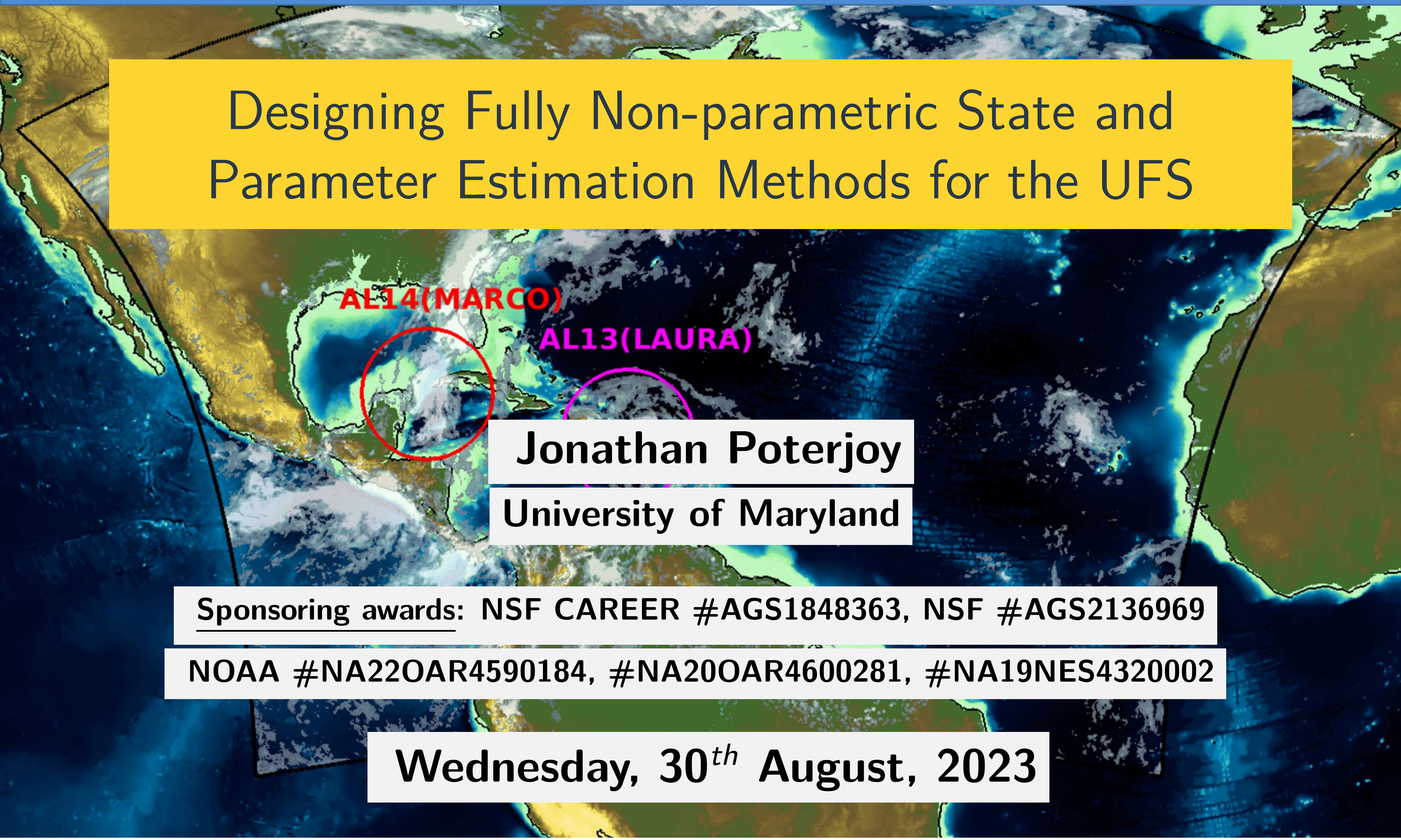
Designing Fully Non-parametric State and Parameter Estimation Methods for the UFS



Jonathan Poterjoy

University of Maryland

Sponsoring awards: NSF CAREER #AGS1848363, NSF #AGS2136969

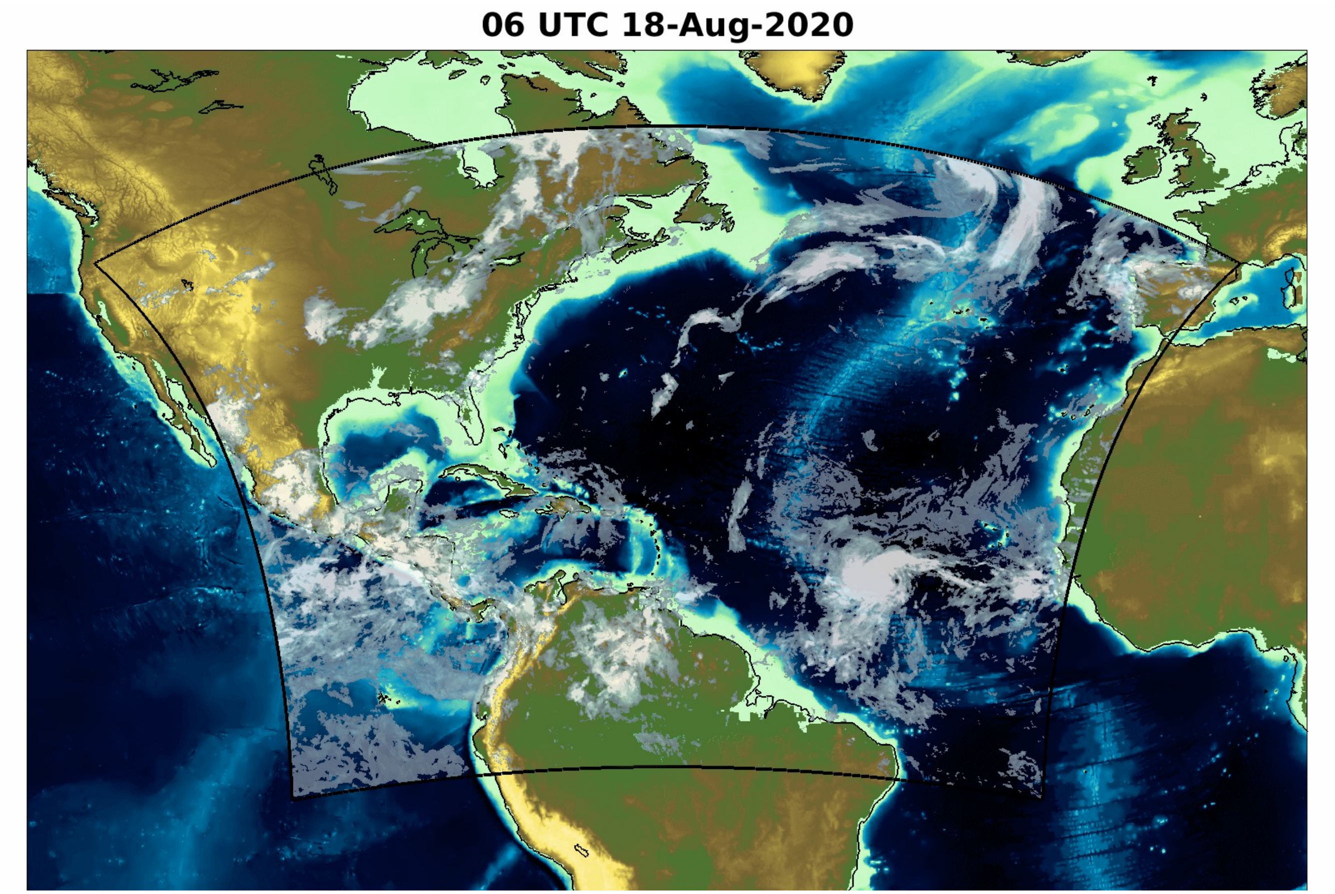
NOAA #NA22OAR4590184, #NA20OAR4600281, #NA19NES4320002

Wednesday, 30th August, 2023

Example application

The **Unified Forecast System (UFS)**: community-based coupled models for "Earth system" prediction at NOAA.

NOAA Hurricane Analysis and Forecast System

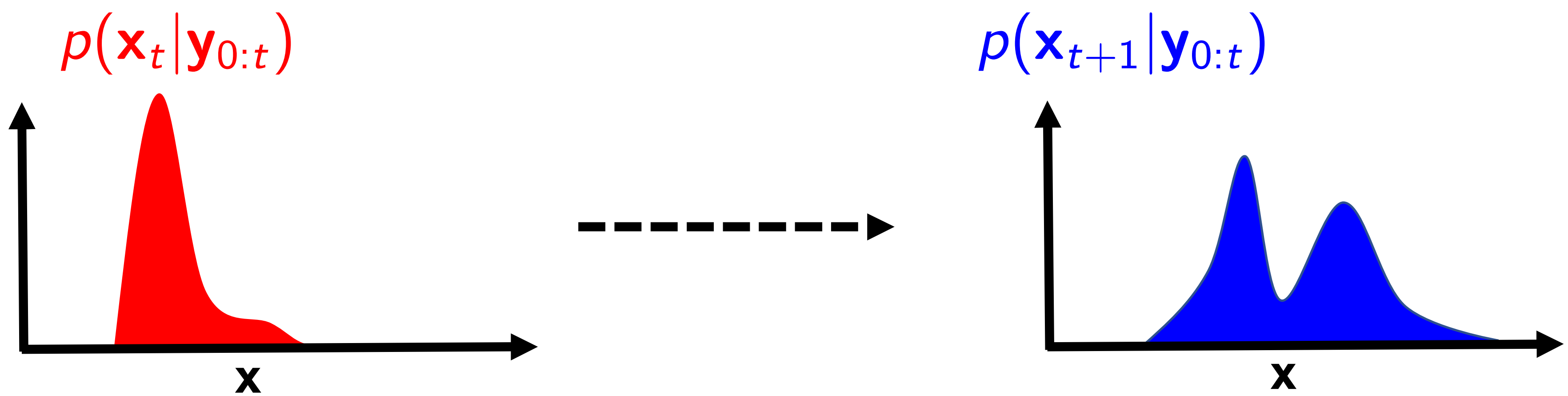


- Operational starting this hurricane season.
- (Left) simulated visible imagery from model "analyses" produced during data assimilation.

Video courtesy of Kenta Kurosawa

Bayesian filtering problem

The goal is to estimate a model state's pdf conditioned on observations.

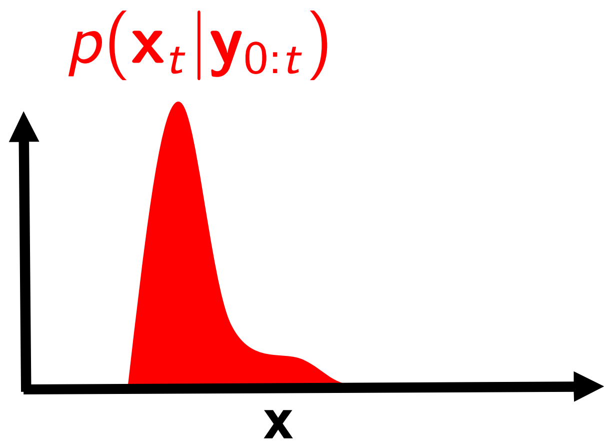


$\mathbf{x}_t$ and $\mathbf{y}_t$ are given by:

$$
\begin{aligned}
\mathbf{x}_{t+1} &= M(\mathbf{x}_t) + \eta_t, \\
\mathbf{y}_t &= H(\mathbf{x}_t) + \epsilon_t.
\end{aligned}
$$

Ensemble data assimilation (DA)

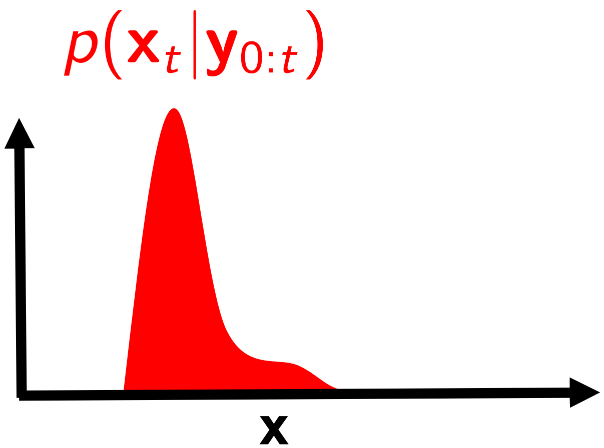
Draw $\mathbf{x}_t^n$ for $n = 1, 2, ..., N_e$, from $p(\mathbf{x}_t|\mathbf{y}_{0:t})$.

