamics, while the observation noise in **KBF gets involved** in the dynamics of $\mathbf{y}$.

Review of discrete-time DA and continuous-time DA

Kalman Filter vs Kalman-Bucy Filter

Kalman Filter

$$\mathbf{x}_{n+1} = \mathbf{M}\mathbf{x}_n + \Sigma_{\mathbf{x}}\varepsilon_{\mathbf{x},n}$$

$$\boxed{\mathbf{y}_{n+1}} = \mathbf{H}\mathbf{x}_{n+1} + \boxed{\Sigma_{\mathbf{y}}\varepsilon_{\mathbf{y},n+1}},$$

$\Delta t = t_{n+1} - t_n$ **Measurement error**

$$\mathbf{y}_{n+1} - \mathbf{y}_n = \mathbf{H}(\mathbf{M}-\mathbf{I})\mathbf{x}_n + \mathbf{H}\Sigma_{\mathbf{x}}\varepsilon_{\mathbf{x},n} + \boxed{\Sigma_{\mathbf{y}}(\varepsilon_{\mathbf{y},n+1} - \varepsilon_{\mathbf{y},n})}.$$

Kalman-Bucy Filter

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{M}\mathbf{x} + \Sigma_{\mathbf{x}}\dot{\mathbf{W}}_{\mathbf{x}}$$

$$\boxed{\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t}} = \mathbf{H}\mathbf{x} + \boxed{\Sigma_{\mathbf{y}}\dot{\mathbf{W}}_{\mathbf{y}}},$$

Observation model noise

$$\mathbf{y}_{n+1} - \mathbf{y}_n = \mathbf{H}\mathbf{x}_n\Delta t + \boxed{\Sigma_{\mathbf{y}}\varepsilon_{\mathbf{y},n}\sqrt{\Delta t}},$$

- The observation noise of KF does not goes to zero as $\Delta t \to 0$, unlike the KBF. In fact, the KBF only looks like KF in the integral sense:

$$\boxed{\mathbf{y}(t0+\Delta t) - \mathbf{y}(t0)} = \int_{t_0}^{t_0+\Delta t} \mathbf{H}\mathbf{x}(s)ds + \int_{t_0}^{t_0+\Delta t} \Sigma_{\mathbf{y}}\mathrm{d}\mathbf{W}_{\mathbf{y}}$$

  where the **observation increments of KBF** are the **"observations" of KF**.

- By solely looking at **y**, the **y** in KF is very different from the **y** in KBF. Because the observation noise in **KF does not accumulate over time,** thus **independent** of the underlying dynamics, while the observation noise in **KBF gets involved** in the dynamics of **y**.

Conditional Gaussian Nonlinear Systems:
A generalization to Kalman-Bucy Filter

$$\frac{\mathrm{d}\mathbf{u}_1}{\mathrm{d}t} = \mathbf{A}_0(\mathbf{u}_1, t) + \mathsf{A}_1(\mathbf{u}_1, t)\mathbf{u}_2 + \Sigma_1(\mathbf{u}_1, t)\dot{\mathbf{W}}_1,$$
$$\frac{\mathrm{d}\mathbf{u}_2}{\mathrm{d}t} = \mathbf{a}_0(\mathbf{u}_1, t) + \mathsf{a}_1(\mathbf{u}_1, t)\mathbf{u}_2 + \Sigma_2(\mathbf{u}_1, t)\dot{\mathbf{W}}_2,$$

where $\mathbf{u}_1 \in \mathbb{C}^{N_1}$ ares observed variables and $\mathbf{u}_2 \in \mathbb{C}^{N_2}$ are hidden variables.

- **Nonlinear & Non-Gaussian**: $\mathbf{A}_0$, $\mathbf{a}_0$, A_1, a_1, Σ_1, Σ_2 can nonlinearly depend on $\mathbf{u}_1$. Thus, the CGNS can be *highly nonlinear*, the marginal distributions of $\mathbf{u}_1$, $\mathbf{u}_2$ can be *strongly non-Gaussian*.
- **Conditional Gaussian**: Given an observed trajectory of $\mathbf{u}_1$, the posterior of $\mathbf{u}_2$ is Gaussian:

$$\mathbf{u}_2(t)|\mathbf{u}_1(s \le t) \sim \mathcal{N}\big(\boldsymbol{\mu}_2(t), \mathsf{R}_2(t)\big),$$

with mean $\boldsymbol{\mu}_2(t)$ and covariance $\mathsf{R}_2(t)$ solvable through the analytic formulae

$$\frac{\mathrm{d}\boldsymbol{\mu}_2}{\mathrm{d}t} = (\mathbf{a}_0 + \mathsf{a}_1\boldsymbol{\mu}_2) + \mathsf{R}_2\mathsf{A}_1^*(\Sigma_1\Sigma_1^*)^{-1}\left(\frac{\mathrm{d}\mathbf{u}_1}{\mathrm{d}t} - (\mathbf{A}_0 + \mathsf{A}_1\boldsymbol{\mu}_2)\right),$$
$$\frac{\mathrm{d}\mathsf{R}_2}{\mathrm{d}t} = \mathsf{a}_1\mathsf{R}_2 + \mathsf{R}_2\mathsf{a}_1^* + \Sigma_2\Sigma_2^* - (\mathsf{R}_2\mathsf{A}_1^*)(\Sigma_1\Sigma_1^*)^{-1}(\mathsf{A}_1\mathsf{R}_2).$$

CGNS gallery
- The noisy Lorenz model
- Boussinesq equation
- Rotating shallow water equations

…

[Chen and Majda, 2018]

A multi-step nonlinear DA based on CGNSs

Problem setup

- To make things more concrete, consider **a multi-layer flow system with surface Lagrangian tracer observations:**

$$\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{v}_1(\mathbf{x}_i, t) + \boldsymbol{\Sigma}_{\mathbf{x}}\dot{\mathbf{W}}_i, \qquad i = 1, \ldots, I$$

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = (\mathsf{L} + \mathsf{D})\mathbf{v} + \boxed{\mathbf{B}(\mathbf{v}, \mathbf{v})} + \mathbf{F} + \boldsymbol{\Sigma}_{\mathbf{v}}\dot{\mathbf{W}}_{\mathbf{v}},$$

Nonlinear quadratic form

where $\mathbf{x}_i = (x_i, y_i)^{\mathrm{T}}$ is the observed position of the ith tracer, and $\mathbf{v} = (\ldots, \mathbf{v}_\ell, \ldots)^{\mathrm{T}}, \ell = 1, \ldots, L$ is the unobserved flow velocity field that consists of planar velocities $\mathbf{v}_\ell = (u_\ell, v_\ell)$ of L layers.

