Localization modeling with a neural network, toward strongly coupled atmosphere-ocean data assimilation

May 30, 2019

The 17th JCSDA Technical Review Meeting and Science Workshop

Takuma Yoshida and Eugenia Kalnay

Outline – Localization modeling with a neural network

1. Background and approach
2. Experiments with toy “correlation models”
3. Background error correlations in the atmosphere-ocean systems
4. Experiment as a localization function

Localization for atmos-ocean strongly coupled DA

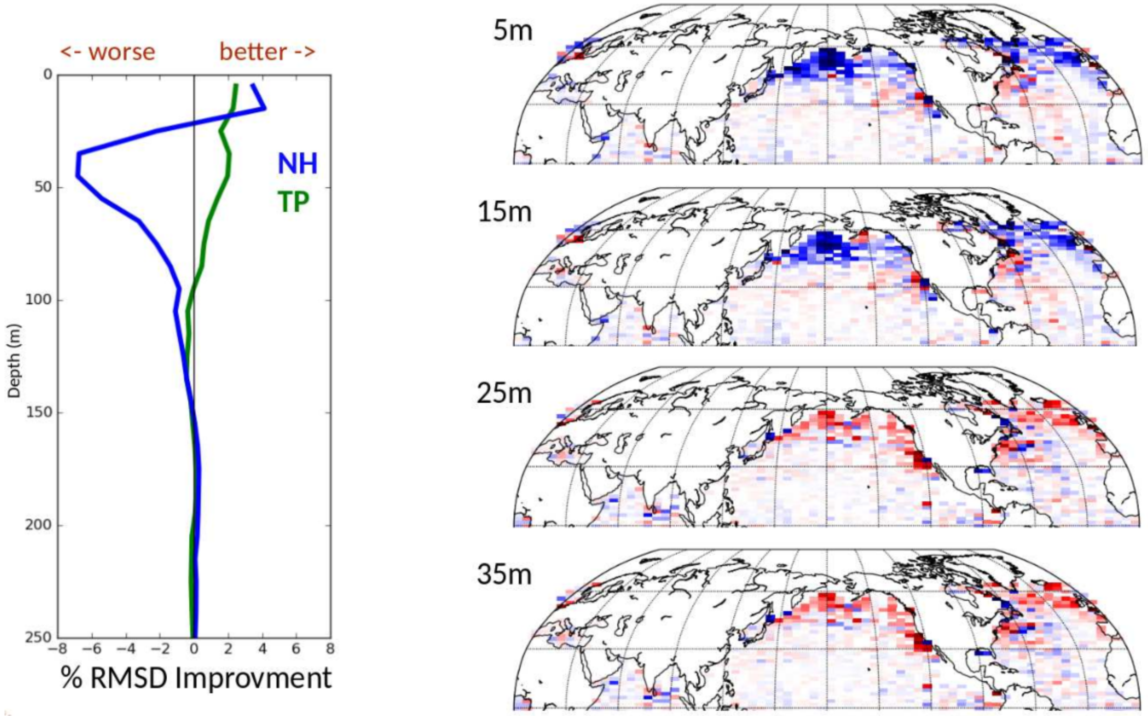


Figure 3.13: STRONG - WEAK change in observation minus forecast (O-F) RMSD for ocean temperature. Averaged over the tropics (TP) and Northern Hemisphere (NH) at various depths (left) and shown spatially (right). For the spatial plot blue is an RMSD improvement, red is a degradation.

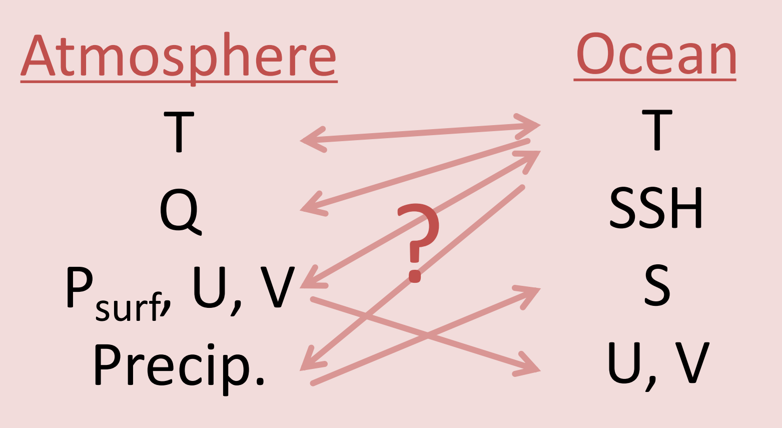
Homework from Travis Sluka (2018)

› *Using vertical and variable localization as a form of correlation cutoff method is vital when using a limited ensemble size. These cross-domain state variables with small correlations need to be removed in the data assimilation step in order to avoid the detrimental impact of spurious correlations in the ensemble.*

Localization not based on distance

- Variable localization (Kang et al., 2011)
  - By not assimilating CO_2 concentration observation to humidity, for example, the analysis accuracy improves (LETKF experiments with dynamics-carbon coupled model)

- Then how can we optimize localization for coupled Earth system models with growing complexity?
  - We may use physical intuitions as Kang et al.
  - We don't want to try switching on/off assimilation between every pair of variables



What is a localization function (obs space)?

Attributes of analysis variable x_i

- Latitude/longitude/level
- Analysis variable type
- Time (e.g., seasonal, diurnal)

Attributes of observable y_j

- Latitude/longitude/level/wavelength
- Observation type

Some multivariate function

Localization weight $[\rho]_{ij}$ $(0 \leq [\rho]_{ij} \leq 1)$

Mean squared error correlation as a "distance"

Reduction of analysis error variance by single-observation assimilation

- The relative accuracy of the observation to the background (~ DFS in the observation space)
- The square of background error correlation between observed and analyzed variables

$$\frac{\sigma_{\mathrm{b}i}^2 - \sigma_{\mathrm{a}i}^2}{\sigma_{\mathrm{b}i}^2} = \frac{[(\mathbf{HB})_{1i}]^2}{(\mathbf{HBH}^{\mathrm{T}} + R)B_{ii}} = \frac{\sigma_{y\mathrm{b}}^2}{\sigma_{y\mathrm{b}}^2 + \sigma_{y\mathrm{o}}^2}\mathrm{corr}^2(\delta x_{\mathrm{b}i}, \delta y_{\mathrm{b}})$$

We should only assimilate observations whose background error is well-correlated with the analysis variable's background error (correlation-cutoff method)

- Improves EnKF with a toy atmosphere-ocean coupled model (Yoshida and Kalnay, 2018 MWR)

How to construct a localization function?

Attributes of analysis variable x_i

- Latitude/longitude/level
- Analysis variable type
- Time (e.g., seasonal, diurnal)

Attributes of observable y_j

- Latitude/longitude/level/wavelength
- Observation type

Some multivariate function

Localization weight
$[\rho]_{ij}$ $(0 \leq [\rho]_{ij} \leq 1)$

How to construct a localization function?