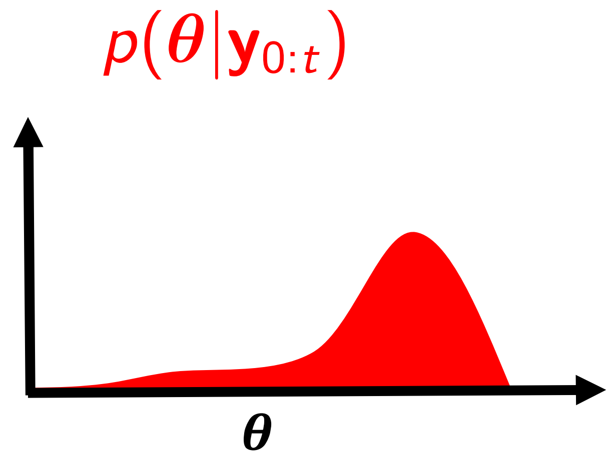


$\mathbf{x}_{t+1}^n = M(\mathbf{x}_t^n) + \eta_t^n$ are samples from forecast density.

Joint state-parameter estimate

Draw $\mathbf{x}_t^n, \boldsymbol{\theta}^n$ for $n = 1, 2, \ldots, N_e$, from $p(\mathbf{x}_t, \boldsymbol{\theta} | \mathbf{y}_{0:t})$.





$\mathbf{x}_{t+1}^n = M(\mathbf{x}_t^n; \boldsymbol{\theta}^n) + \eta_t^n$ are samples from forecast density.

DA with a non-parametric prior

Particle filters (PFs) use ensemble members ("particles") to approximate prior (forecast) distributions for using Bayes' rule.

DA with a non-parametric prior

Particle filters (PFs) use ensemble members ("particles") to approximate prior (forecast) distributions for using Bayes' rule.

For state estimation

$$
\begin{aligned}
p(\mathbf{x}_t|\mathbf{y}_{0:t}) &\propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1}), \\
&\approx p(\mathbf{y}_t|\mathbf{x}_t)\frac{1}{N_e}\sum_{n=1}^{N_e}\delta(\mathbf{x}-\mathbf{x}_t^n),
\end{aligned}
$$

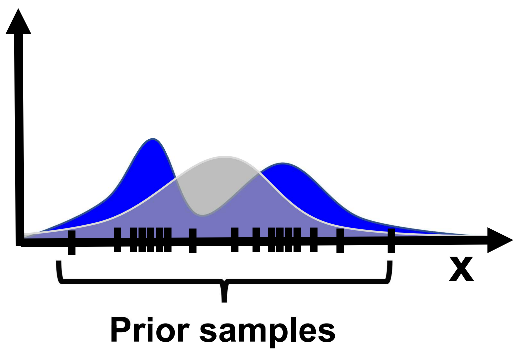
where $\delta(\mathbf{x}-\mathbf{x}_t^n)$ is a Dirac delta function; it returns a zero everywhere except for where $\mathbf{x}=\mathbf{x}_n^t$.

DA with a non-parametric prior

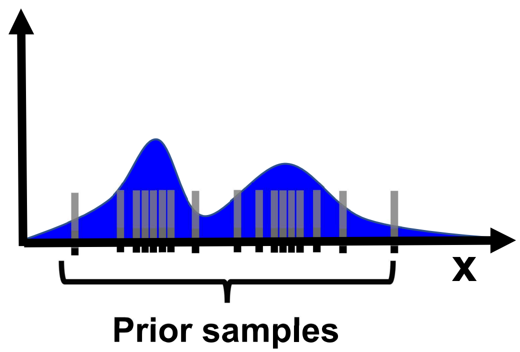
Particle filters (PFs) use ensemble members ("particles") to approximate prior (forecast) distributions for using Bayes' rule.

In the context of DA schemes currently used for NWP:

- (Left) EnKFs use a sample estimate of mean and covariance and fit a Gaussian prior density.
- (Right) PFs use a sum of Dirac delta functions—to form a "non-parametric" prior density.



vs.

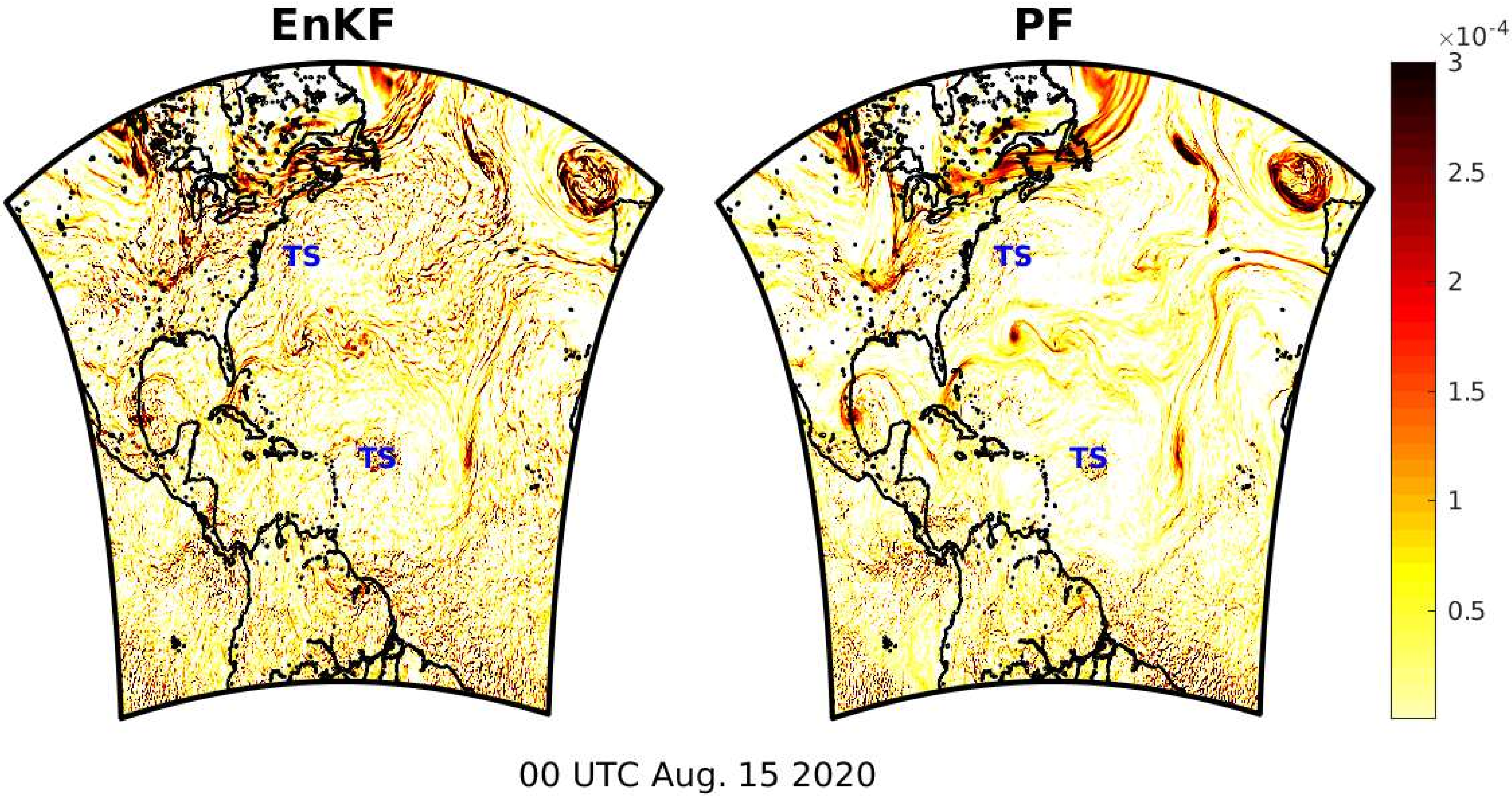


DA with a non-parametric prior

Particle filters (PFs) use ensemble members ("particles") to approximate prior (forecast) distributions for using Bayes' rule.

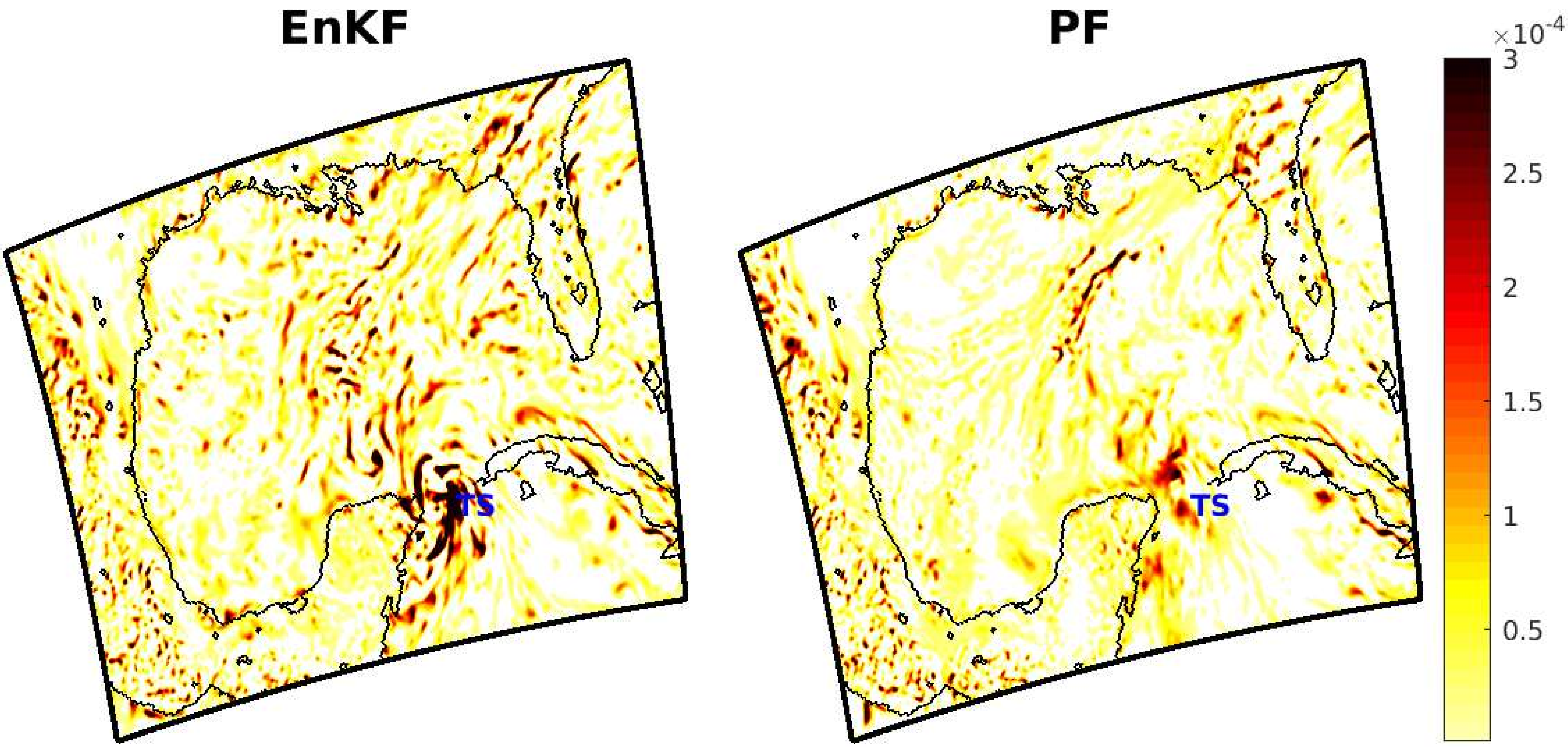
- Even for nonlinear $M(\mathbf{x})$ and $H(\mathbf{x})$, and non-Gaussian errors PFs converge to the Bayesian solution as
  i. ensemble sizes increase.
  ii. model and observation errors are accurately described.
- Like EnKFs, approximations are needed to cope with "curse of dimensionality"—through localization/inflation/regularization (Poterjoy 2016; Poterjoy et al. 2019; Poterjoy 2022ab).

Data assimilation with a “localized” PF



250-mb relative vorticity analyses from the Hurricane Analysis and Forecasting System (HAFS).