The goal is to infer all layers of flow with the surface tracer observations.

- **Lagrangian DA** is more challenging than Euler DA, as the observation process is nonlinear (even with a linear flow model).

A multi-step nonlinear DA based on CGNSs

Multi-step DA

Consider a two-layer flow $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2)^{\mathrm{T}}$ with surface tracer observations $\mathbf{x}(s)$. We aim for the posterior

$$P_{\mathbf{V}(t)|\mathbf{X}(s), s\le t}\big(\mathbf{v}|\{\mathbf{x}(s)\}_{s\le t}\big), \tag{5}$$

1. The **first DA step** solves the **surface-layer** flow posterior $P\big(\mathbf{v}_1|\{\mathbf{x}(s)\}_{s\le t}\big)$ given tracer obs.
2. **Sample** from $P\big(\mathbf{v}_1|\{\mathbf{x}(s)\}_{s\le t}\big)$ to get $\{\mathbf{v}_1^{(n)}\}$, as pseudo-observations of the upper-layer flow.
3. The **second DA step** solves the **lower-layer** flow posterior $P\big(\mathbf{v}_2|\mathbf{v}_1^{(n)}, \{\mathbf{x}(s)\}_{s\le t}\big)$ for each sample. The ultimate posterior that combines N_s samples is

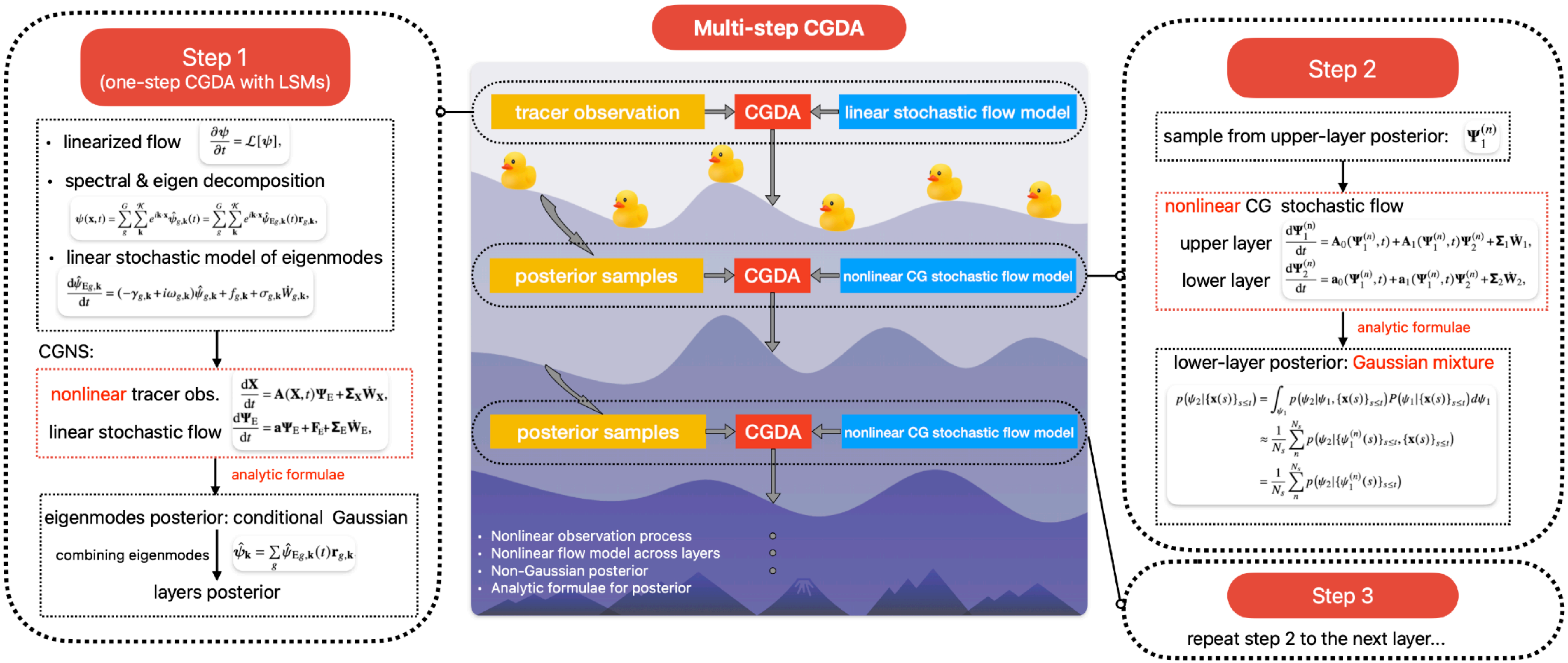
$$\begin{aligned}
P\big(\mathbf{v}_2|\{\mathbf{x}(s)\}_{s\le t}\big) &= \int_{\mathbf{v}_1} P\big(\mathbf{v}_1, \mathbf{v}_2|\{\mathbf{x}(s)\}_{s\le t}\big) d\mathbf{v}_1 \;\; \textit{(marginal probability)} \\
&= \int_{\mathbf{v}_1} P\big(\mathbf{v}_2|\mathbf{v}_1, \{\mathbf{x}(s)\}_{s\le t}\big) P\big(\mathbf{v}_1|\{\mathbf{x}(s)\}_{s\le t}\big) d\mathbf{v}_1 \;\; \textit{(conditional probability)} \\
&\boxed{\approx \frac{1}{N_s}\sum_{n}^{N_s} P\big(\mathbf{v}_2|\mathbf{v}_1^{(n)}, \{\mathbf{x}(s)\}_{s\le t}\big) \;\; (Monte - Carlo\ estimation)} \\
&\approx \frac{1}{N_s}\sum_{n}^{N_s} P\big(\mathbf{v}_2|\{\mathbf{v}_1^{(n)}(s)\}_{s\le t}, \{\mathbf{x}(s)\}_{s\le t}\big) \;\; (if\ conditioned\ on\ trajectory).
\end{aligned} \tag{6}$$

A multi-step nonlinear DA based on CGNSs

CG surrogate models for each DA step

- **Step 1:**
  - Fit a linear stochastic model (LSM) for the flow (drop nonlinear terms $\mathbf{B}(\mathbf{v}, \mathbf{v})$, using stochastic noises to approximate it instead.)
  - Apply CGDA to tracer observations and the flow LSM. (Step 1 is often trivial because the observations is strongly correlated with the surface layer.)
- **Step 2**:
  - Fit a conditional Gaussian nonlinear model for the flow (drop nonlinear terms of $\mathbf{v}_2$, using stochastic noises to approximate the simplification. But <u>keep the nonlinear terms of $\mathbf{v}_1$</u>.)