Attributes of analysis variable x_i

- Latitude/longitude/level
- Analysis variable type
- Time (e.g., seasonal, diurnal)

Attributes of observable y_j

- Latitude/longitude/level/wavelength
- Observation type

Localization function in observation space $g \circ f$

Nonlinear regression f

Expected strength of background error correlation between (x_i, y_j)

Increasing function g

Localization weight $[\rho]_{ij}$ $(0 \leq [\rho]_{ij} \leq 1)$

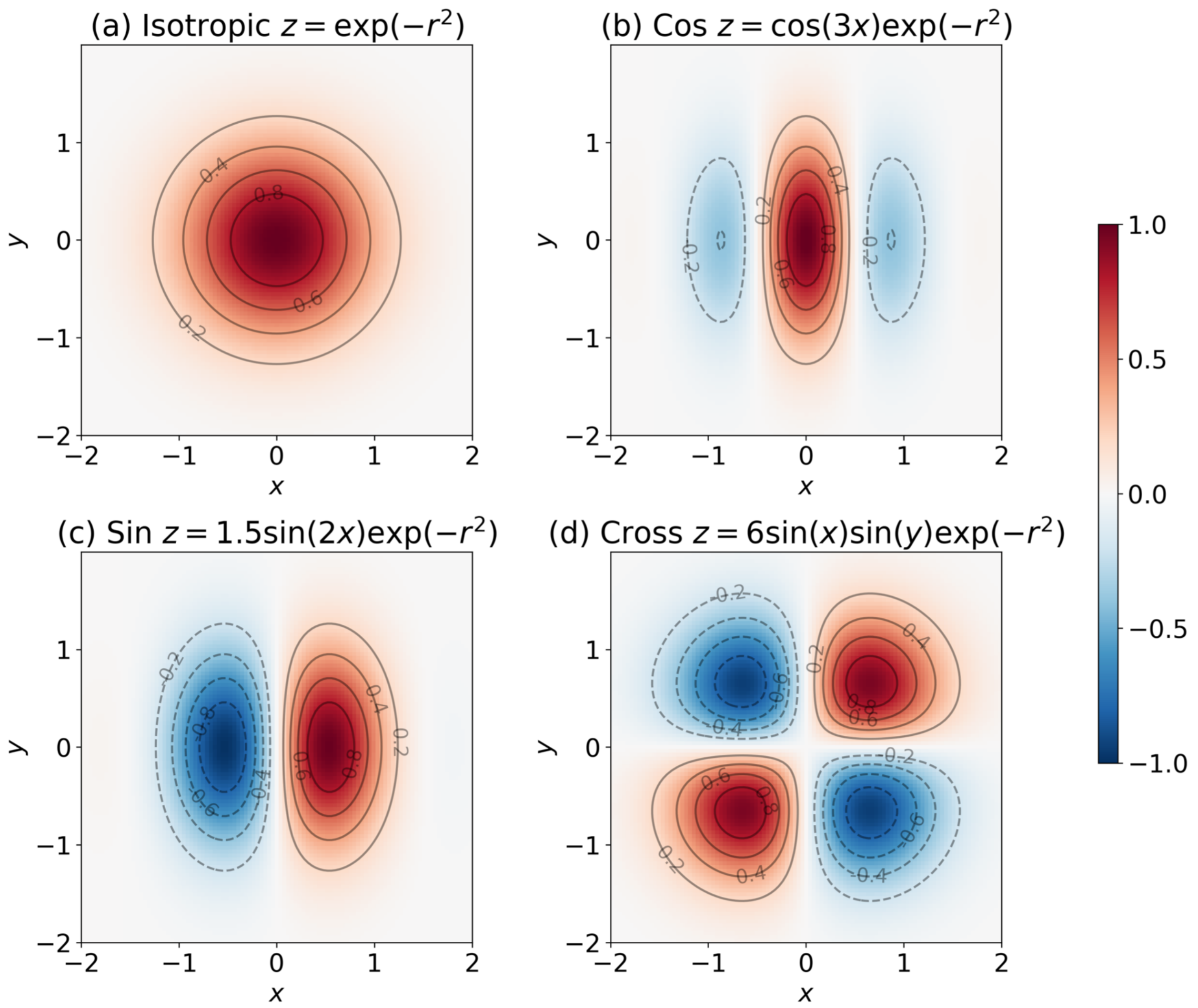
This is univariate and far easier

This can be done with a neural network (or other nonlinear regression) and data from ensemble DA

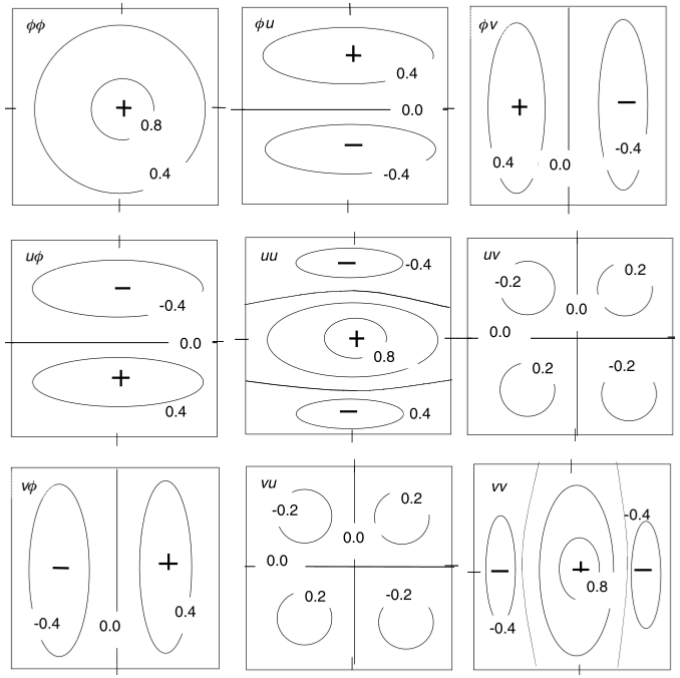
Outline – Localization modeling with a neural network

1. Background and approach
2. Experiments with toy “correlation models”
3. Background error correlations in the atmosphere-ocean systems
4. Experiment as a localization function

Toy correlation models



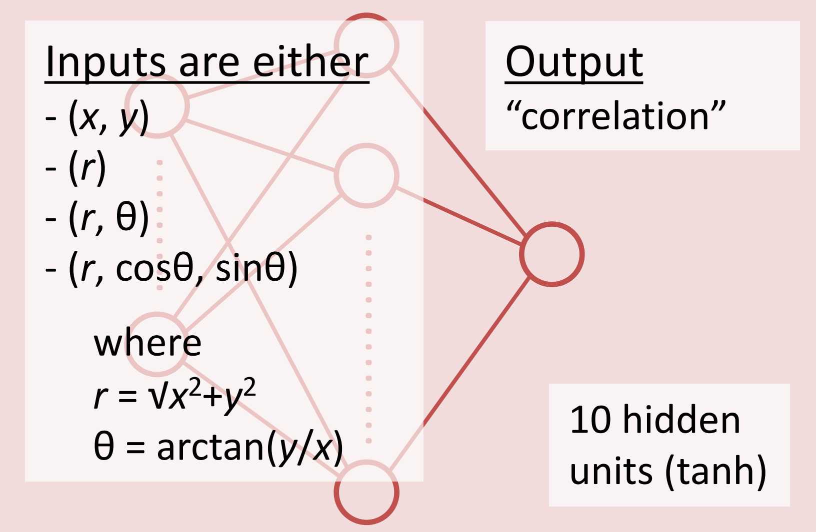
Multivariate error correlations under geostrophy