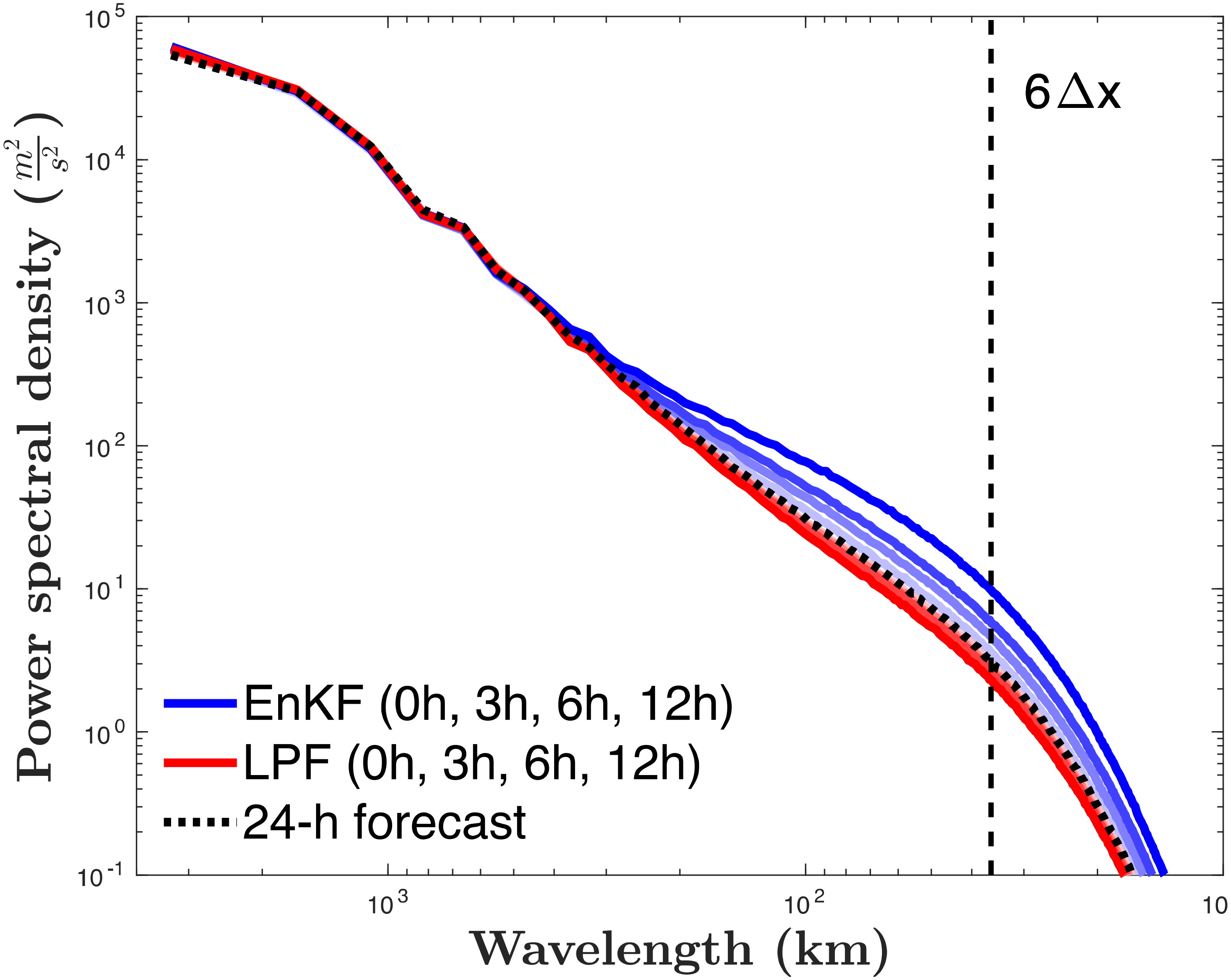
DA representation of tropical cyclones



850-mb ζ during RI of Hurricanes Marco and Laura (2020).

Gaussian DA-induced bias in KE spectrum

Average zonal Kinetic energy spectrum for single members:



NWP application: HAFS

HAFS, and other models, rely on ensemble-variational ("EnVar") data assimilation.

Motivation:

- EnVar is chosen for practical reasons; e.g., use of a high-resolution deterministic "control."
- EnKF typically updates ensemble—short-term forecast from ensemble provides background error covariance for EnVar (in prototype versions of HAFS).
- Posterior EnKF members are re-centered on EnVar analysis.

NWP application: HAFS

DA comparisons:

- "EnKF-Var" $\leftarrow$ HAFS ensemble updated with EnKF and Var
- "PF-Var" $\leftarrow$ HAFS ensemble updated with LPF and Var

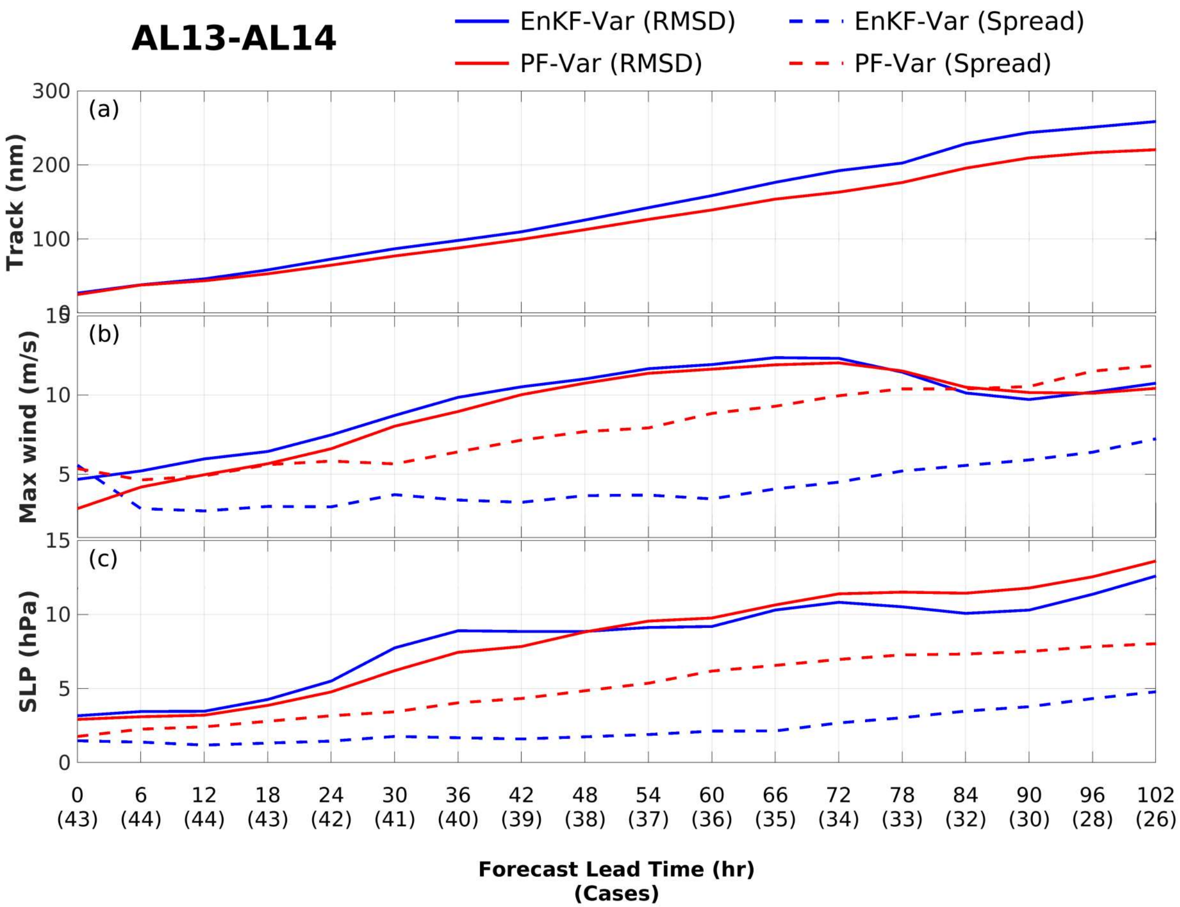
In both experiments, role of EnKF or LPF is to update 40 HAFS ensemble members about a variational analysis.

Verification:

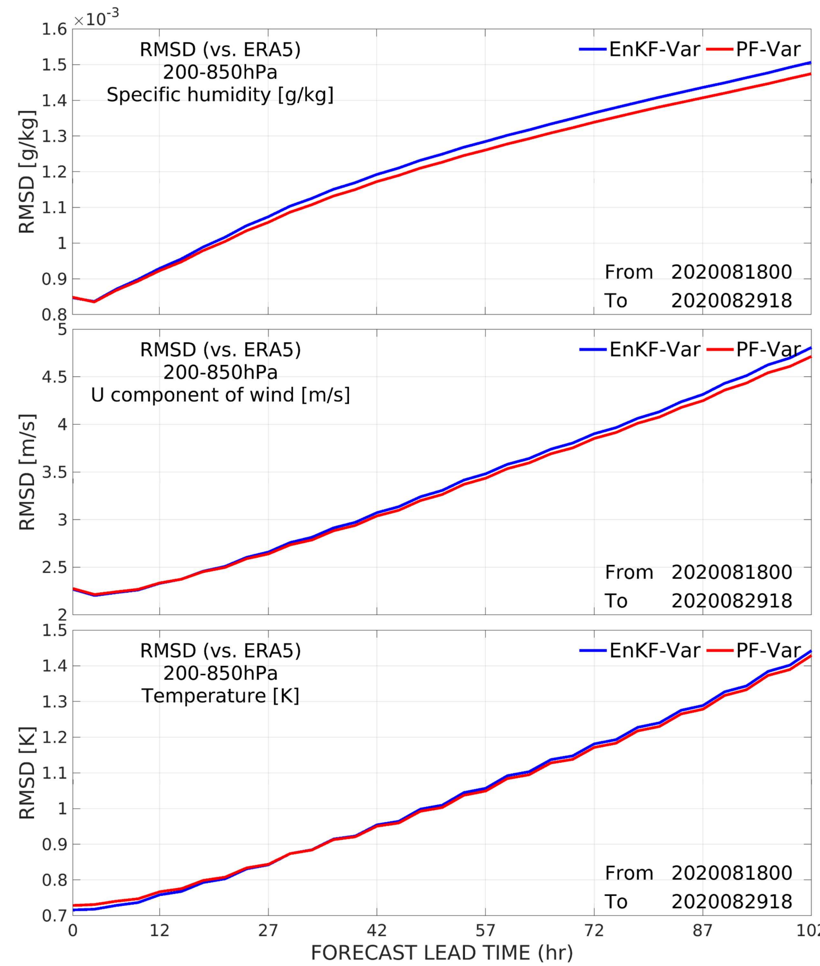
- 10-member forecasts generated every 6 h for 2 weeks
- Storm features verified using NHC Best Track data
- Synoptic scale features verified using ERA5

Verification (2 weeks of forecasts)

Track and intensity RMSEs for Laura and Marco (2020)



Domain-average RMSEs from ERA5



- LPF soon to be applied for hourly-updated GFS (Slivinski et al. 2022) (NOAA/WPO Award: #NA23OAR4590379).

New direction: non-parametric likelihoods

Full potential of LPF still yet to be explored:

$$
\begin{aligned}
p(\mathbf{x}_t|\mathbf{y}_{0:t}) &\propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1}), \\
&\approx p(\mathbf{y}_t|\mathbf{x}_t)\frac{1}{N_e}\sum_{n=1}^{N_e}\delta(\mathbf{x}-\mathbf{x}_t^n),
\end{aligned}
$$

New direction: non-parametric likelihoods

Full potential of LPF still yet to be explored:

$$
\begin{aligned}
p(\mathbf{x}_t|\mathbf{y}_{0:t}) &\propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1}),\\
&\approx p(\mathbf{y}_t|\mathbf{x}_t)\frac{1}{N_e}\sum_{n=1}^{N_e}\delta(\mathbf{x}-\mathbf{x}_t^n),\\
&\propto \sum_{n=1}^{N_e} p(\mathbf{y}_t|\mathbf{x}_t^n)\delta(\mathbf{x}-\mathbf{x}_t^n).
\end{aligned}
$$

Large freedom exists in how we specify $p(\mathbf{y}_t|\mathbf{x}_t^n)$.

Current methodology

Revisiting present choices for $p(\mathbf{y}_t|\mathbf{x}_t^n)$:

Assume $\mathbf{y}_t = H(\mathbf{x}_t^{truth}) + \boldsymbol{\epsilon}_t$, and apply assumptions for distribution of $\boldsymbol{\epsilon}_t$.