A multi-step nonlinear DA based on CGNSs



- The **multi-step DA procedure** can effectively improve information propagation across layers.
- By using the **CGNSs surrogate models**:
  - **Nonlinearity of the tracer / upper-layer flow** is preserved in model and characterized in DA.
  - **Analytic formulae** for posterior mean and covariance is available.
  - The lower-layer posterior distribution is a Gaussian mixture, which can be highly **non-Gaussian**.

$$\boxed{\approx \frac{1}{N_s}\sum_{n}^{N_s} P\left(\mathbf{v}_2|\mathbf{v}_1^{(n)}, \{\mathbf{x}(s)\}_{s\leq t}\right) \ (Monte-Carlo\ estimation)}$$

Numerical tests:
two-layer QG flow with surface tracer observations

$$\text{(Tracer)} \quad \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{v}_1(\mathbf{x}_i, t) + \boldsymbol{\Sigma}_{\mathbf{x}} \dot{\mathbf{W}}_i, \qquad i = 1, \ldots, I$$

$$\text{(Flow)} \quad \begin{aligned} &\frac{\partial q_1}{\partial t} + J(\psi_1, q_1) + \beta \frac{\partial \psi_1}{\partial x} + U_1 \frac{\partial}{\partial x} \nabla^2 \psi_1 + \frac{k_d^2}{2} \left(U_1 \frac{\partial \psi_2}{\partial x} - U_2 \frac{\partial \psi_1}{\partial x} \right) = -\nu \Delta^s q_1, \\ &\frac{\partial q_2}{\partial t} + J(\psi_2, q_2) + \beta \frac{\partial \psi_2}{\partial x} + U_2 \frac{\partial}{\partial x} \nabla^2 \psi_2 + \frac{k_d^2}{2} \left(U_2 \frac{\partial \psi_1}{\partial x} - U_1 \frac{\partial \psi_2}{\partial x} \right) = -\left(U_2 \frac{\partial h}{\partial x} + \kappa \nabla^2 \psi_2 \right) - \nu \Delta^s q_2. \end{aligned}$$

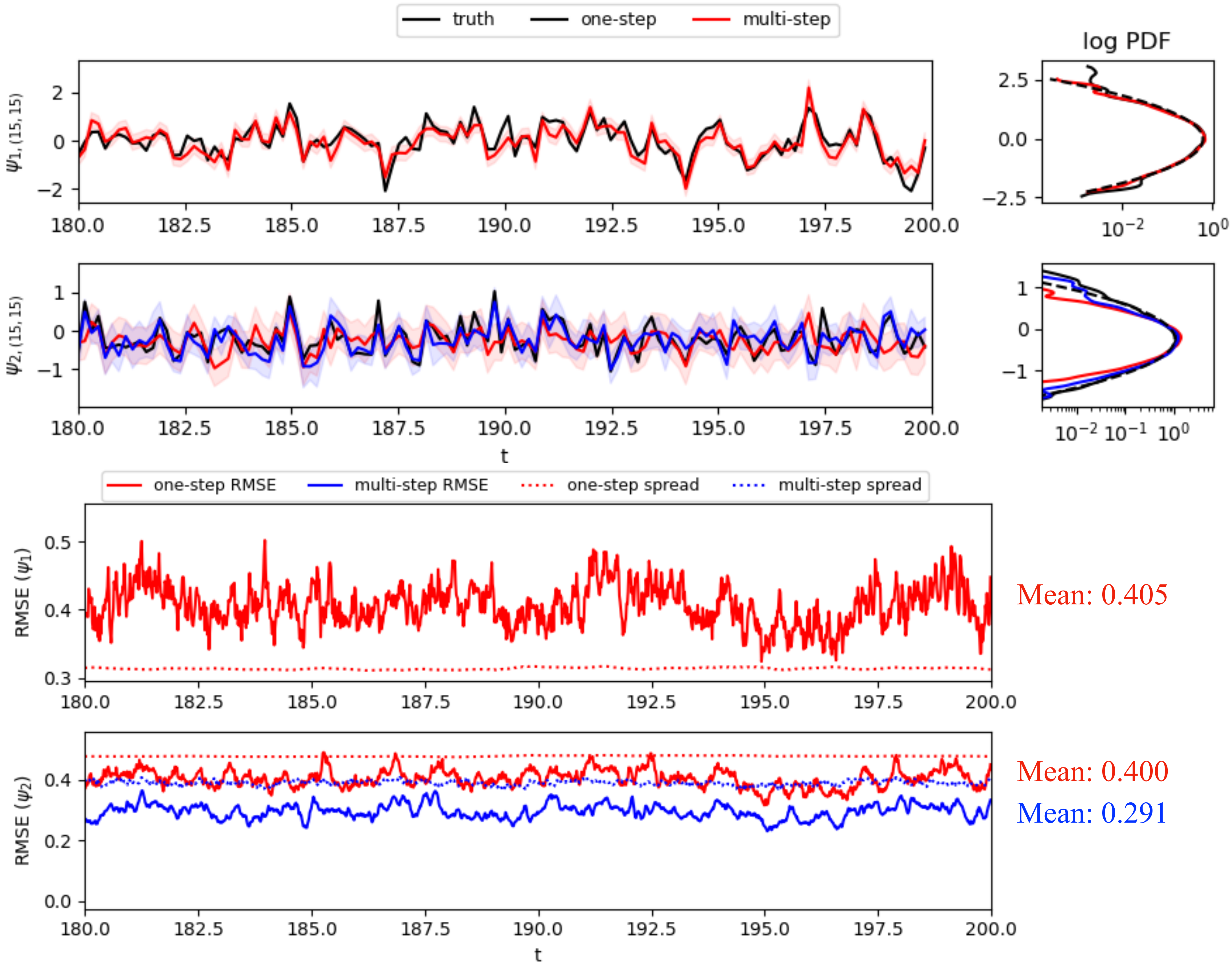
where $\mathbf{x}_i = (x_i, y_i)^{\mathrm{T}}$ is the observed position of the ith tracer, planar velocities $\mathbf{v}_\ell = (u_\ell, v_\ell)$ of L layers.

$q_1 = \nabla^2 \psi_1 + \frac{k_d^2}{2}(\psi_2 - \psi_1)$, $\quad q_2 = \nabla^2 \psi_2 + \frac{k_d^2}{2}(\psi_1 - \psi_2) + h$ is the potential vorticity.