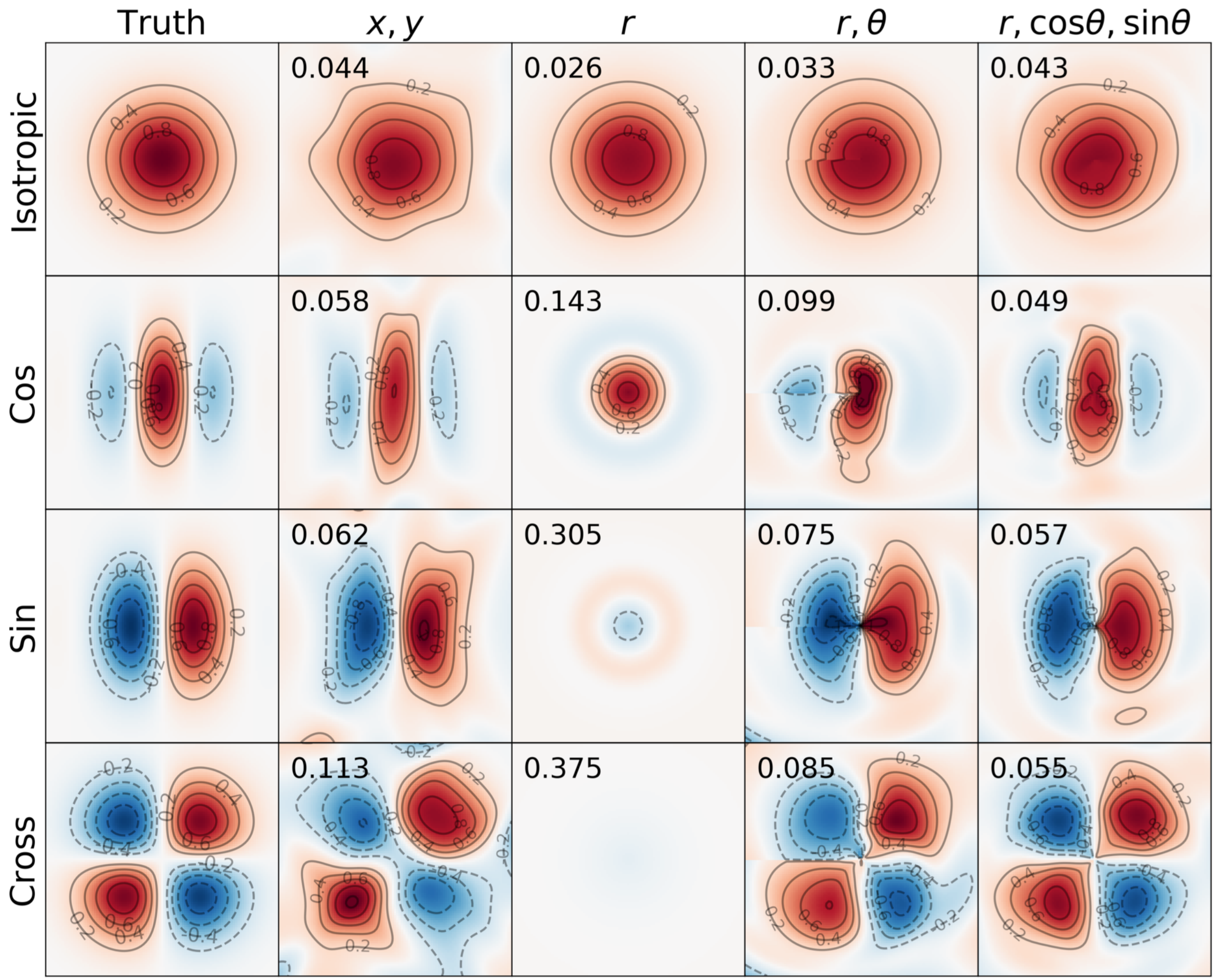
Neural network + noisy training data

For each correlation model,
we prepare 1000 training data:

1. Randomly sample $|x|, |y| < 2$
2. Calculate value at (x, y) and
   add Gaussian error with STDV = 0.2



$$\mathbf{q} = \mathbf{W}_2 \tanh[\mathbf{W}_1\,\mathbf{p} + \mathbf{b}_1] + \mathbf{b}_2$$

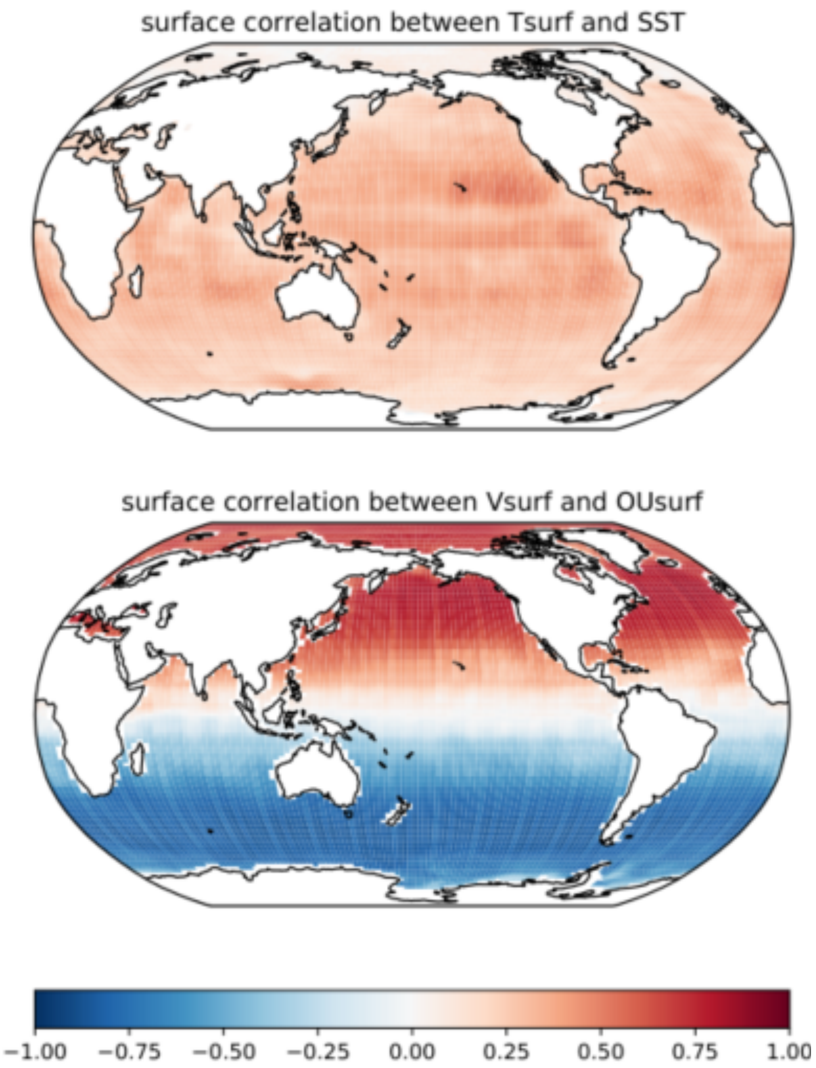


Superimposed numbers are RMS regression error to validation dataset

Outline – Localization modeling with a neural network

1. Background and approach
2. Experiments with toy “correlation models”
3. Background error correlations in the atmosphere-ocean systems
4. Experiment as a localization function

Surface ensemble correlations (FOAM, 1.4-2.8 deg ocean)



Surface air T and SST

› Error correlation of 0.4 means strongly coupled DA can reduce up to 16% of variance

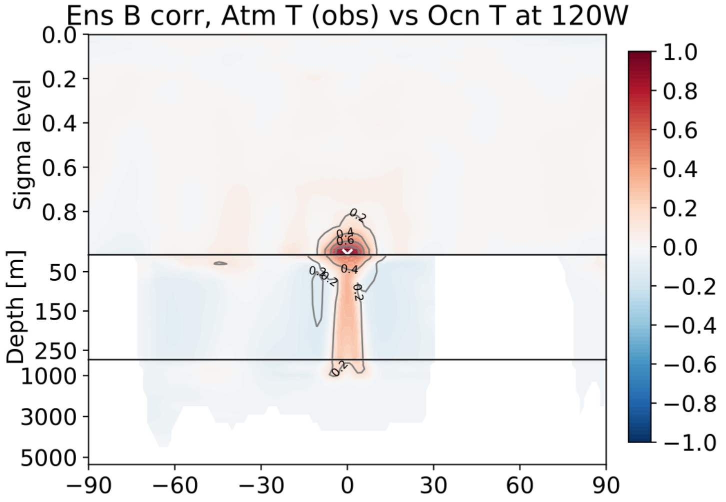
V-wind and U-current

› Can be explained by the linearized Ekman layer dynamics, where subsurface water transport is to the right of wind direction in NH