Current methodology

Revisiting present choices for $p(\mathbf{y}_t|\mathbf{x}_t^n)$:

Assume $\mathbf{y}_t = H(\mathbf{x}_t^{truth}) + \boldsymbol{\epsilon}_t$, and apply assumptions for distribution of $\boldsymbol{\epsilon}_t$.

For $\boldsymbol{\epsilon}_t^n = \mathbf{y}_t - H(\mathbf{x}_t^n)$,

$$p(\mathbf{y}_t|\mathbf{x}_t^n) \approx \mathcal{N}(\boldsymbol{\epsilon}_t^n; \mathbf{0}, \mathbf{R}_t).$$

Current methodology

Revisiting present choices for $p(\mathbf{y}_t|\mathbf{x}_t^n)$:

Assume $\mathbf{y}_t = H(\mathbf{x}_t^{truth}) + \boldsymbol{\epsilon}_t$, and apply assumptions for distribution of $\boldsymbol{\epsilon}_t$.

For $\boldsymbol{\epsilon}_t^n = \mathbf{y}_t - H(\mathbf{x}_t^n)$,

$$p(\mathbf{y}_t|\mathbf{x}_t^n) \approx \mathcal{N}(\boldsymbol{\epsilon}_t^n; \mathbf{0}, \mathbf{R}_t).$$

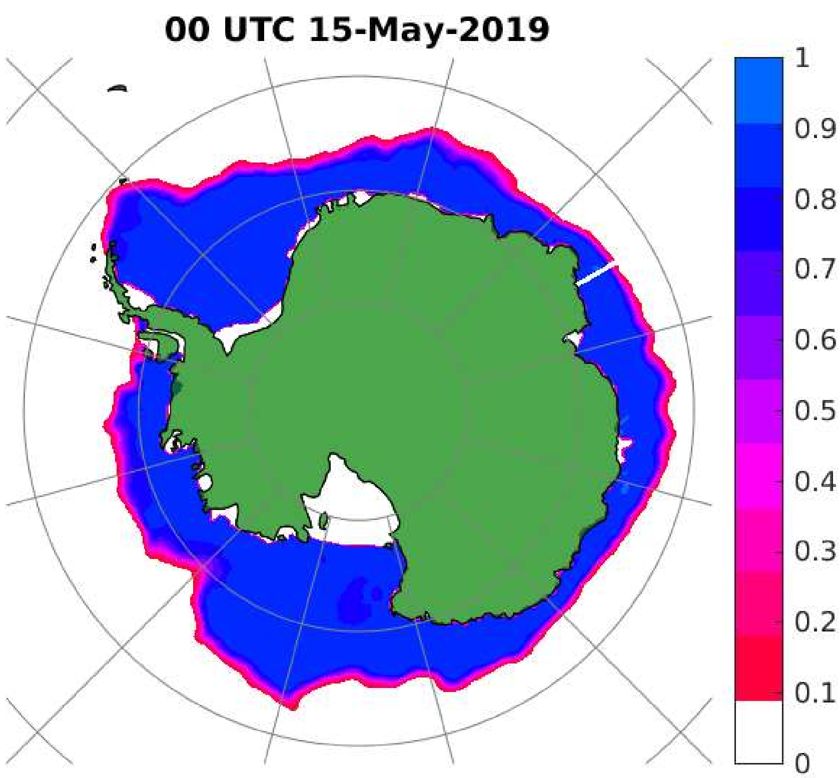
Two approximations:

1. The distribution of ϵ is independent of $\mathbf{x}_t$.
2. The distribution of ϵ is modeled as a Gaussian.

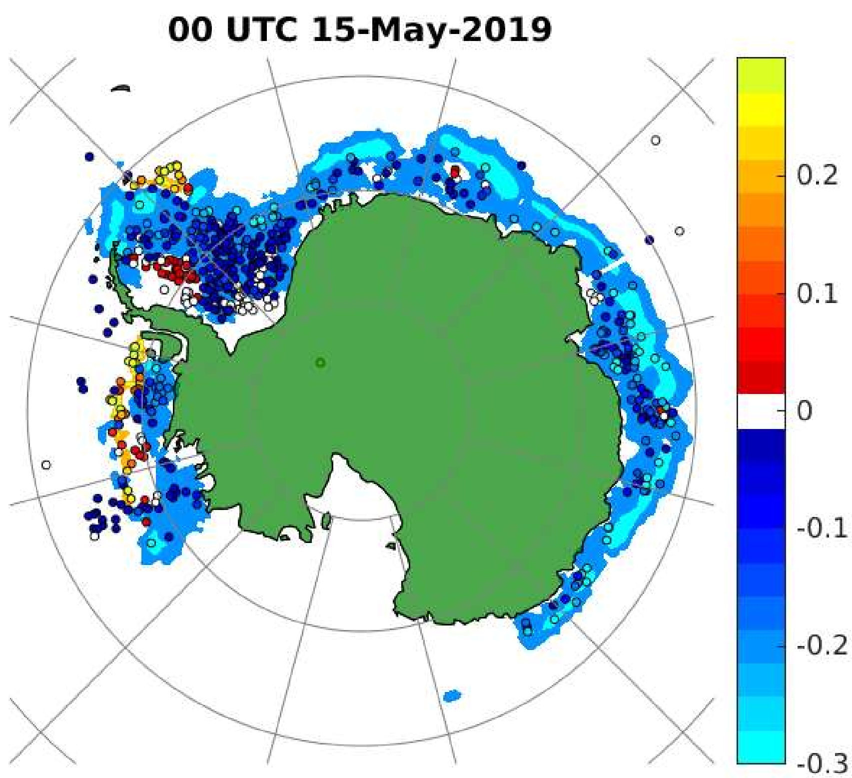
Motivating UFS application

Prototype 6-h coupled ocean/sea ice ensemble DA over Antarctic using MOM6/CICE6:

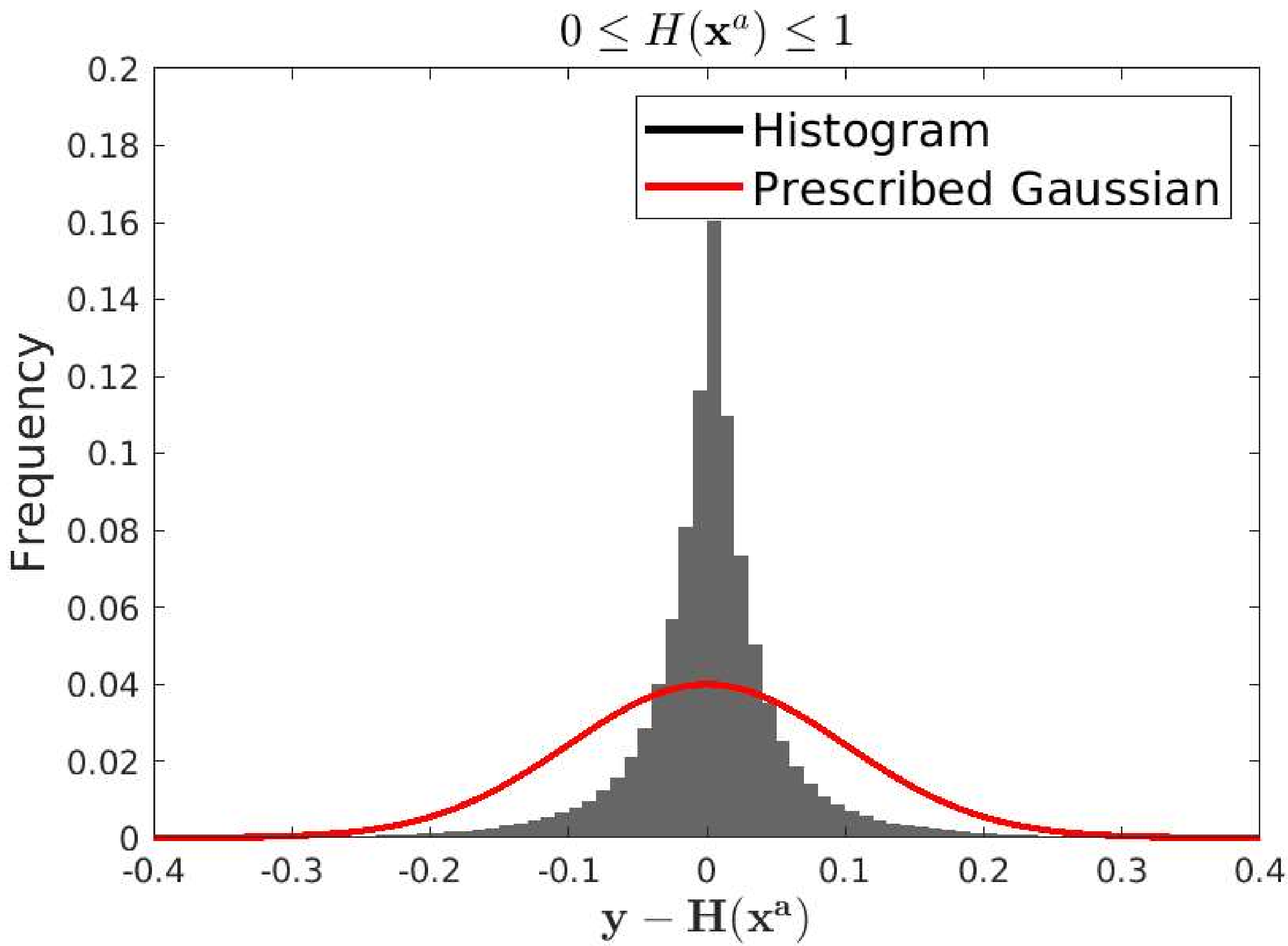
Prior mean



Increments and innovations

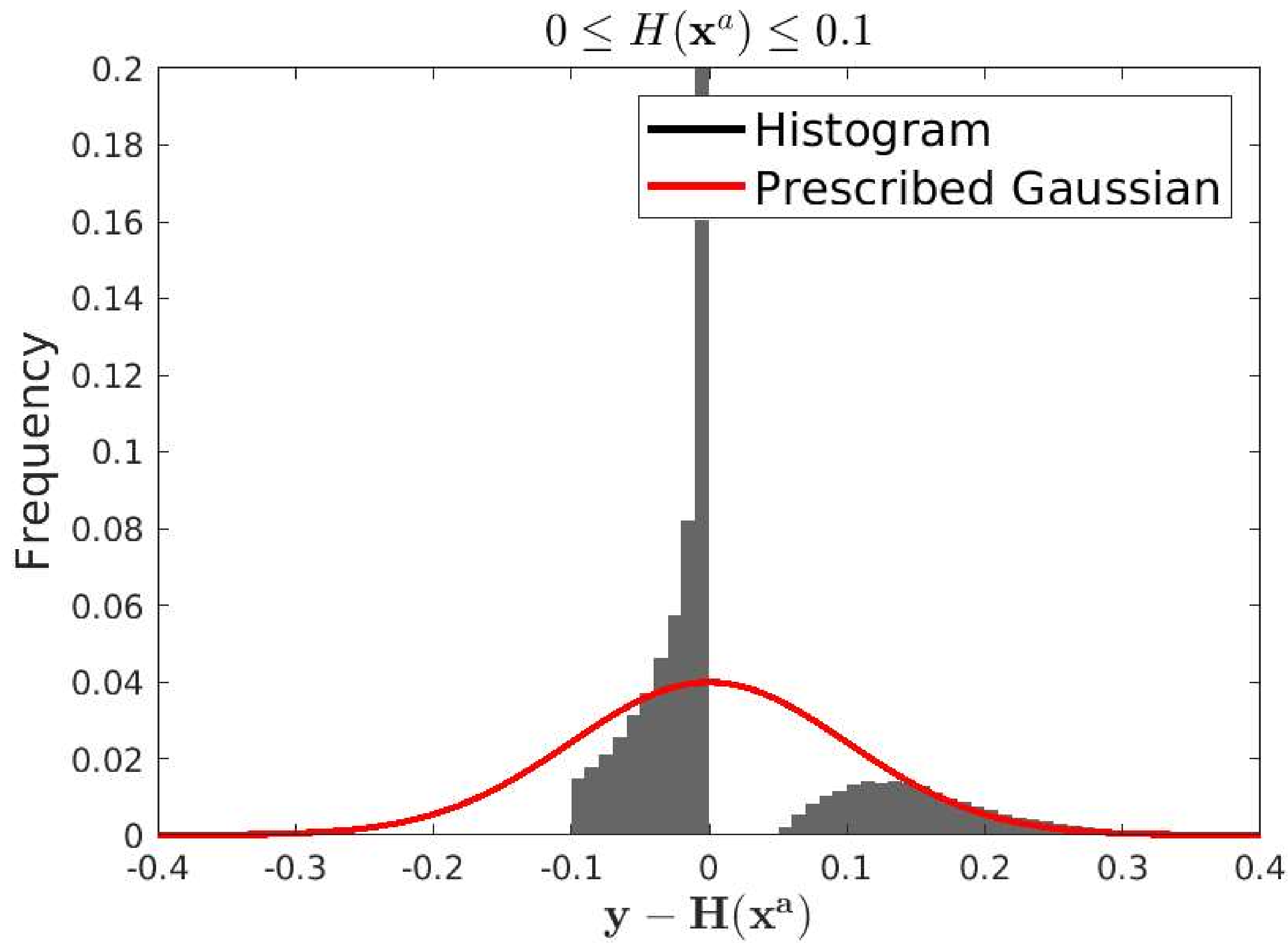


Motivating example



Histogram of $y - H(\mathbf{x}^a)$ for SSMIS sea ice concentration across **all measurements**.