- **Default DA setup:**
  - **Observations**: tracer positions $\mathbf{x}_i = (x_i, y_i)^{\mathrm{T}}$ of total $L = 256$ tracers only in the surface layer. Observation (model) noise $\sigma_x = \sigma_y = 0.1$. Observations are available every time step.
  - **Flow (QG)**: integration time step $dt = 0.002(\sim 4.4h)$; periodic domain $[0, 2\pi)^2$ $(\sim [0, 1257km)^2)$with 128×128 grids (16384 in total).
  - **Multi-step CGDA**: reduced order models use a spectral truncation radius of 16 (797 modes in total); default ensemble size $N_s = 16$.
- **DA with measurement errors (compared to EnKF):**
  - **Measurement error** $\epsilon_o \sim \mathcal{N}(0, \sigma_o^2)$, $\sigma_o = 0.0001$ (small measurement error, typical error scale of a GPS).
  - **EnKF***: ensemble size=40; constant multiplicative inflation and GC localization; full-resolution forecast model.

*EnKF is unstable for sparse, highly accurate observations (known as catastrophic filter divergence, Gottwald & Majda, 2013)

One-step CGDA vs multi-step CGDA

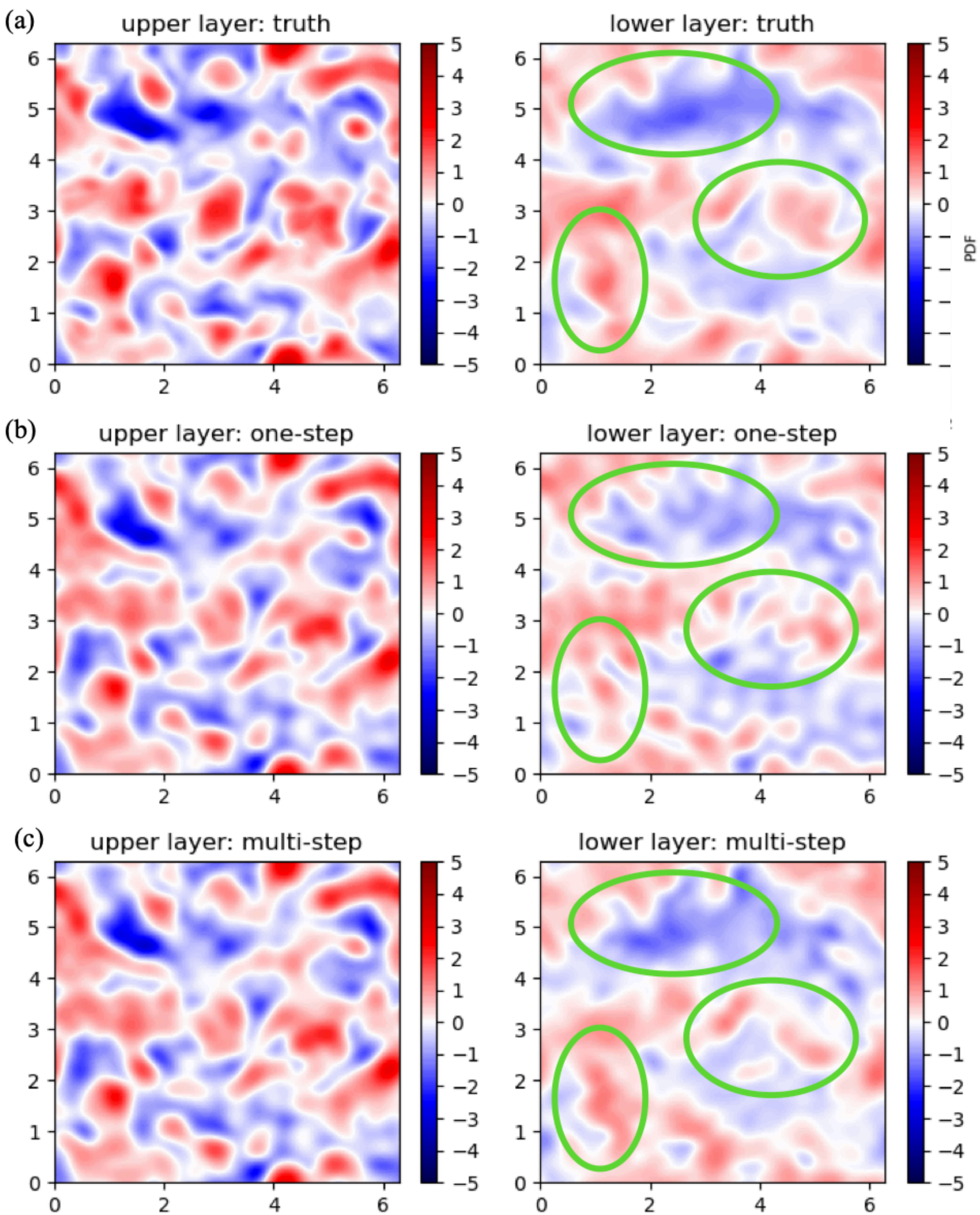


Mean: 0.405

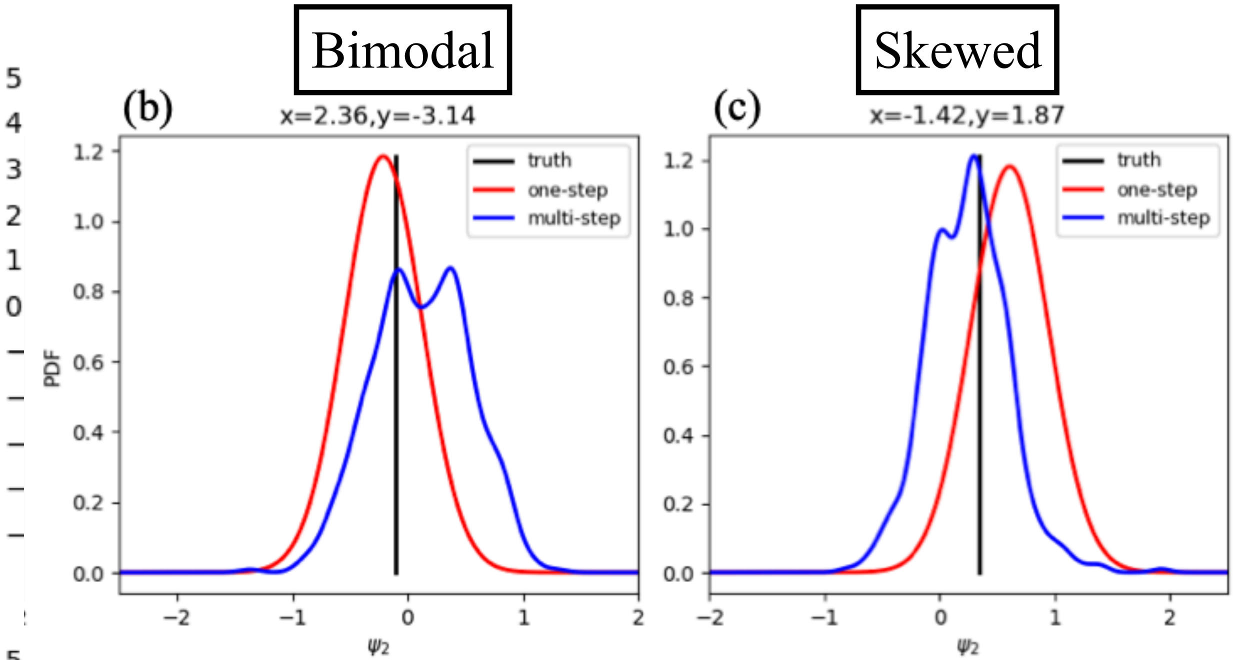
Mean: 0.400

Mean: 0.291

One-step CGDA vs multi-step CGDA



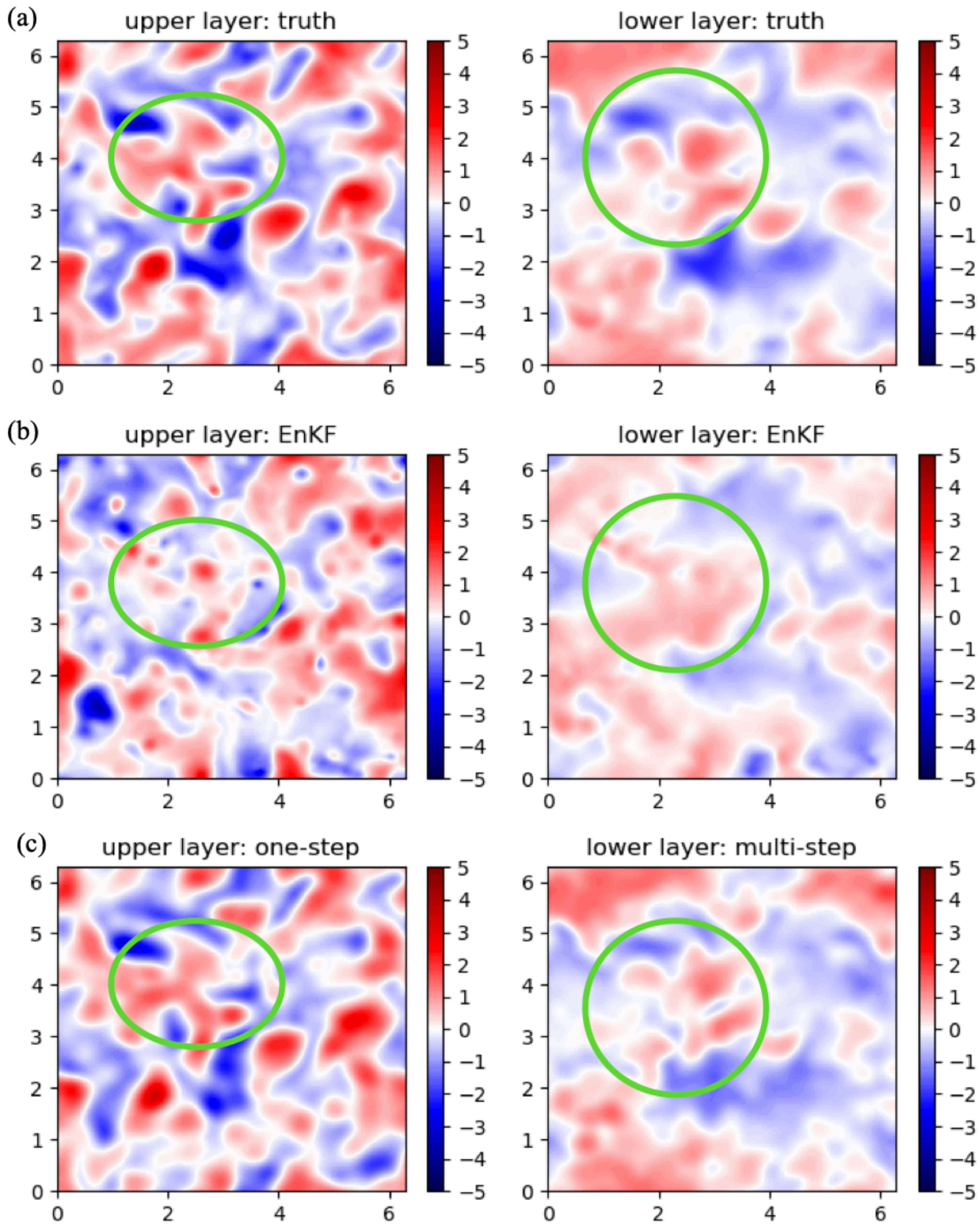
Snapshots of the truth and posterior mean



Non-Gaussian posterior distribution

- Compared to one-step CGDA, multi-step CGDA **better recovers the lower layer flow** by incorporating flow nonlinearities in the second step.
- Multi-step CGDA can capture **non-Gaussian posterior distributions** with Gaussian mixtures.

Multi-step vs EnKF (with small measurement error)