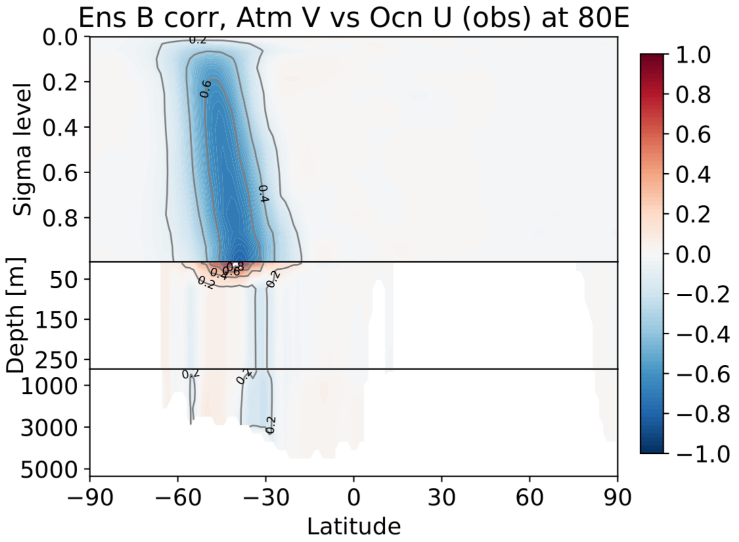
Two strongest (pointwise) mean **B** correlations seen in FOAM-LETKF

FOAM stands for Fast Ocean Atmosphere Model (Jacob 1997)

More examples of FOAM ensemble correlations



An air-T observation can constrain temperature analyses of atmosphere and ocean

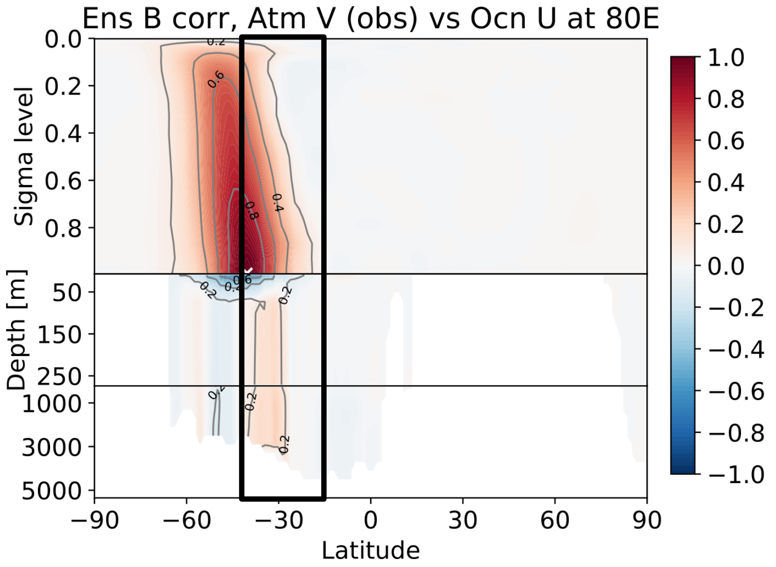


A surface U-current observation can constrain U-current and V-wind fields

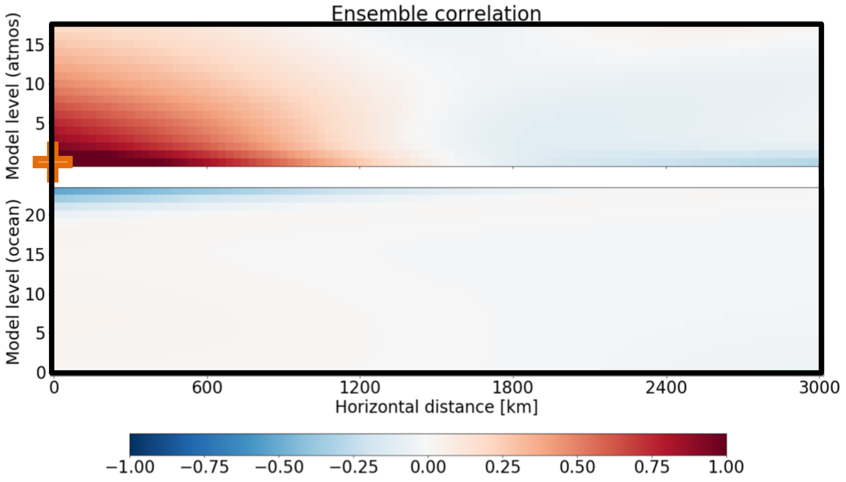
B correlation to an observation background: proxy to single observation assimilation increment (normalized by **B** variance)

Contour is RMS of ensemble correlation, which includes time-dependent (or flow-dependent) portion