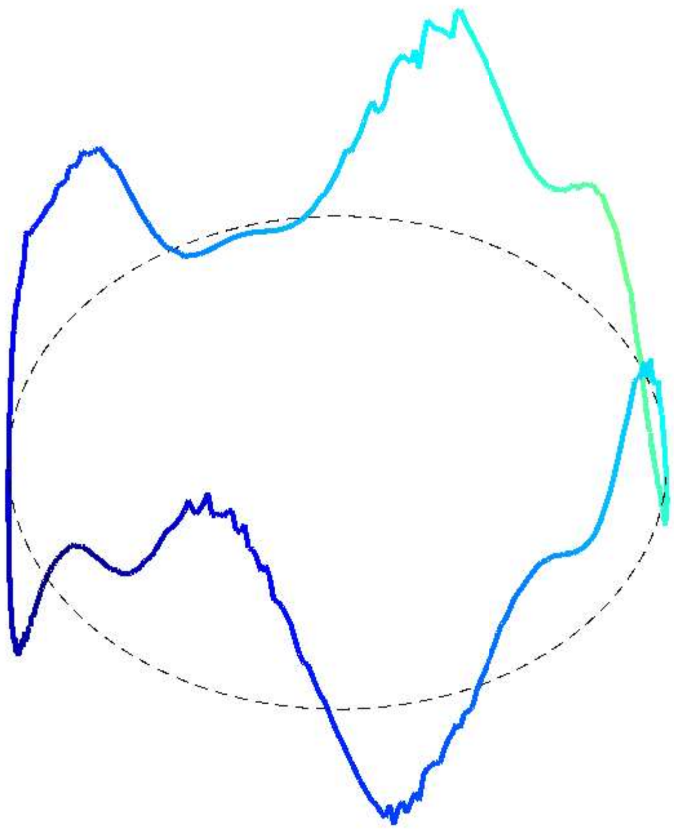
Motivating example



Histograms of $y - H(\mathbf{x}^a)$ for SSMIS sea ice concentration, **stratified by** $H(\mathbf{x}^a)$.

Idealized application

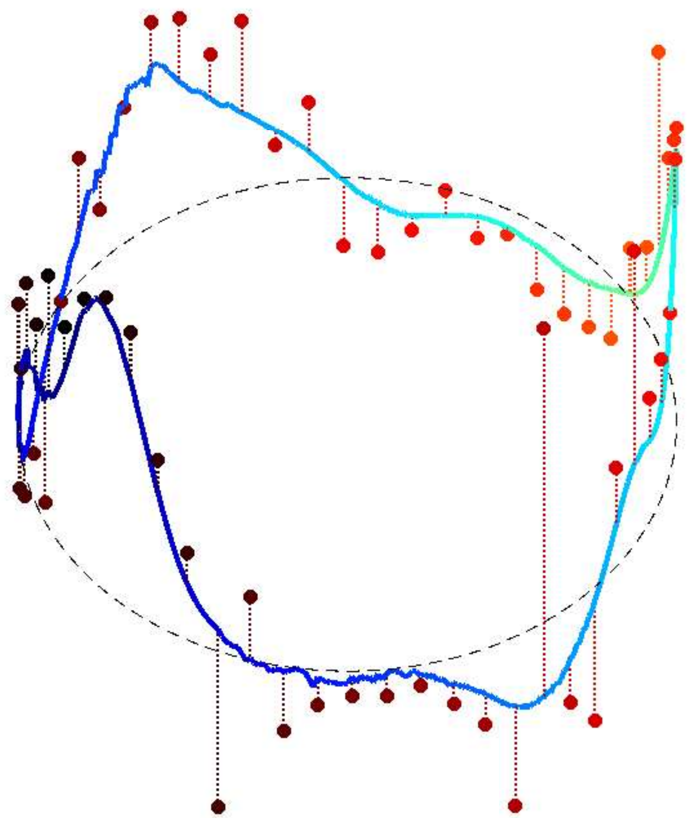
Assimilating obs with non-Gaussian, state-dependent errors



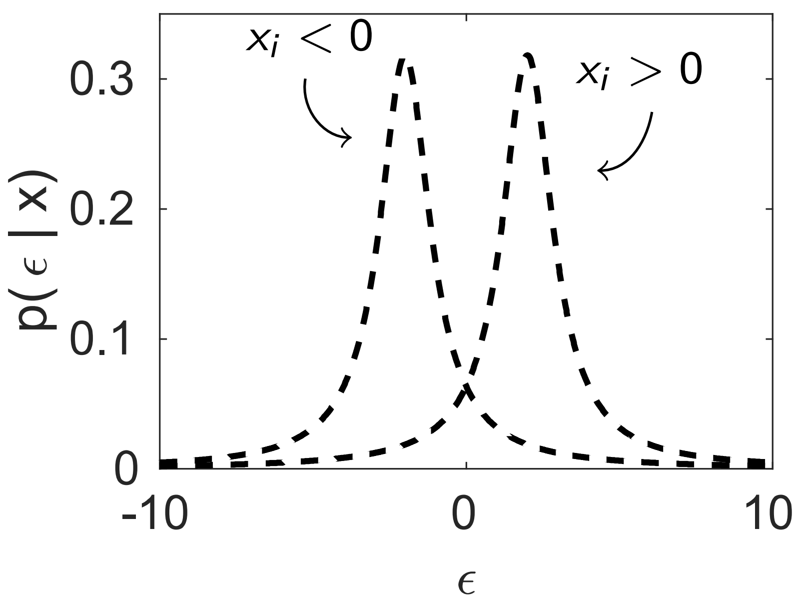
- Model III of Lorenz (2005) on periodic domain
- Model configuration supports chaotic behavior
- Characterized by $N_x = 480$ variables on periodic domain
- Data Assimilation: iterative local particle filter (*Poterjoy 2022, QJRMS; Poterjoy 2022, MWR*)

Idealized application

Assimilating obs with non-Gaussian, state-dependent errors



- Observations: directly measure every 8^{th} variable at $\Delta t = 0.05$
- $y_i = x_i + \epsilon$ for $i = 1, 2, \ldots, N_y$



A data-driven approach

Forming non-parametric estimates for $p(\mathbf{y}_t | \mathbf{x}_t^n)$:

Strategy 1. Form a kernel representation of distributions for ϵ given $\mathbf{x}$.

A data-driven approach

Forming non-parametric estimates for $p(\mathbf{y}_t|\mathbf{x}_t^n)$:

Strategy 1. Form a kernel representation of distributions for $\boldsymbol{\epsilon}$ given $\mathbf{x}$.

Strategy 2. Form a kernel representation of distributions for $\mathbf{y}$ given $\mathbf{x}$.

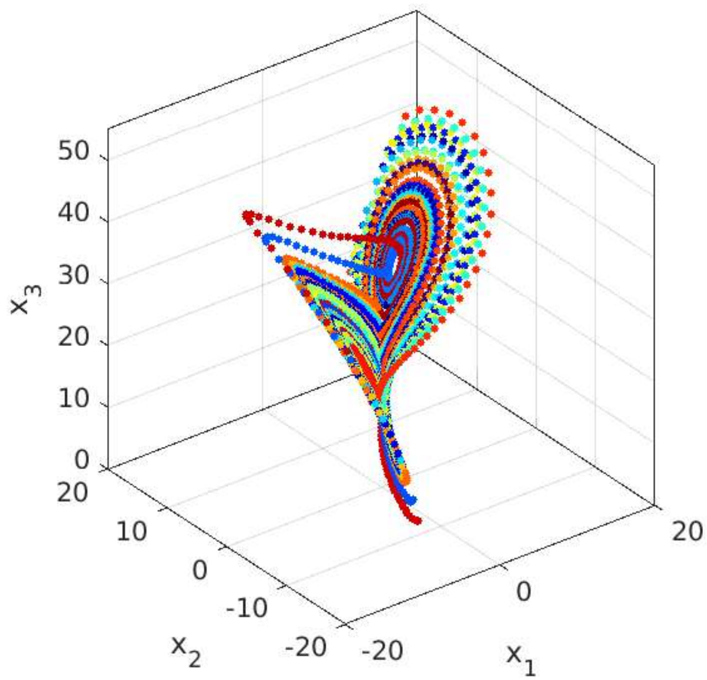
- More general
- Larger training sample needed

A data-driven approach

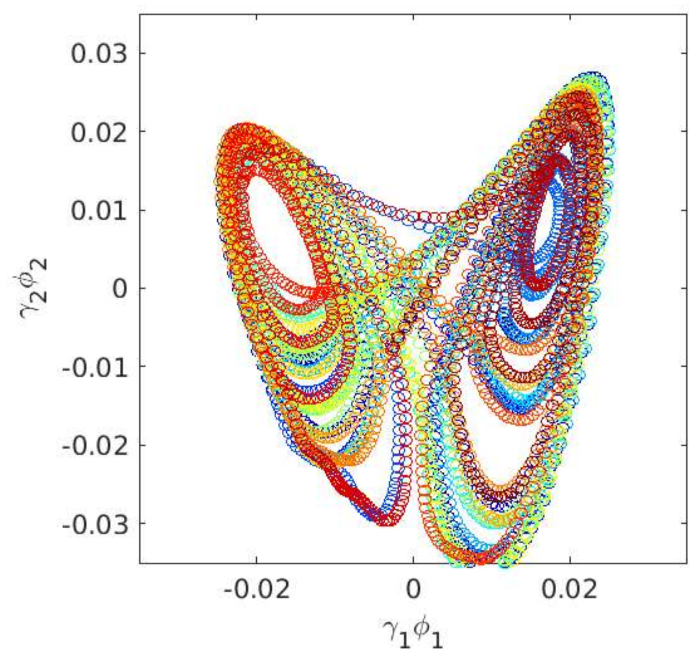
Forming non-parametric estimates for $p(\mathbf{y}_t | \mathbf{x}_t^n)$:

1. Adopt feature space representation of ϵ (or $\mathbf{y}$) and $\mathbf{x}$ from data using nonlinear manifold learning method (*diffusion maps*; Coifman and Lafon 2006; Berry and Harlim 2016).

Model (Lorenz 1963)



2-D representation



A data-driven approach

Forming non-parametric estimates for $p(\mathbf{y}_t|\mathbf{x}_t^n)$:

2. Represent data-driven estimates of $p(\epsilon|\mathbf{x})$ or $p(\mathbf{y}|\mathbf{x})$ using *kernel embeddings of conditional distributions* (Song et al. 2013; Berry and Harlim 2017).