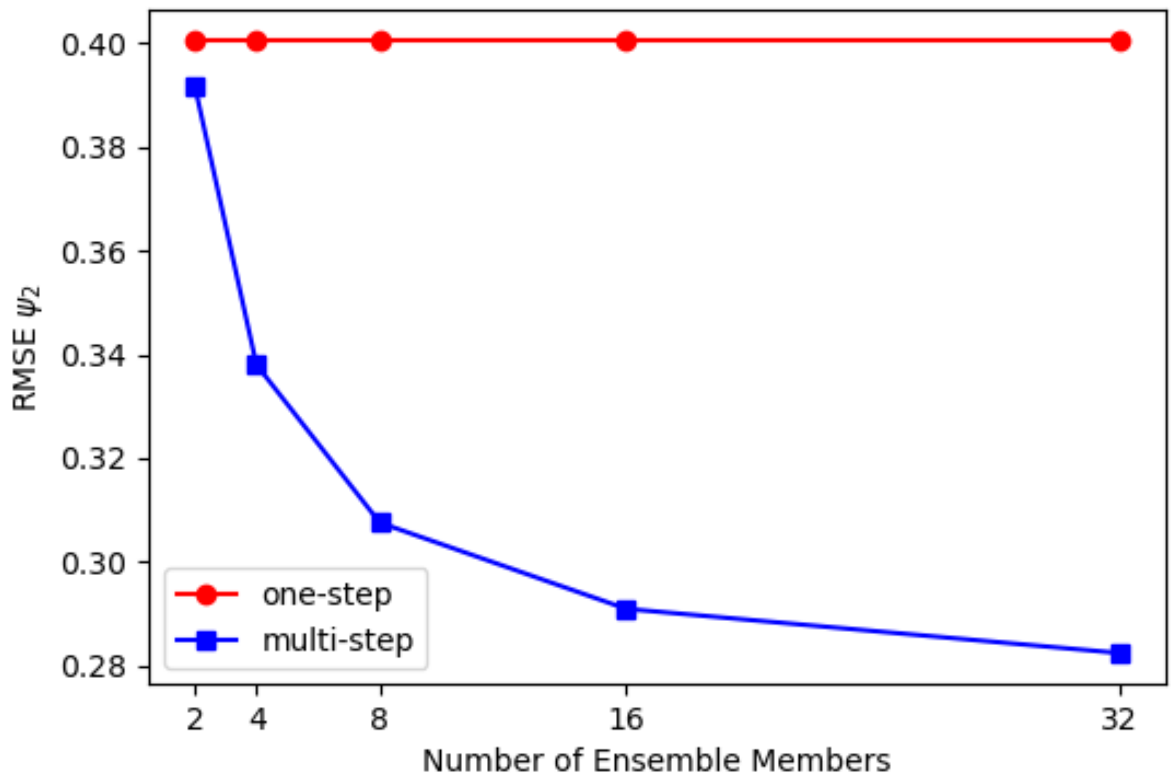
Time mean posterior RMSEs

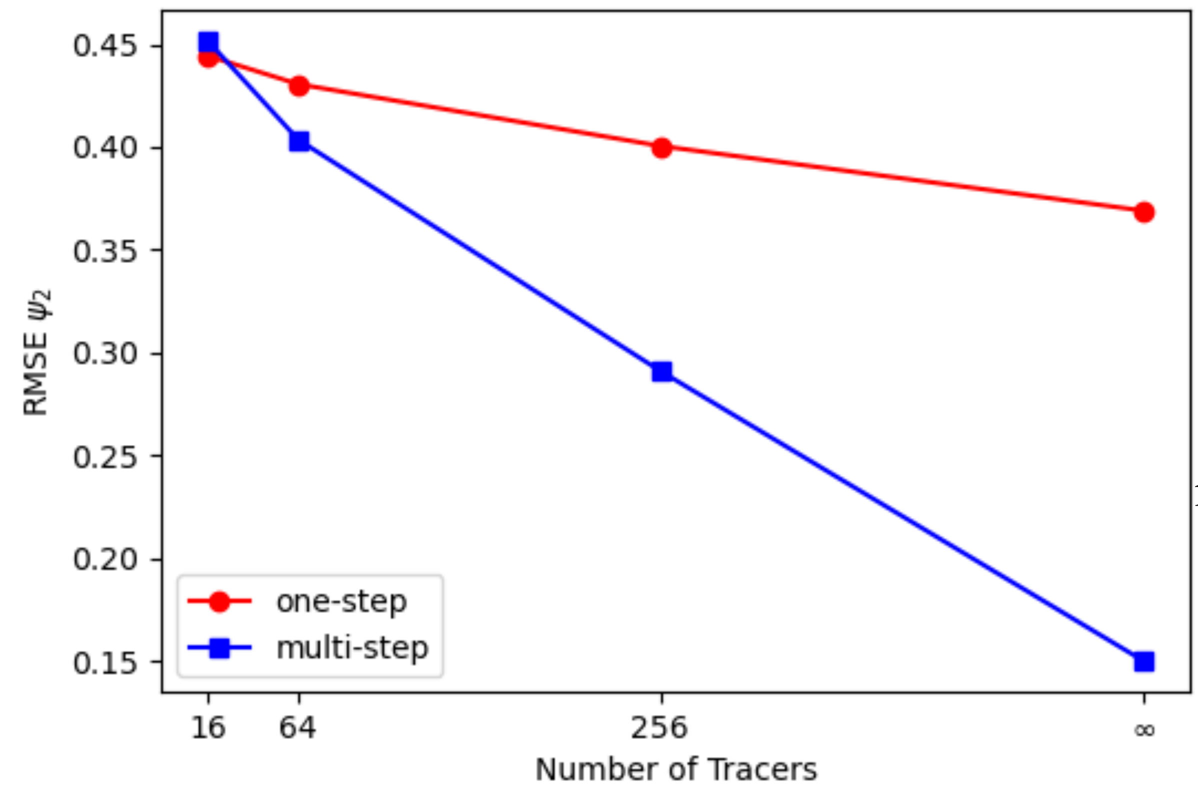
	One-step CGDA	Multi-step CGDA	EnKF
Upper layer	0.403	0.403	0.921
Lower layer	0.400	0.328	0.459

Multi-step CGDA w/ 16 ensemble members (unparalleled)
~ **6 times faster** than
EnKF w/ 40 ensemble members (paralleled)