Error correlation reproduced by neural networks

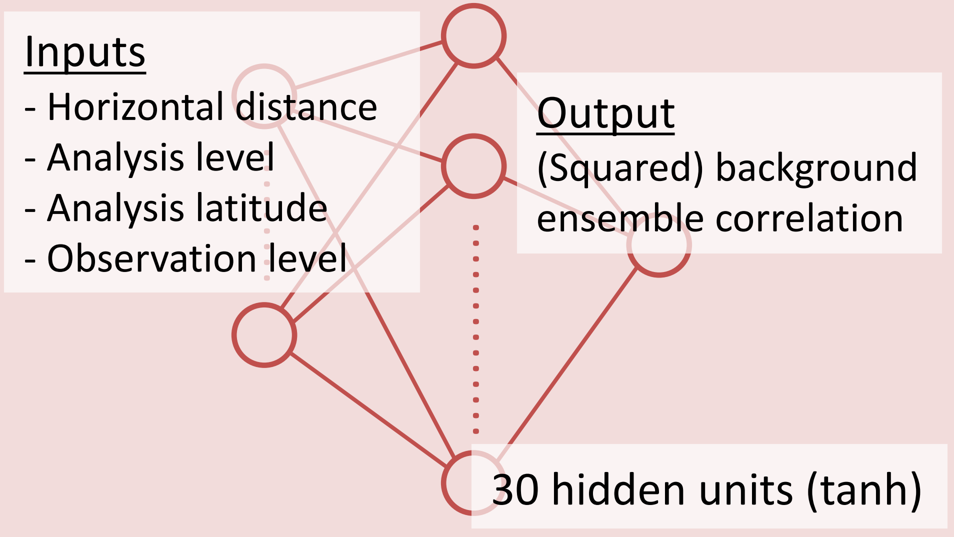


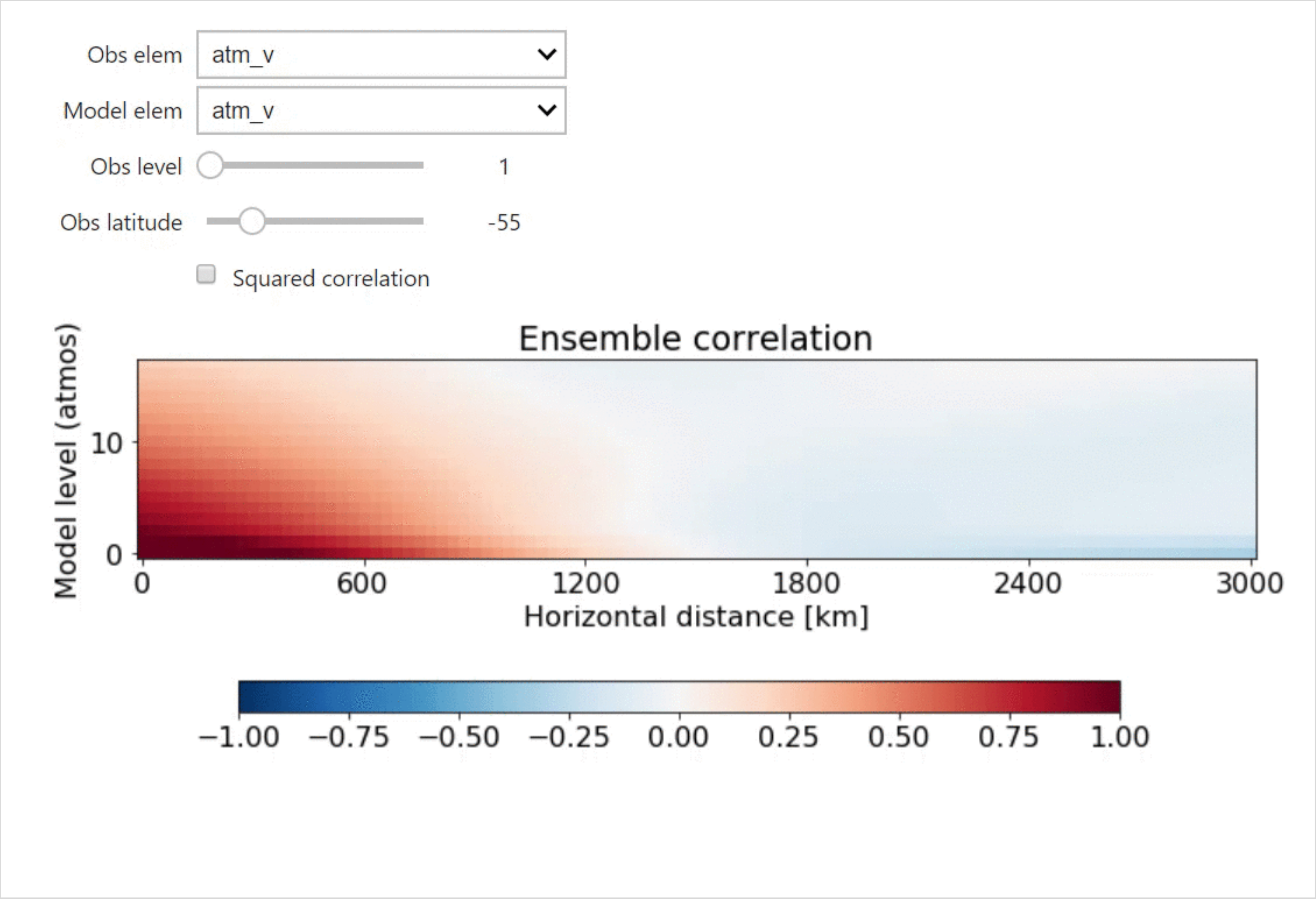
Many instances simply averaged



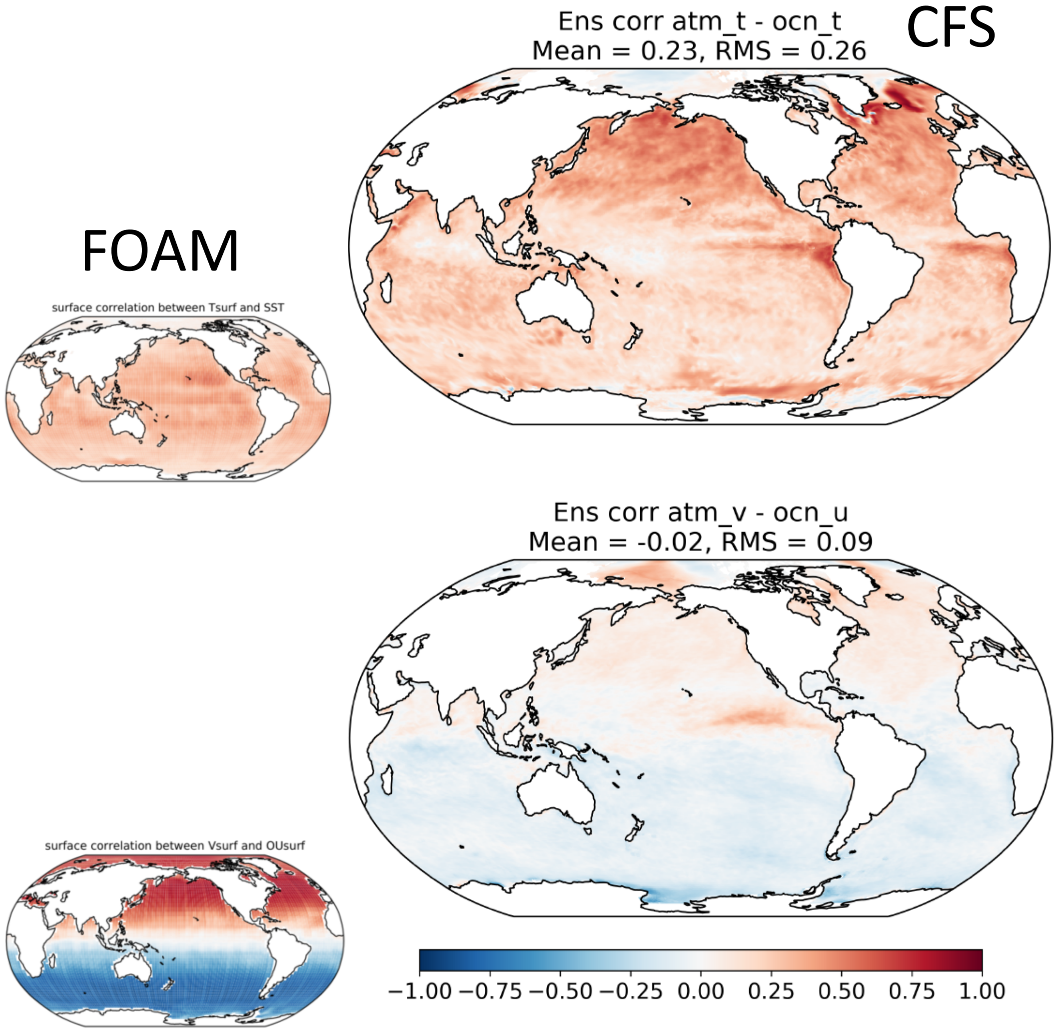
B correlation reproduced by neural networks

One network for each
- Observation type
- Analysis variable type




Obs elem
atm_v
Model elem
atm_v
Obs level
1
Obs latitude
-55
Squared correlation
Ensemble correlation
Model level (atmos)
10
0
0
600
1200
1800
2400
3000
Horizontal distance [km]
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

CFSv2-LETKF (0.25-0.5 deg ocean; 40 mem; mean of June 2006)



- Strong T-T correlation at sea ice (FOAM lacks) boundaries and equatorial upwelling
- The small wind-current error correlations in CFS may be associated to more internally chaotic ocean current in CFS
  - i.e., less dependent to atmosphere
  - Estimation by a single column model without horizontal dynamics (Smith et al., 2017) had the similar conclusion with FOAM
- Every model is a limited representation of the truth or the error structure

Special thanks to Travis Sluka, Sreenivas Pentakota, and Eviatar Bach

Outline – Localization modeling with a neural network

1. Background and approach
2. Experiments with toy “correlation models”
3. Background error correlations in the atmosphere-ocean systems
4. Experiment as a localization function

Attributes of analysis variable x_i
- Latitude/longitude/level
- Analysis variable type
- Time (e.g., seasonal, diurnal)

Attributes of observable y_j
- Latitude/longitude/level/wavelength
- Observation type

Localization function in observation space $g \circ f$

Nonlinear regression f

Expected strength of background error correlation between (x_i, y_j)