A data-driven approach

Forming non-parametric estimates for $p(\mathbf{y}_t|\mathbf{x}_t^n)$:

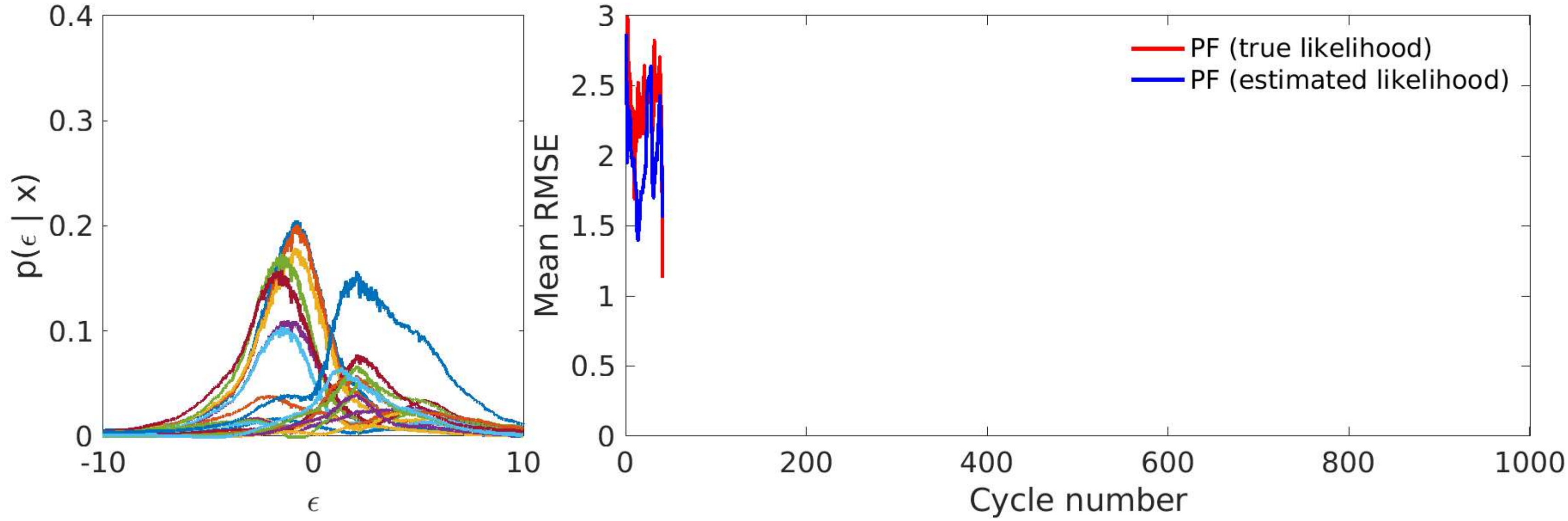
2. Represent data-driven estimates of $p(\epsilon|\mathbf{x})$ or $p(\mathbf{y}|\mathbf{x})$ using *kernel embeddings of conditional distributions* (Song et al. 2013; Berry and Harlim 2017).

Results in an $N \times N$ matrix, $\mathbf{A}$, with elements corresponding to each $p(\epsilon_i|\mathbf{x}_j)$ [or $p(\mathbf{y}_i|\mathbf{x}_j)$] for N pairs of data used during training.

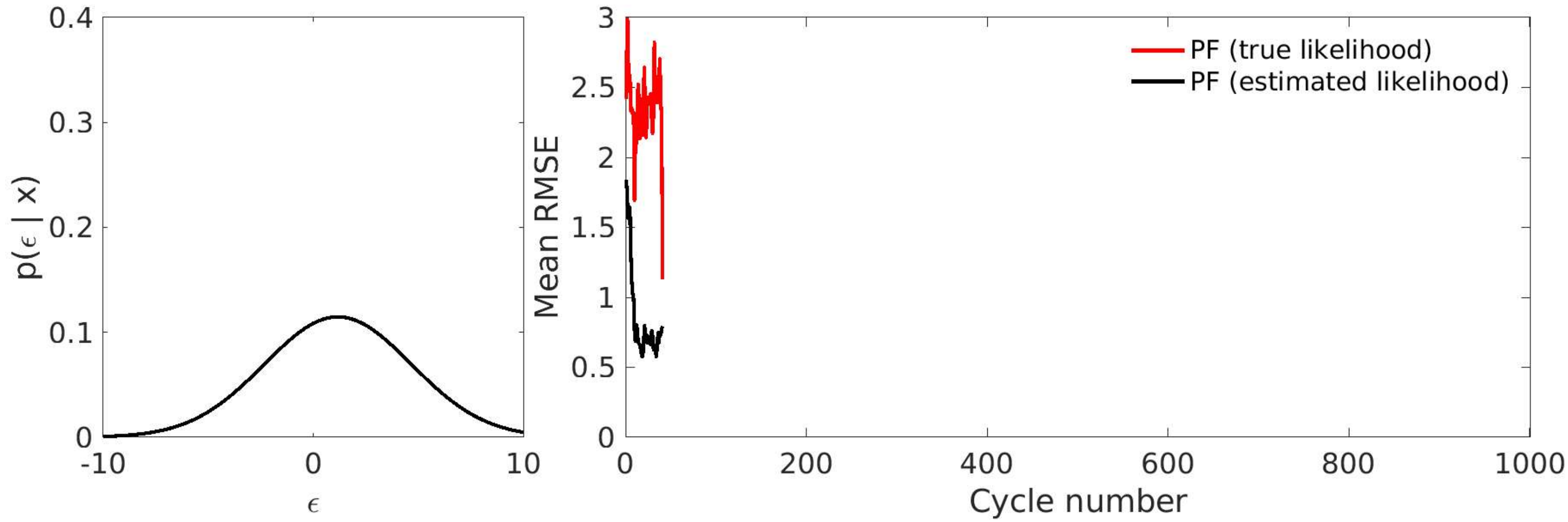
To use with LPF: the likelihood of a given $\mathbf{x}_t^n$ is taken as the $\mathbf{A}_{i,j}$ that is closest to $\mathbf{y}_t$ and $\mathbf{x}_t^n$ (by diffusion distance).

Lorenz example (training time = 40 cycles)

Posterior RMSEs with non-parametric $p(\epsilon_t|\mathbf{x}_t)$



Best Gaussian estimate of $p(\epsilon_t)$ (with QC)



Data-driven likelihoods

Estimating $p(\mathbf{y}_t|\mathbf{x}_t^n)$ instead of $p(\epsilon_t|\mathbf{x}_t^n)$ allows for greater flexibility.

Another application:

- We observe the "square" of model variables without knowing this function; i.e., H only <u>selects</u> state variables near obs.
- The distribution for ϵ_t is still unknown.

Data-driven likelihoods

Estimating $p(\mathbf{y}_t|\mathbf{x}_t^n)$ instead of $p(\epsilon_t|\mathbf{x}_t^n)$ allows for greater flexibility.

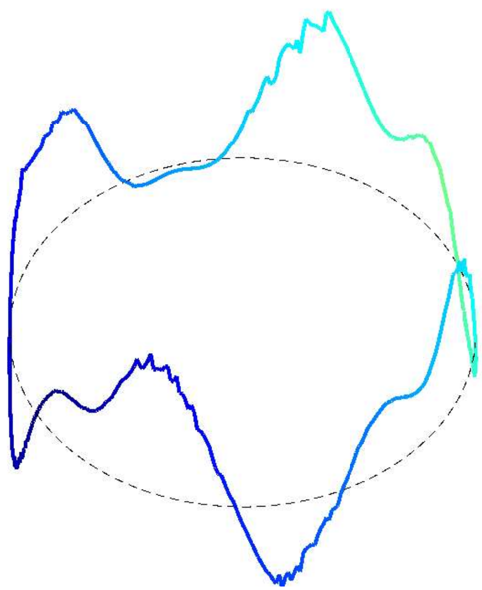
Another application:

- 5 unknown parameters: $\boldsymbol{\theta} = [K, b, I, c, F]^\top$, control frequency, amplitude, coupling, forcing for waves:

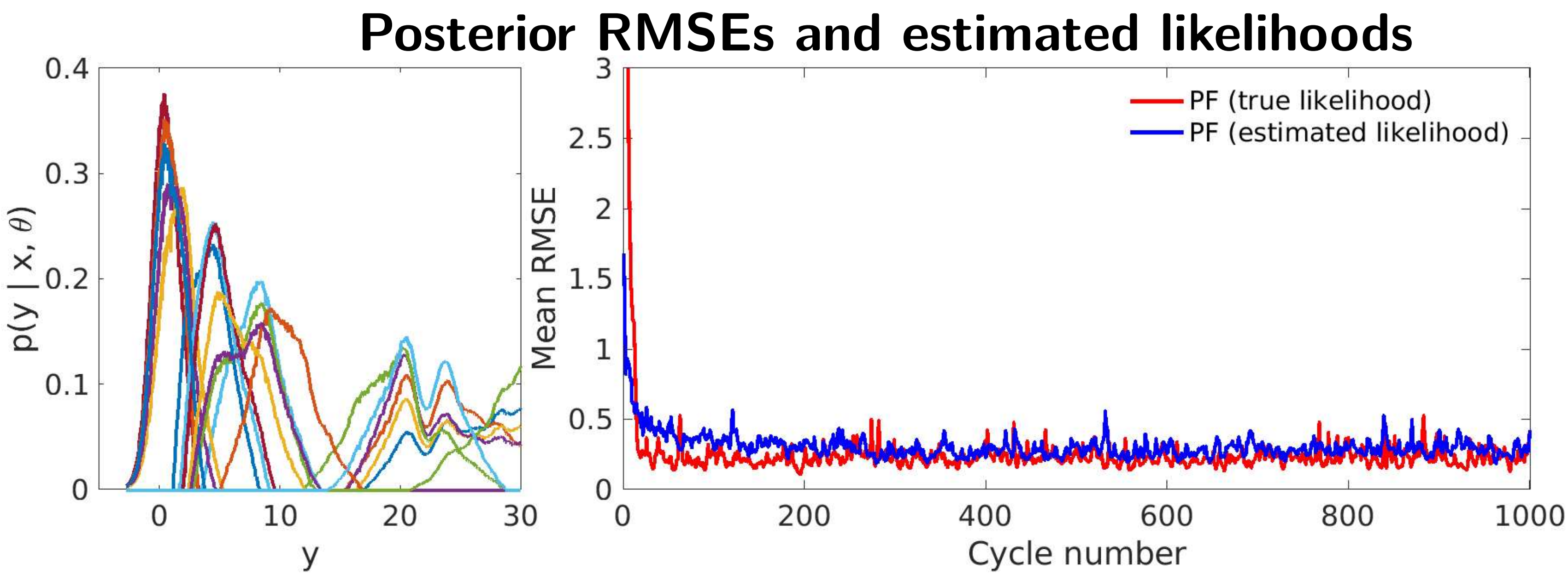
$$\frac{dZ_j}{dt} = [X, X]_{K,j} + b^2[Y, Y]_{1,j} + c[Y, X]_{1,j} - X_j - bY_j + F,$$

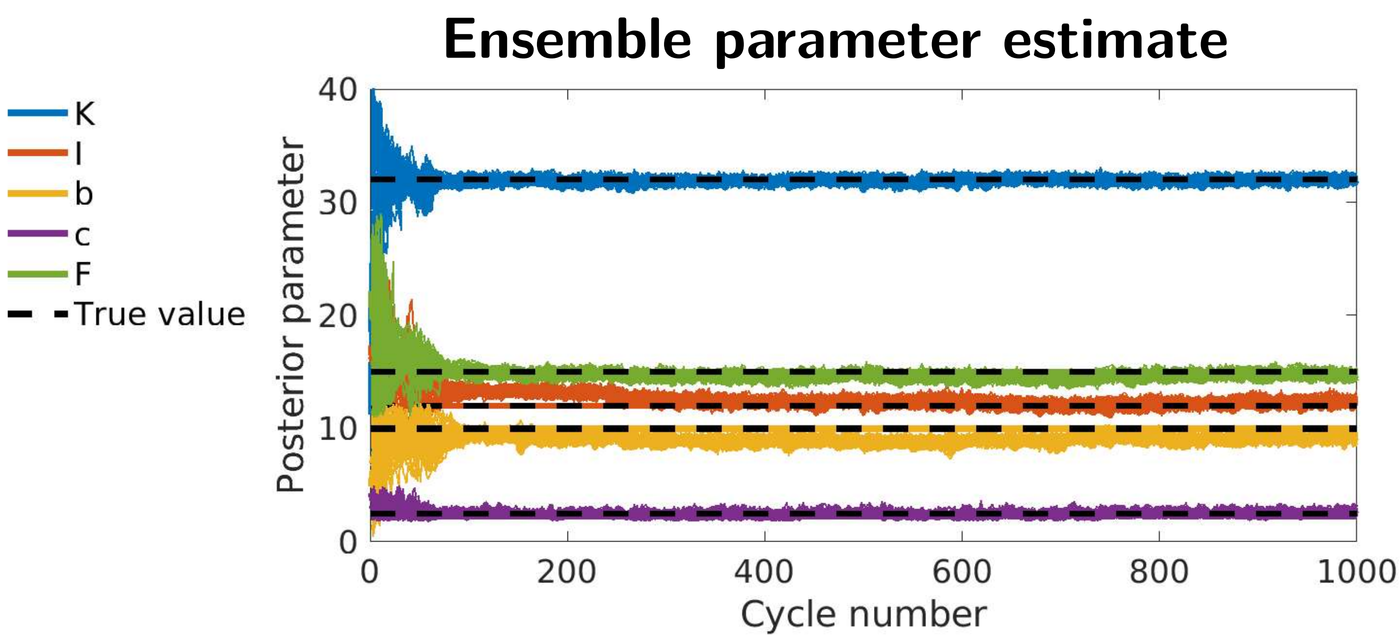
$$X_j = \sum_{i=-I}^{i=I} f(I, i) Z_{j+i},$$

$$Y_j = Z_j - X_j.$$



Joint state-parameter estimation





Advantages for parameter estimation

Reminder: goal is to sample from $p(\mathbf{x}_t, \boldsymbol{\theta}|\mathbf{y}_{0:t})$ where

$$p(\mathbf{x}_t, \boldsymbol{\theta}|\mathbf{y}_{0:t}) \quad \propto \quad p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta})p(\mathbf{x}_t, \boldsymbol{\theta}).$$