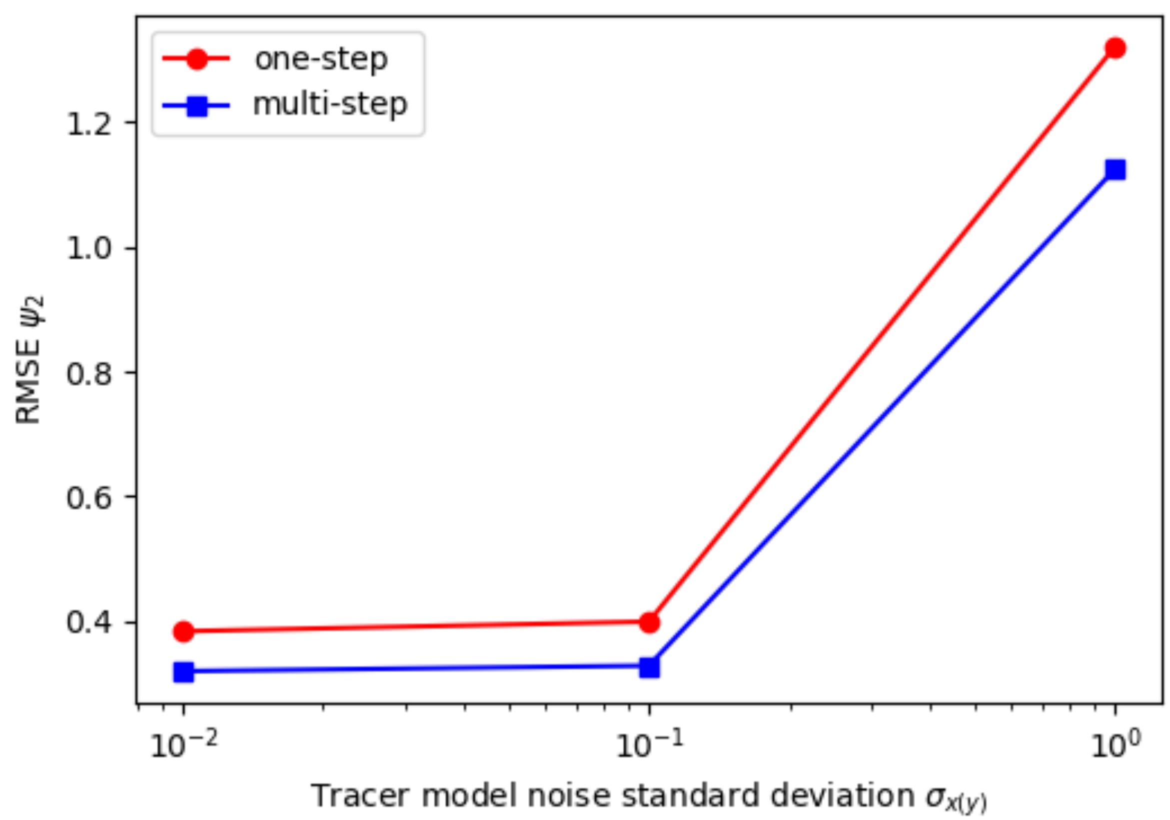
- EnKF is **unstable** for **frequent, sparse, accurate observations** (catastrophic filter divergence).
- Multi-step CGDA is more robust than EnKF under this setup, and obtains **smaller posterior RMSEs with fewer computational costs**.