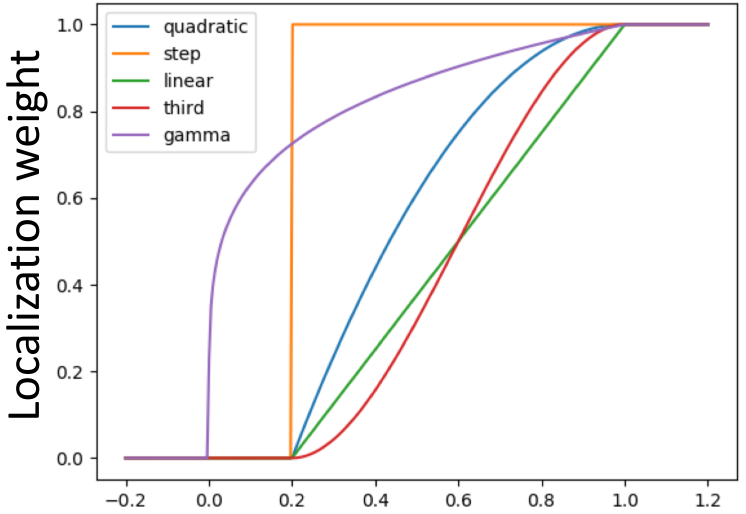
Increasing function g

Localization weight $[\rho]_{ij}$ $(0 \leq [\rho]_{ij} \leq 1)$

Preliminary 64-member single window OSSE

- For neural net-based experiment (*"neural"*), piecewise quadratic function with cutoff squared correlation of 0.2 is used
  - *Neural* experiment is not tuned at all. Results are just for proof-of-concept
- Control SCDA experiment (*"control"*) uses
  - 1000 km horizontal localization for atmosphere
  - 400 km horizontal localization for ocean
  - 3 levels vertical localization for atmosphere
  - 5 levels vertical localization for ocean
  - Frolov et al. (2016) cross-localization

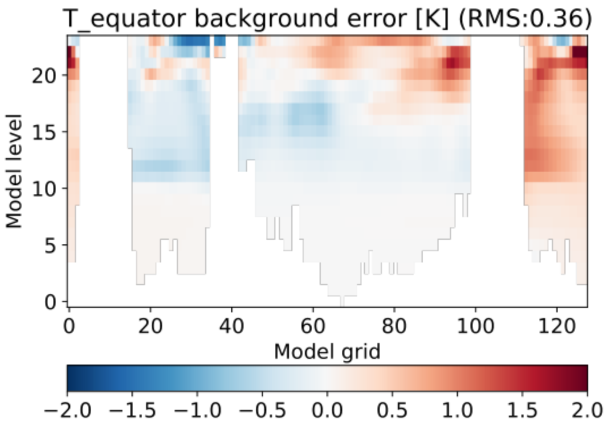
Examples of non-decreasing function *g*



$$\bar{\zeta}_A - \bar{\zeta}_O = \sqrt{\left[\frac{x_A - x_O}{0.5(L_{x|A} + L_{x|O})}\right]^2 + \left[\frac{y_A - y_O}{0.5(L_{y|A} + L_{y|O})}\right]^2 + \cdots + \left(\frac{z_A}{L_{z|A}} - \frac{z_O}{L_{z|O}}\right)^2 + \left[\frac{t_A - t_O}{0.5(L_{t|A} + L_{t|O})}\right]^2}$$

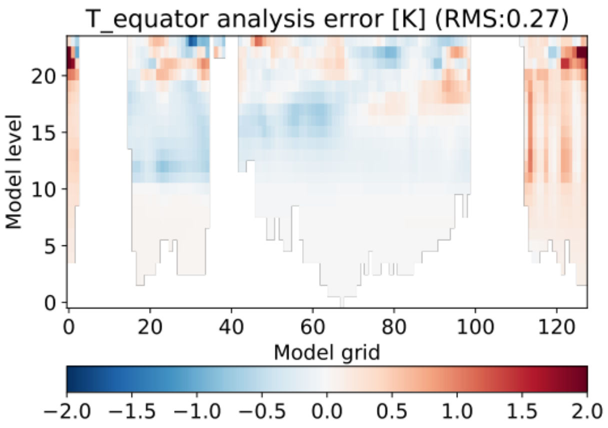
First window analysis: ocean T at equator

(Common) background error

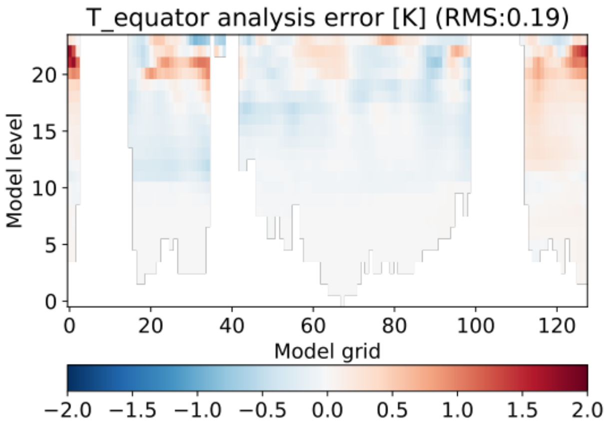


Neural experiment has smaller analysis error throughout the depth and also have smoother error field

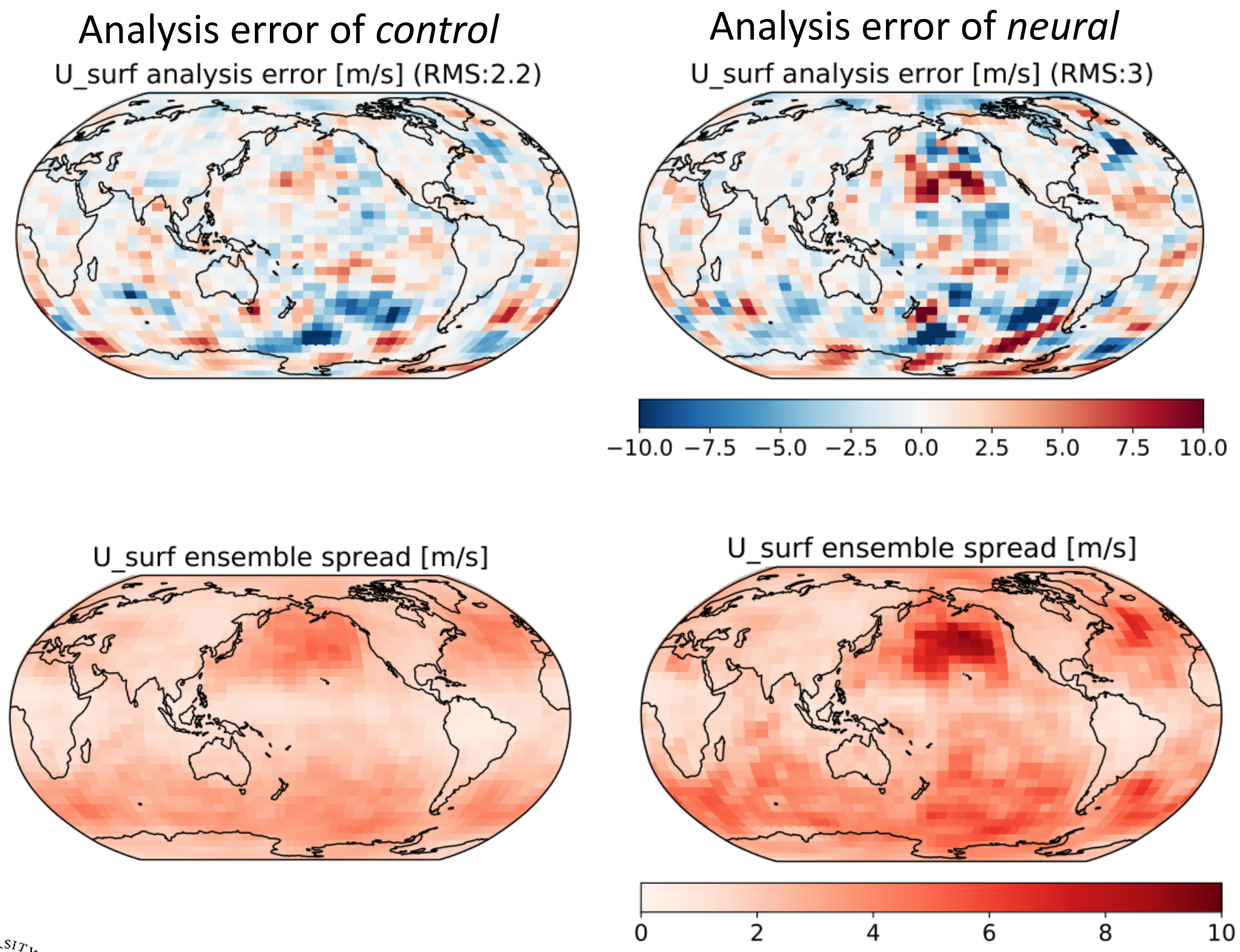
Analysis error of *control*



Analysis error of *neural*



First window analysis: atmosphere surface U



- *Neural* experiment has larger analysis error over ocean (where wind is not directly observed)
  - It seems discounting the U (analyzed) - T (observed) correlation too much

- It has good error-spread relationship for both observed and unobserved regions
  - What if cycled?