Advantages for parameter estimation

Reminder: goal is to sample from $p(\mathbf{x}_t, \boldsymbol{\theta}|\mathbf{y}_{0:t})$ where

$$p(\mathbf{x}_t, \boldsymbol{\theta}|\mathbf{y}_{0:t}) \;\;\propto\;\; p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta})p(\mathbf{x}_t, \boldsymbol{\theta}).$$

Can (in principle) be done in two steps:

1. Perform state update; i.e., sample from $p(\mathbf{x}_t|\mathbf{y}_t)$.
2. Perform parameter update; i.e., sample from $p(\boldsymbol{\theta}|\mathbf{x}_t, \mathbf{y}_t)$.

Advantages for parameter estimation

Reminder: goal is to sample from $p(\mathbf{x}_t, \boldsymbol{\theta}|\mathbf{y}_{0:t})$ where

$$p(\mathbf{x}_t, \boldsymbol{\theta}|\mathbf{y}_{0:t}) \quad \propto \quad p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta})p(\mathbf{x}_t, \boldsymbol{\theta}).$$

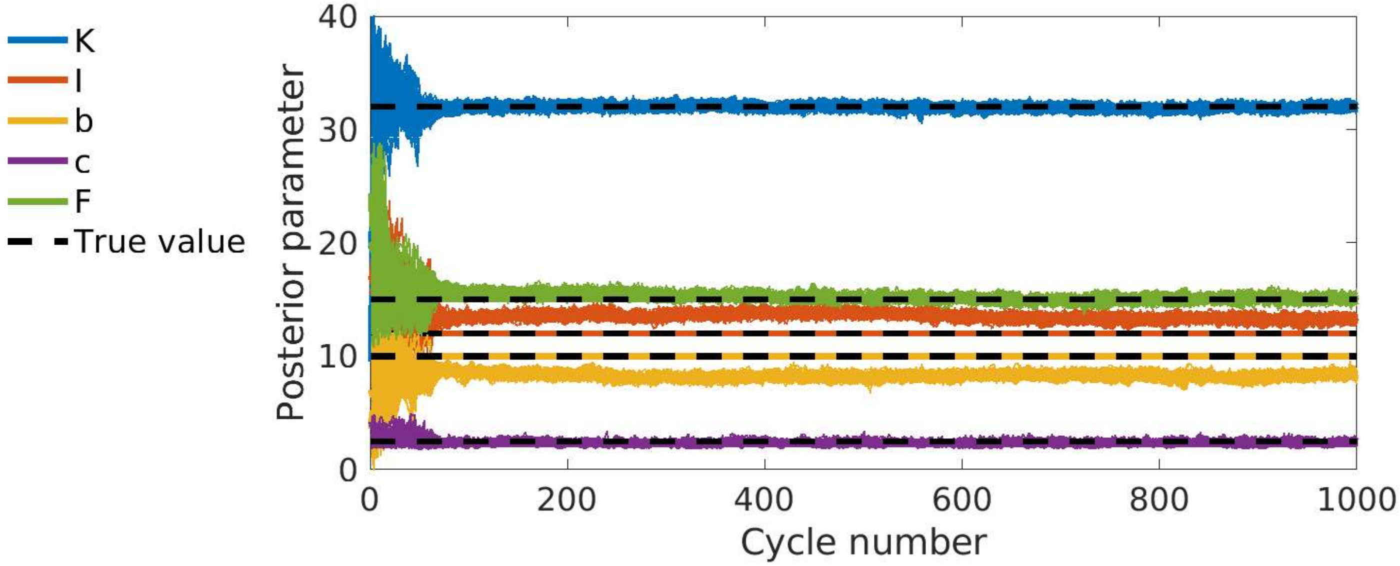
Can (in principle) be done in two steps:

1. Perform state update; i.e., sample from $p(\mathbf{x}_t|\mathbf{y}_t)$.
2. Perform parameter update; i.e., sample from $p(\boldsymbol{\theta}|\mathbf{x}_t, \mathbf{y}_t)$.

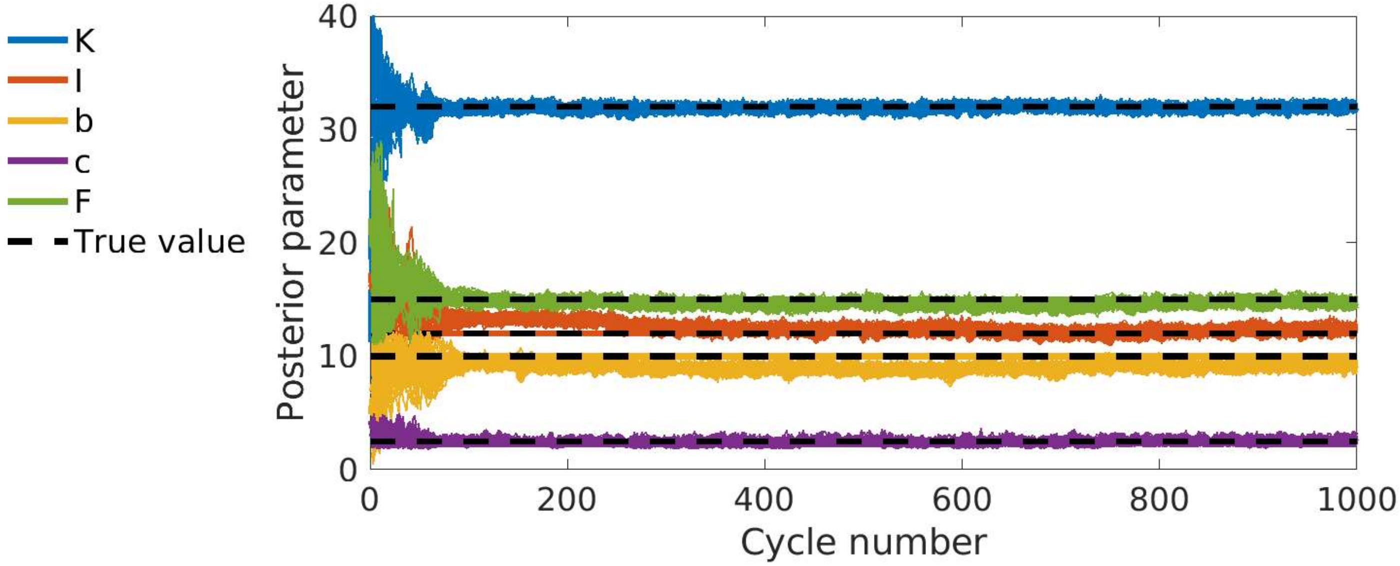
Estimating likelihoods trained on subsets of variables in $\mathbf{x}_t$ is advantageous, even if observations are direct point estimates of scalar state variables in $\mathbf{x}_t$.

Advantages for parameter estimation

Likelihoods only consider point at observation location



Likelihoods consider 11 nearest variables to observation location



Summary

Non-Gaussian DA is now feasible for high-dimensional applications, such as weather prediction.

Early results using the Poterjoy (2022) "iterative" local PF are encouraging, but the full benefits still need to be explored.

Maturity of non-Gaussian DA encourages the use of new research focusing on likelihoods used for state and parameter estimation.

References

Berry, T. and J. Harlim, 2017: Correcting biased observation model error in data assimilation. Mon. Wea. Rev. 145, 2833 – 2853.

Coifman, R., and S. Lafon, 2006: Diffusion maps .Appl. Comput. Harmonic Anal., 21, 5 – 30.

Kurosawa, K., and Poterjoy, J., 2021: Data assimilation challenges posed by nonlinear operators: A comparative study of ensemble and variational filters and smoothers, Mon. Wea. Rev. 149, 2369 – 2389.

Kurosawa, K. and J. Poterjoy, 2023: A statistical hypothesis testing strategy for adaptively blending particle filters and ensemble Kalman filters for data assimilation. Mon. Wea. Rev., 151, 105 – 125.

McCurry, J., J. Poterjoy, K. Knopfmeier, and L. Wicker, 2023: An Evaluation of Non-Gaussian Data Assimilation Methods in Moist Convective Regimes. Mon. Wea. Rev., 151, Provisionally accepted.

Poterjoy, J. 2022: Implications of multivariate non-Gaussian data assimilation for multi-scale weather prediction. Mon. Wea. Rev. 150, 1475 – 1493.

Poterjoy, J., 2022: Regularization and tempering for a moment-matching localized particle filter. Q. J. Roy. Meteor. Soc., Published online 31 May 2022.

Song, K. Fukumizu, and A. Gretton, 2013: Kernel embeddings of conditional distributions: A unified kernel framework for non-parametric inference in graphical models. IEEE Signal Process. Mag., 30, 98 – 111.

Data-driven likelihoods

Constructing $\mathbf{A}$:

Data-driven likelihoods

Constructing $\mathbf{A}$:

i. Collect training data

- $\{\mathbf{z}_k\}_{k=1}^{N}$, where $\mathbf{z}_k = H_k(\tilde{\mathbf{x}}_m)$ is a randomly drawn obs-space posterior member (index m) valid at time of observation

Data-driven likelihoods

Constructing $\mathbf{A}$:

i. Collect training data

- $\{\mathbf{z}_k\}_{k=1}^{N}$, where $\mathbf{z}_k = H_k(\tilde{\mathbf{x}}_m)$ is a randomly drawn obs-space posterior member (index m) valid at time of observation

Aside: Each $\mathbf{z}_k$ serves as a proxy for the portion of true state that impacts measurements. $H()$ *does not need to be a traditional measurement operator.*

Data-driven likelihoods

Constructing $\mathbf{A}$:

i. Collect training data

- $\{\mathbf{z}_k\}_{k=1}^N$, where $\mathbf{z}_k = H_k(\tilde{\mathbf{x}}_m)$ is a randomly drawn obs-space posterior member (index m) valid at time of observation
- $\{\mathbf{d}_k\}_{k=1}^N$, where $\mathbf{d}_k = \mathbf{y}_k - \mathbf{z}_k$

Data-driven likelihoods

Constructing $\mathbf{A}$:

i. Collect training data

- $\{\mathbf{z}_k\}_{k=1}^{N}$, where $\mathbf{z}_k = H_k(\tilde{\mathbf{x}}_m)$ is a randomly drawn obs-space posterior member (index m) valid at time of observation
- $\{\mathbf{d}_k\}_{k=1}^{N}$, where $\mathbf{d}_k = \mathbf{y}_k - \mathbf{z}_k$

Aside: Each $\mathbf{d}_k$ serves as a proxy for ϵ. *Replace* $\{\mathbf{d}_k\}_{k=1}^{N}$ *with* $\{\mathbf{y}_k\}_{k=1}^{N}$ *if estimating* $p(\mathbf{y}_i|\mathbf{x}_j)$.

Data-driven likelihoods

Constructing $\mathbf{A}$:

i. Collect training data
   - $\{\mathbf{z}_k\}_{k=1}^{N}$, where $\mathbf{z}_k = H_k(\tilde{\mathbf{x}}_m)$ is a randomly drawn obs-space posterior member (index m) valid at time of observation
   - $\{\mathbf{d}_k\}_{k=1}^{N}$, where $\mathbf{d}_k = \mathbf{y}_k - \mathbf{z}_k$

ii. Form kernel estimate of each $p(\mathbf{d}_k)$ or $p(\mathbf{y}_k)$ from data.