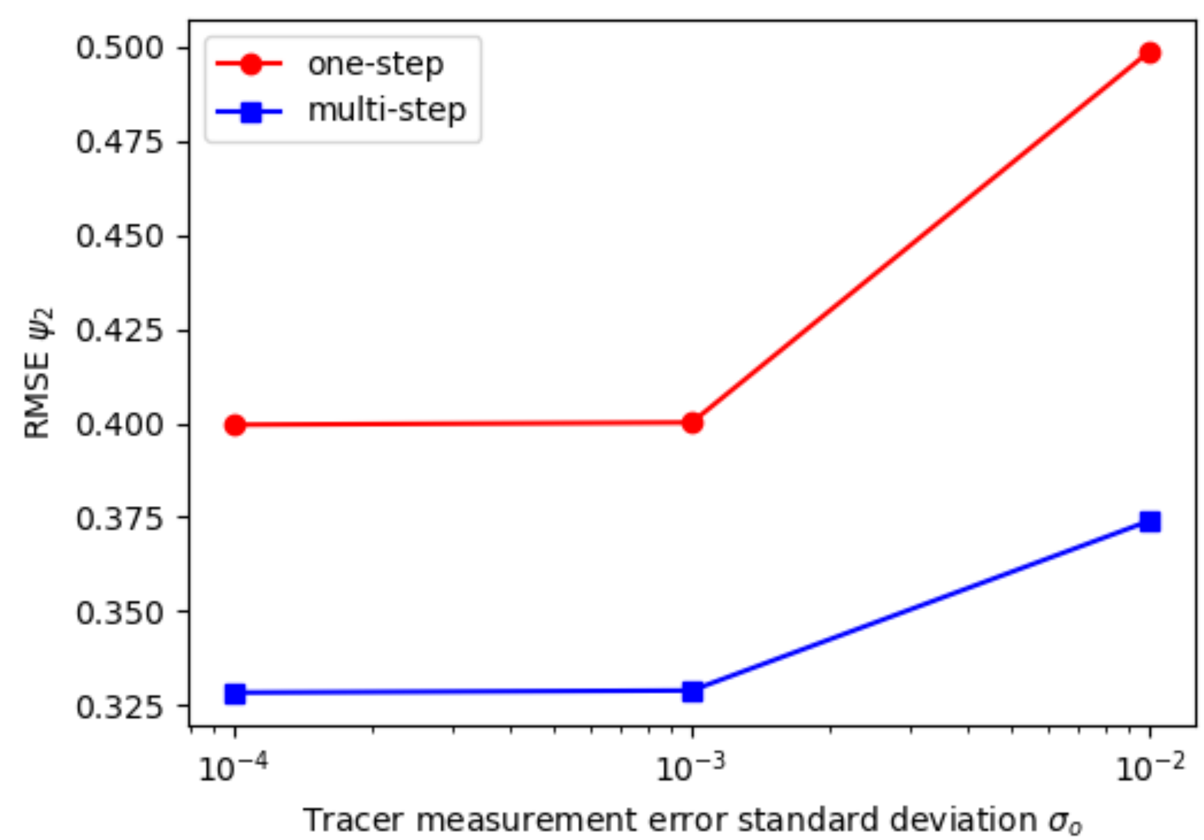
Sensitivity tests





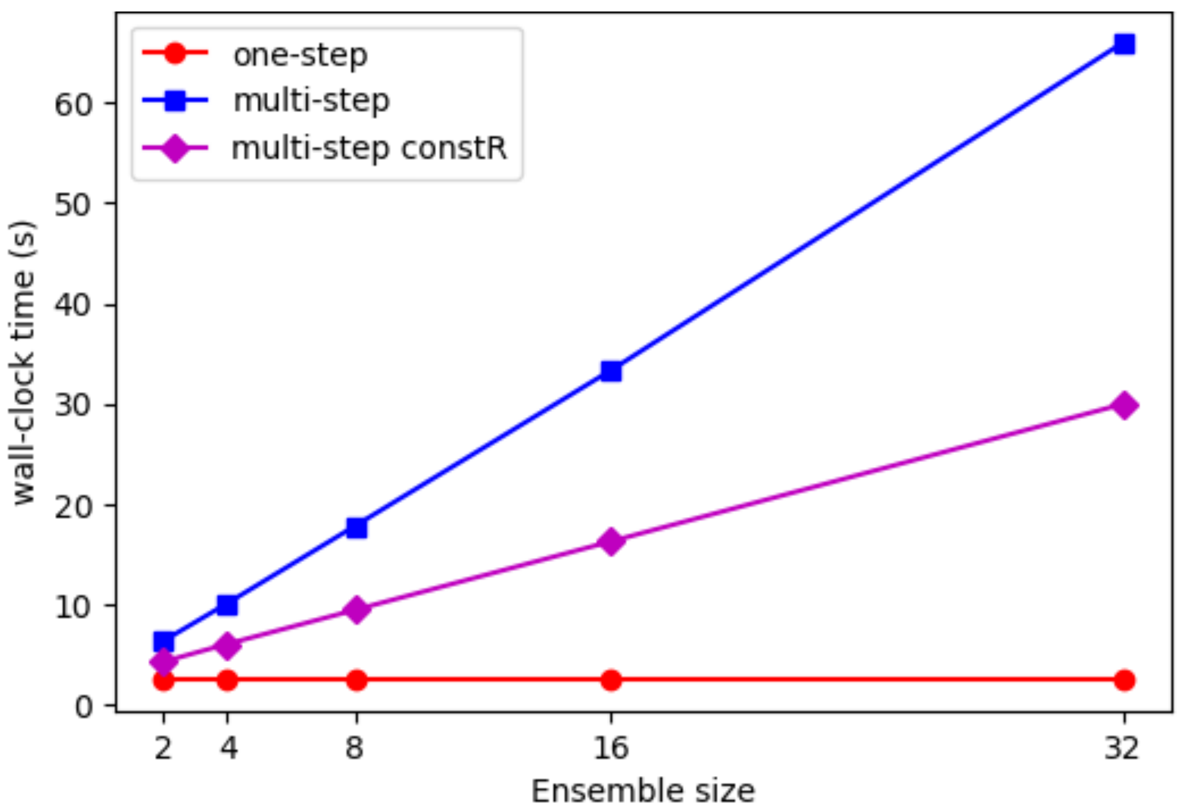
Without measurement error

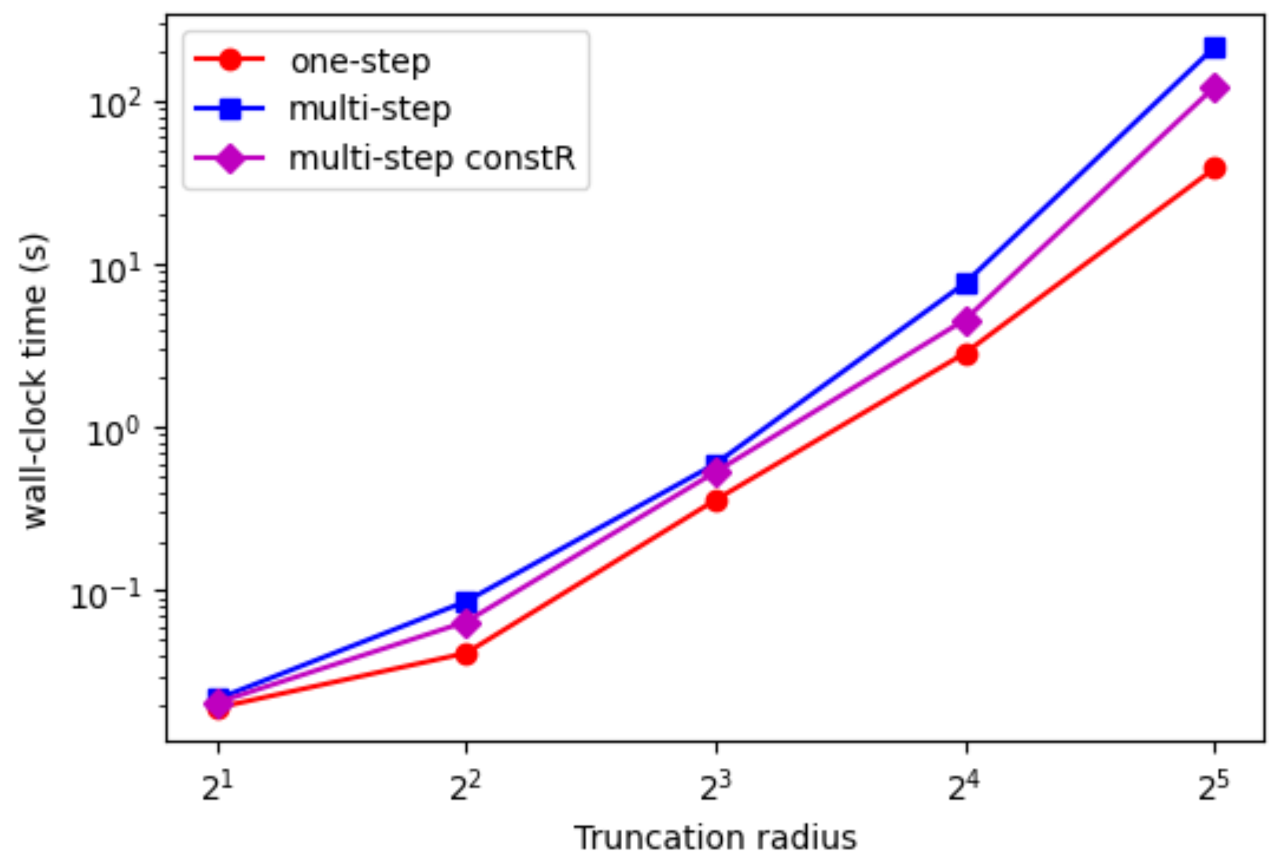




With measurement error

Computational costs





$$O(N_t \times N_s \times |\mathscr{K}_{\mathrm{tr}}|^3)$$

N_t : number of assimilation steps

N_s : ensemble size

$|\mathscr{K}_{\mathrm{tr}}|$: number of truncated modes / grids

Summary

- A **multi-step nonlinear DA** method is developed for multi-layer flow fields. By sampling from the posterior of the filtered layer and used as pseudo-observations for the next step, information can be effectively propagate to adjacent layers.
- For each DA step, **conditional Gaussian nonlinear surrogate models** are systematically used, which allows
  - Nonlinearity of the upper layer to be preserved and characterized in each DA step;
  - Analytic formulae for solving the posterior mean and covariance;
  - The lower-layer posterior is given by a Gaussian mixture, which can be highly non-Gaussian.
- Due to the use of CGNS, the multi-step CGDA is well-suited for assimilating **frequent**, **sparse**, and **highly accurate** observations, a scenario that EnKF can struggle with. However, since reduced-order models are involved, the performance of the multi-step CGDA is also dependent on the quality of these reduced-order models.

Future work

- For discrete-time DA :
  - To assimilate infrequent observations, turn to ML: build a neural network surrogate model, that learns the state transition operator instead of the differential operator, thus allowing large time steps (Conditional Gaussian Koopman Neural Network; CGKN, Chen et al, 2024, 2025).
  - To handle measurement errors: a combined formulation of the discrete- and continuous-time DA.
- A multi-step smoother for parameter estimation.

[A Closed-Form Nonlinear Data Assimilation Algorithm for Multi-Layer Flow Fields, paper available on Arxiv]