Computation cost (compared to an ordinary localization function)

- Computation time taken for three windows of analyses (including fcst)
  - Both evaluates up to 3000 km distance and ±16 vertical levels
  - SCDA with ordinary localization function (Frolov et al. 2016)
    - 5 mins 48 secs
  - SCDA with neural net-based weighting
    - 6 mins 8 secs (+5.7%)
    - The neural net evaluation is roughly less than 10% of total LETKF computation
- Direct increase of computation time is acceptable
  - Indirect increase is expected due to increased number of observations whose localization weight is actually evaluated (common problem for any advanced localization method; alleviated combined with physical cut-off distance)

Summary – Localization modeling with a neural network

- Variable and spatial localization is key for strongly coupled DA
- Mean squared ensemble correlation can be used as a “distance”
  - An objective criterion for spatial/variable localization (correlation-cutoff method)
- Neural networks can be used for localization modeling
  - Mathematically qualified as a localization function for serial EnSRF and LETKF
  - Acceptable training and evaluation costs
  - We’re working on cycle DA experiments
- Training data is available from ensemble DA
  - However, error correlation structure highly depends on processes resolved by the coupled models

Backups

Derivation

We start our derivation from the state-update equations of the Kalman filter (Kalman 1960). Assuming that the background error covariance $\mathbf{B}$ and observation error covariance $\mathbf{R}$ are correctly specified, and that the observation errors are not correlated with the background errors, the analysis error covariance $\mathbf{A}$ is given by

$$\mathbf{A} = (\mathbf{I} - \mathbf{KH})\mathbf{B}, \quad (1)$$

$$\mathbf{K} = \mathbf{BH}^{\mathrm{T}}(\mathbf{HBH}^{\mathrm{T}} + \mathbf{R})^{-1}, \quad (2)$$

where $\mathbf{K}$ is the Kalman gain, $\mathbf{I}$ is the identity matrix, and $\mathbf{H}$ is a linearized observation operator (e.g., Gelb et al. 1974).

Consider the analysis error variance of the ith model variable,

$$A_{ii} = B_{ii} - \sum_{k=1}^{n}\sum_{l=1}^{p} K_{il}H_{lk}B_{ki}, \quad (3)$$

where n is the size of the state vector and p is the number of observations. The fractional decrease of the uncertainty for the ith model variable is given by

$$\frac{B_{ii} - A_{ii}}{B_{ii}} = \frac{\sum_{k=1}^{n}\sum_{l=1}^{p} K_{il}H_{lk}B_{ki}}{B_{ii}}. \quad (4)$$

Assuming that there is only one observation ($p = 1$), the observation error variance can be expressed by a scalar as $R = \sigma_{y_o}^2$. With this assumption, we can reduce Eq. (4) to

$$\frac{B_{ii} - A_{ii}}{B_{ii}} = \frac{[(\mathbf{HB})_{1i}]^2}{(\mathbf{HBH}^{\mathrm{T}} + R)B_{ii}}, \quad (5)$$

where we used the Kalman gain [Eq. (2)] and the single observation assumption repeatedly (note that $\mathbf{HB}$ and $\mathbf{HBH}^{\mathrm{T}}$ are a $1 \times n$ matrix and a scalar, respectively). We then rewrite the covariance between the background errors of the observable (δy_b) and the ith model variable (δx_{bi}) as a product of their correlation and standard deviations (σ_{y_b} and $\sigma_{bi} = \sqrt{B_{ii}}$ for the observable and the ith model variable, respectively) as $(\mathbf{HB})_{1i} = \sigma_{bi}\sigma_{y_b}\mathrm{corr}(\delta x_{bi}, \delta y_b)$.

We finally obtain

$$\frac{\sigma_{bi}^2 - \sigma_{ai}^2}{\sigma_{bi}^2} = \frac{\sigma_{y_b}^2}{\sigma_{y_b}^2 + \sigma_{y_o}^2}\mathrm{corr}^2(\delta x_{bi}, \delta y_b), \quad (6)$$

where $\sigma_{ai} = \sqrt{A_{ii}}$ is the standard deviation of the analysis error of the ith model variable [a similar derivation for a two-variable example is provided in Hamill et al. (2001)]. It is informative to compare this equation with the analysis uncertainty reduction in the univariate analysis, in which a single state variable is directly observed by a single observation,

$$\frac{\sigma_b^2 - \sigma_a^2}{\sigma_b^2} = \frac{\sigma_b^2}{\sigma_b^2 + \sigma_o^2}, \quad (7)$$

where σ_b^2, σ_a^2, and σ_o^2 are the error variances for the background, analysis, and observation, respectively. Equation (6) is similar to Eq. (7), except that the right-hand side is multiplied by the square of the correlation between the background errors of the analyzed and observed variables.

UNIVERSITY OF MARYLAND 18 56

Reasons for choosing neural networks

Method	Advantages	Disadvantages
Linear regression	Simple to implement Training is analytical	Linear
Lookup table	Nonlinear Simple to implement Training is analytical	Discontinuous Assumptions for boundaries Curse of dimensionality
Linear combination of nonlinear basis functions (e.g., polynomial fit)	Nonlinear Training is analytical	Assumptions for basis functions Curse of dimensionality
Neural network	Nonlinear Fewer assumptions Relatively tolerant to input dimensionality	Training requires iteration

Distance-only localization

Attributes of analysis variable x_i

- Latitude/longitude/level
- Analysis variable type
- Time (e.g., seasonal, diurnal)

Attributes of observable y_j

- Latitude/longitude/level/wavelength
- Observation type

Physical distance between x_i, y_j

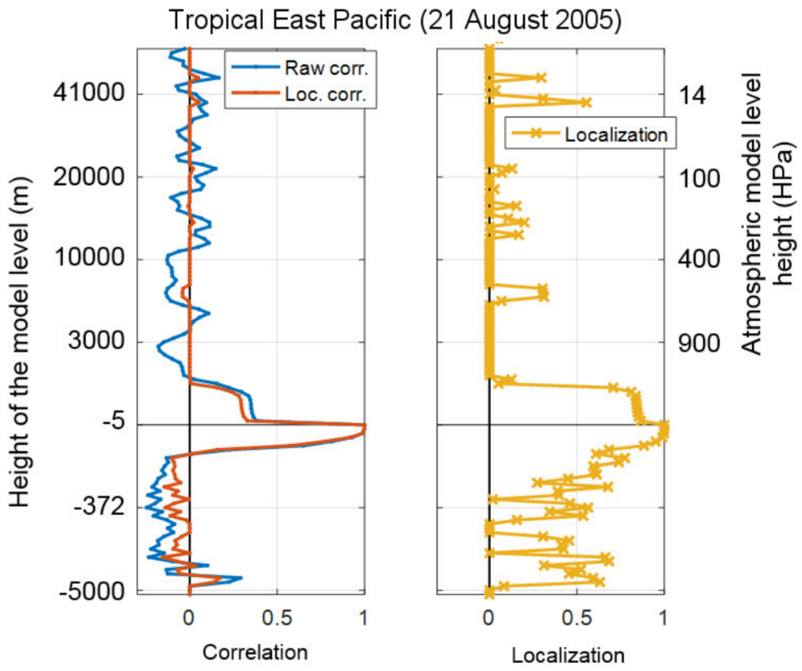
Some decreasing function (e.g., Gaspari and Cohn, 1999)

Localization weight $[\rho]_{ij}$ $(0 \leq [\rho]_{ij} \leq 1)$

$$C(\mathbf{x}, \mathbf{y}) := C_1(\mathbf{x} - \mathbf{y}) := C_0(\|\mathbf{x} - \mathbf{y}\|),$$

Why we bother to use *mean* squared correlations?