Data-driven likelihoods

Constructing $\mathbf{A}$:

i. Collect training data
   - $\{\mathbf{z}_k\}_{k=1}^N$, where $\mathbf{z}_k = H_k(\tilde{\mathbf{x}}_m)$ is a randomly drawn obs-space posterior member (index m) valid at time of observation
   - $\{\mathbf{d}_k\}_{k=1}^N$, where $\mathbf{d}_k = \mathbf{y}_k - \mathbf{z}_k$

ii. Form kernel estimate of each $p(\mathbf{d}_k)$ or $p(\mathbf{y}_k)$ from data.

iii. $\mathbf{A}_{i,j}$ then represented using kernel embeddings of conditional distributions—with basis from diffusion maps in place of Gaussian RBF (Berry and Harlim 2017).

Kernel embeddings of conditional distributions

We can represents likelihoods using kernel embeddings:

$$\boxed{p(\mathbf{d}_i|\mathbf{z}_j) = \sum_{k=1}^{M} \mu_{kj}\phi_k(\mathbf{d}_i)q(\mathbf{d}_i)}$$

See Song et al. (2009,2013)

$$\begin{aligned}
\mu_{kj} &= \sum_{l=1}^{M} \psi_l(\mathbf{z})[\mathbf{C}\tilde{\mathbf{C}}^{-1}]_{kl}, \\
\mathbf{C}_{lk} &= \frac{1}{N}\sum_{j=1}^{N} \phi_l(\mathbf{d}_j)\psi_k(\mathbf{z}_j), \\
\tilde{\mathbf{C}}_{lk} &= \frac{1}{N}\sum_{j=1}^{N} \psi_l(\mathbf{z}_j)\psi_k(\mathbf{z}_j).
\end{aligned}$$

where μ_{kj} coefficients determine dependence across **d** and **z**.

Constructing marginals and basis

For $q(\mathbf{d})$, adopt a kernel estimate:

- Variable bandwidth kernel densities provide non-parametric representation of marginal pdfs.

$$q(\mathbf{d}) = \sum_{k=1}^{N} N(\mathbf{d}_k, \mathbf{B}_k), \text{ where } \mathbf{B}_k \text{ is a covariance.}$$

Constructing marginals and basis

For $q(\mathbf{d})$, adopt a kernel estimate:

- Variable bandwidth kernel densities provide non-parametric representation of marginal pdfs.

$$q(\mathbf{d}) = \sum_{k=1}^{N} N(\mathbf{d}_k, \mathbf{B}_k), \text{ where } \mathbf{B}_k \text{ is a covariance.}$$

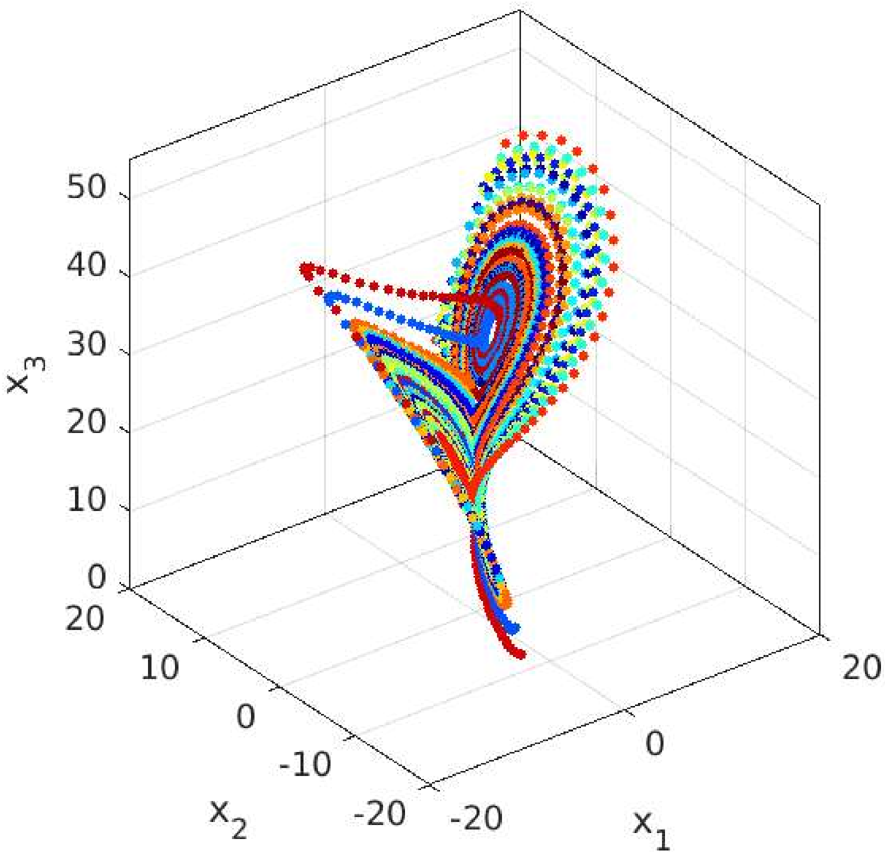
For basis functions, diffusion maps (Coifman and Lafon 2006) is a reasonable choice:

- Manifold learning method for represent data in lower-dimensional space
- Similar strategy applied by Berry and Harlim (2017)

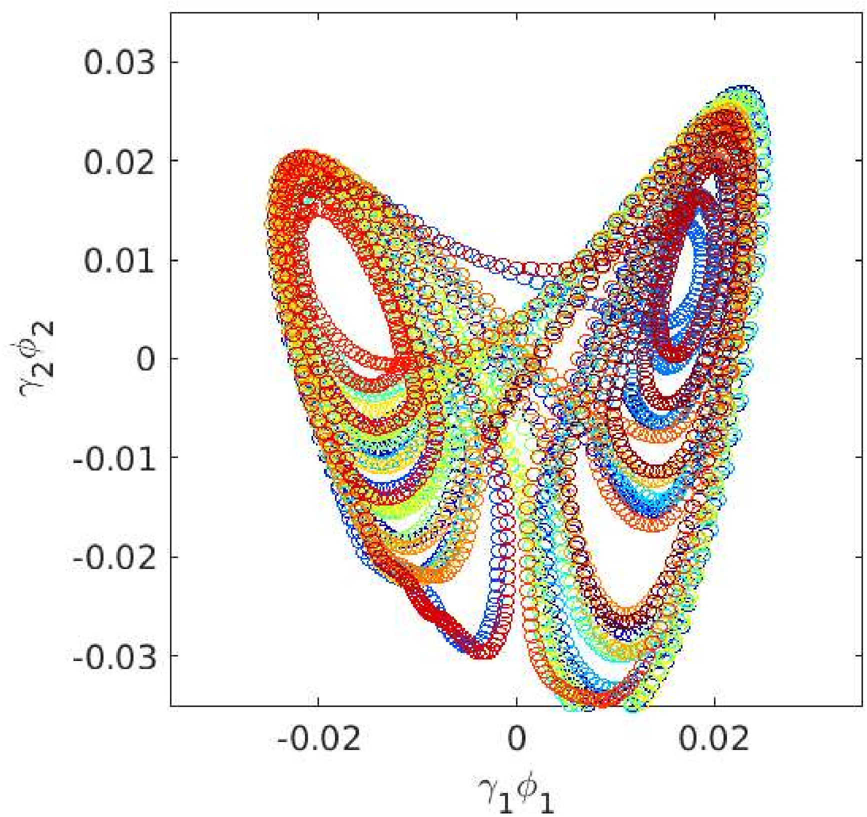
Constructing basis functions

Example: Data produced from Lorenz (1963) model

Model data



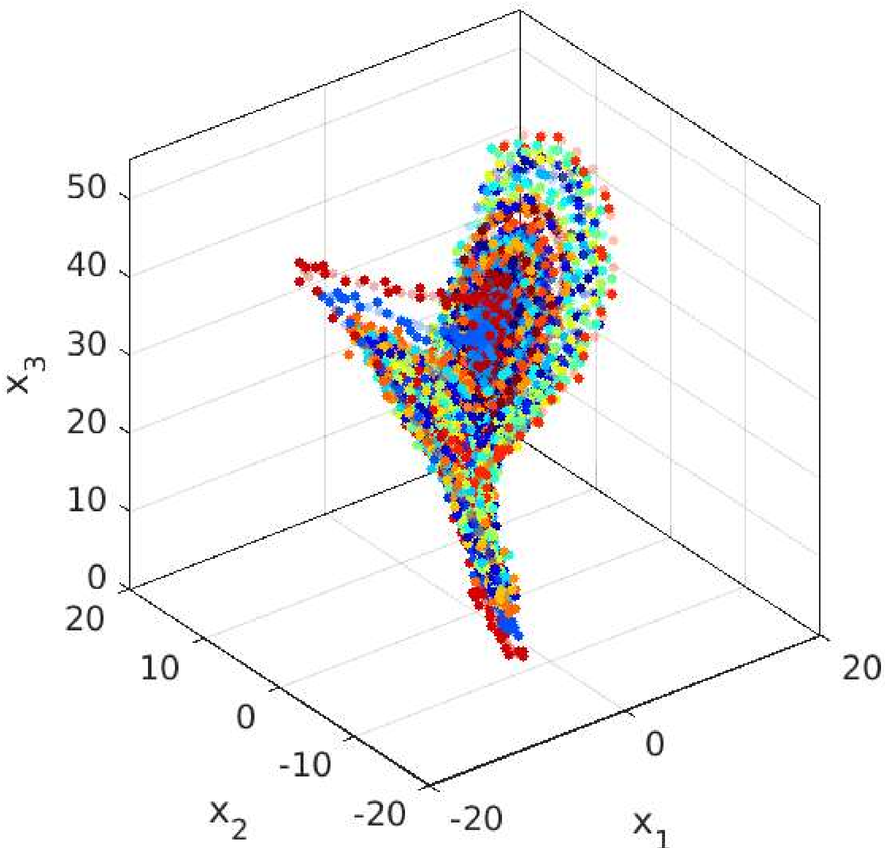
Two-dimensional embeddings



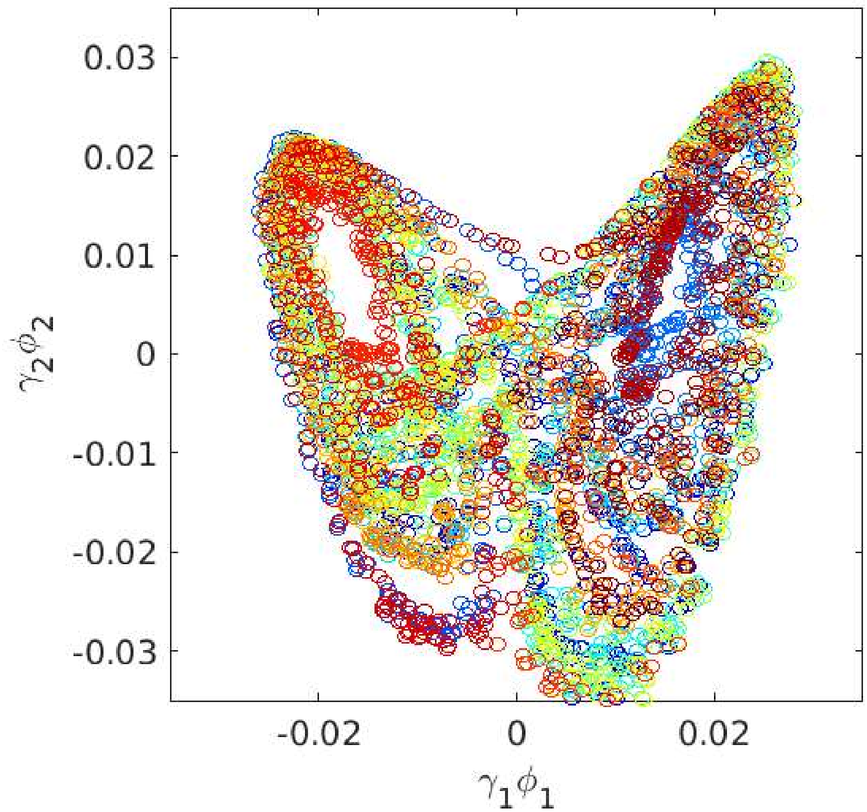
Constructing basis functions

Example: Data produced from Lorenz (1963) model

Observations



Two-dimensional embeddings



Unbiased Gaussian errors