- There exist a few adaptive localization methods to estimate localization function from *instantaneous* ensemble correlations
- Laloyaux et al. (2018) for SCDA:
  - › *the localisation function diagnosed from Ménétrier et al. (2015a,b) is also noisy because of the limited ensemble size (25 members). To fully remove these spurious correlations, several solutions could be considered and combined:*
    - averaging the localisation diagnostic over several cycles to get a cleaner localisation function



Localization function to SST observation, estimated by instantaneous ensemble correlations

What function can be a localization function?

- Well-known sufficient condition for model-space localization:
  - If localization matrix is positive semidefinite, then the analysis is well-defined
  - This is restrictive when constructing a localization function because one localization weight (between a pair of analysis variables) cannot contradict with the other localization weights

- Obs-space localizations for serial EnSRF and LETKF are less restrictive:
  - If only each localization weight (between each pair of an analysis variable and an observable) is nonnegative, then the analysis is well-defined
  - We can choose localization weights independently from each other (at least mathematically)

Gaspari and Cohn (1999)

Validity as a localization function: Serial EnSRF

- As long as localization weight ρ_{ij} between the i^{th} analysis variable and the j^{th} observation is within [0, 1], update by the single observation is well-defined
  - $\rho_{ij}\mathbf{H}_j\mathbf{X}(\mathbf{H}_j\mathbf{X})^{\mathrm{T}} + \mathbf{R}_{jj}$ is positive scalar if $\mathbf{R}_{jj} > 0$ and $\rho_{ij} \geqq 0$
- Recursively, assimilation of any set of observations is well-defined
  - However, combined with localization, the final analysis depends on the order of assimilation (e.g., Kotsuki et al. 2017)

Validity as a localization function: LETKF

- As long as the localization weight ρ_{ij} between i^{th} analysis variable and j^{th} observation is within [0, 1], local update is well-defined
  - For an analysis of i^{th} analysis variable, localization substitutes as $[\mathbf{R}^{-1}]_{jk} \leftarrow \sqrt{(\rho_{ij}\rho_{ik})}[\mathbf{R}^{-1}]_{jk}$ for observations j and k. First deal with nonzero weights; with $\mathbf{D} = \mathrm{diag}(\sqrt{\rho_{i1}}, \ldots, \sqrt{\rho_{ip}})$, $\mathbf{R}^{-1}$ is replaced with $\mathbf{D}\mathbf{R}^{-1}\mathbf{D}$. However, $(\mathbf{D}\mathbf{R}^{-1}\mathbf{D})^{-1} = \mathbf{D}^{-1}\mathbf{R}\mathbf{D}^{-1}$ meaning observation error correlation matrix is unchanged but just variances are changed. For zero weights, we can use the fact that principal submatrices of a positive-semidefinite matrix are positive-semidefinite. These operations keep the matrix positive semidefinite
  - This ensures $(k\text{–}1)\mathbf{I} + \mathbf{Y}^{T}\mathbf{R}^{-1}\mathbf{Y}$ is positive definite symmetric

Computation cost is acceptable

- Sampling
  - 1E+9 pairs *total* for 100 pairs of variable types
  - Several hours with a single processor. Parallelizable
- Learning
  - 8E+6 samples times 3 epochs for *each* pair of observation and analysis variables
  - Tens of minutes with a single processor. Parallelizable

Except for IO, the training cost will be almost independent of model resolution.

- Evaluation of a neural network
  - O(100p) floating point computation for each analysis variable. This is less than LETKF's cost O($k^3 + pk^2$) (p: # of local observations, k: ensemble size)

Known computational issue

- For purely distance-based localization, there is established and fast algorithms to select nearby observations
  - › k-D tree, octree, and other spatial decompositions
  - › One distance evaluation can reject many faraway observations
- Nothing similar is available for selecting relevant observations for neural net-based algorithm
  - › To alleviate the lookup cost, I do not evaluate observations outside a 3000 km range in current experiments

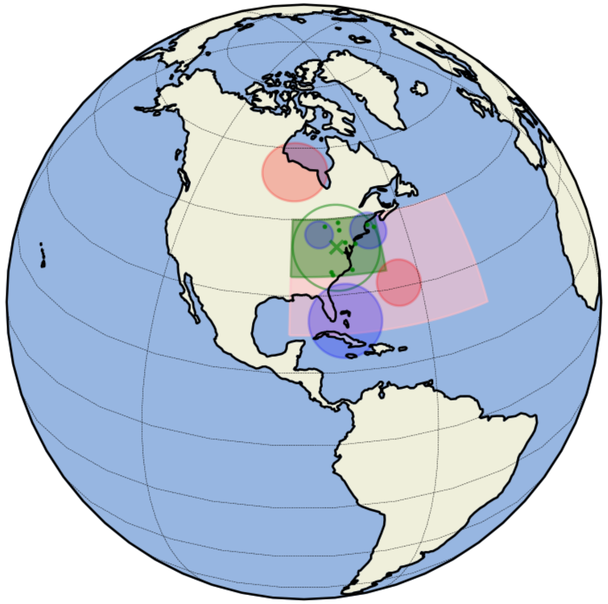


Figure A.1: Horizontal schematic of Octree application on spherical coordinate. The pink box is one of the three-time-divided boxes, which is 45 degrees in longitude and 22.5 degrees in latitude. The green box is one of the four-time-divided boxes. The green cross, dots, and circle show the *center* of the green box (at 39.375°N 78.75°W), hypothetical observations in the green box, and the *radius* of the box after adding the observations (approximately 943 km), respectively. Note that the observations (green dots) exist only in the intersection of the green box and circle. Queries like the blue (red) circles intersect (does not intersect) with the green circle and may (may not) find observations in the green box.

Another computational issue

- For LETKF that uses same localization weight for all the model variable types, we can reuse analysis weights ($\mathbf{w}^{\mathrm{bar}}$ and $\mathbf{W}$) for collocated analysis variables
  - Mentioned in Hunt et al. (2007), implemented in Miyoshi’s SPEEDY-LETKF
  - This reduces the analysis cost by few times
- In principle, we cannot reuse analysis weights for localization method that considers variable types
  - This is also true for “variable localization” of Kang et al. (2011)

Training samples of FOAM

1. Sample time t uniformly randomly from the available period.

2. Sample analysis grid index (i, j, k) uniformly randomly on the model coordinate. Repeat sampling if the analysis grid is topographically masked. For vertical coordinate, model levels are directly used as in Figure XXX for simplicity.

3. Sample uniformly random $(r, \theta, z_{\text{obs}})$ $(r \in [0, r_{\max}), \theta \in [0, 2\pi))$, where $r_{\max} =$ 3000 km, θ is the azimuth angle, and z_{obs} is observation levels in model levels[8].

4. Using analysis location and $(r, \theta, z_{\text{obs}})$, get the observation location. Repeat sampling (step 3-4) if observable is topographically masked.

5. Get background ensemble of the analysis variable and the observable by applying observation operator to the background ensemble of state vectors. Here, observation operators are just horizontal interpolation.

6. Calculate and save the background ensemble (sample) correlation between the analysis variable and the observable.

Comparison: FOAM and CFS oceans