Advances in spatio-temporal modeling of ocean heat content with Argo floats

Thea Sukianto[1], Mikael Kuusela[1], Donata Giglio[2]

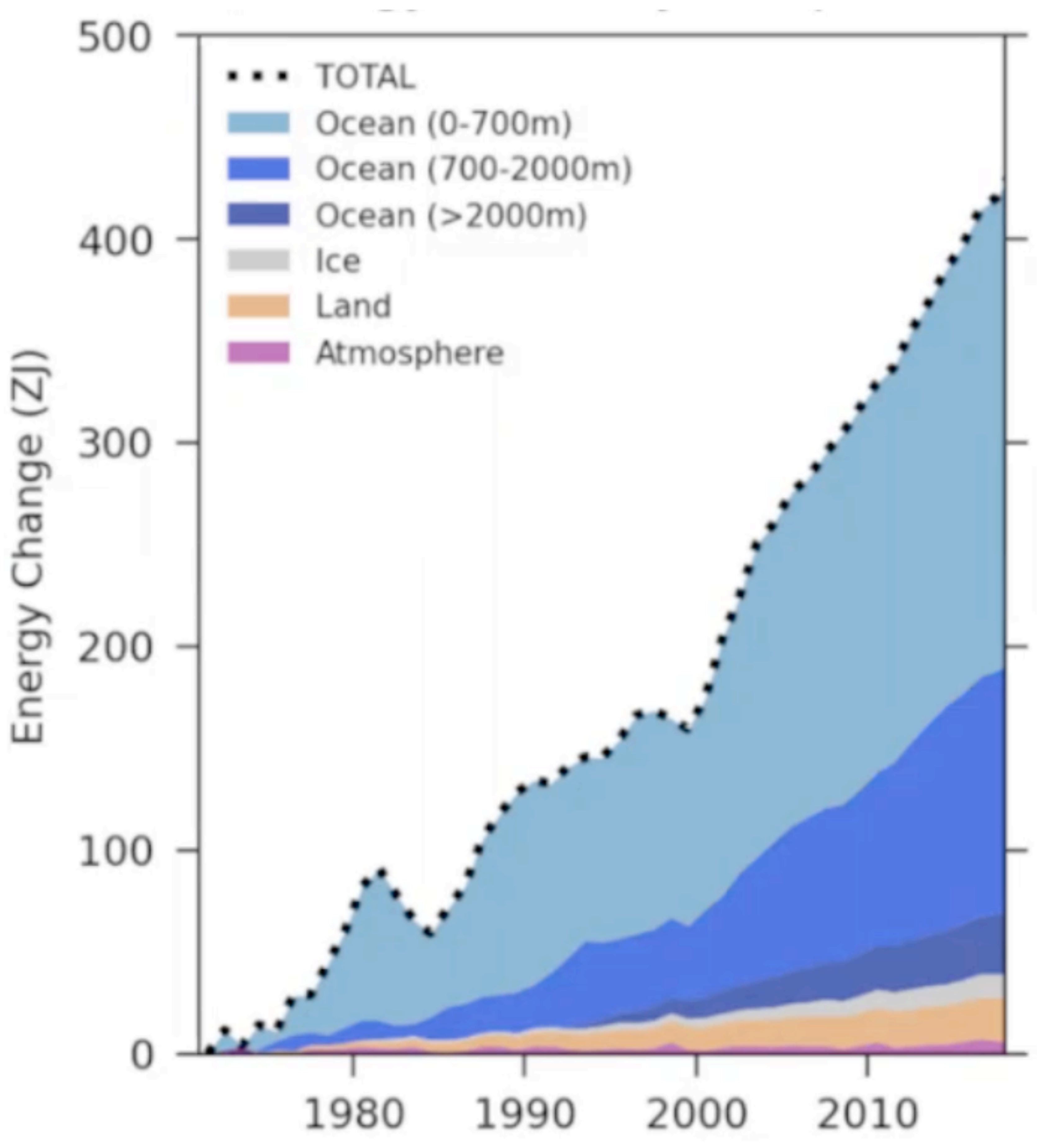
[1]Department of Statistics and Data Science, Carnegie Mellon University
[2]Department of Atmospheric and Oceanic Sciences, University of Colorado Boulder

7th Spatial Statistics Conference
Noordwijk, The Netherlands
16 July 2025

Carnegie Mellon University
Statistics & Data Science

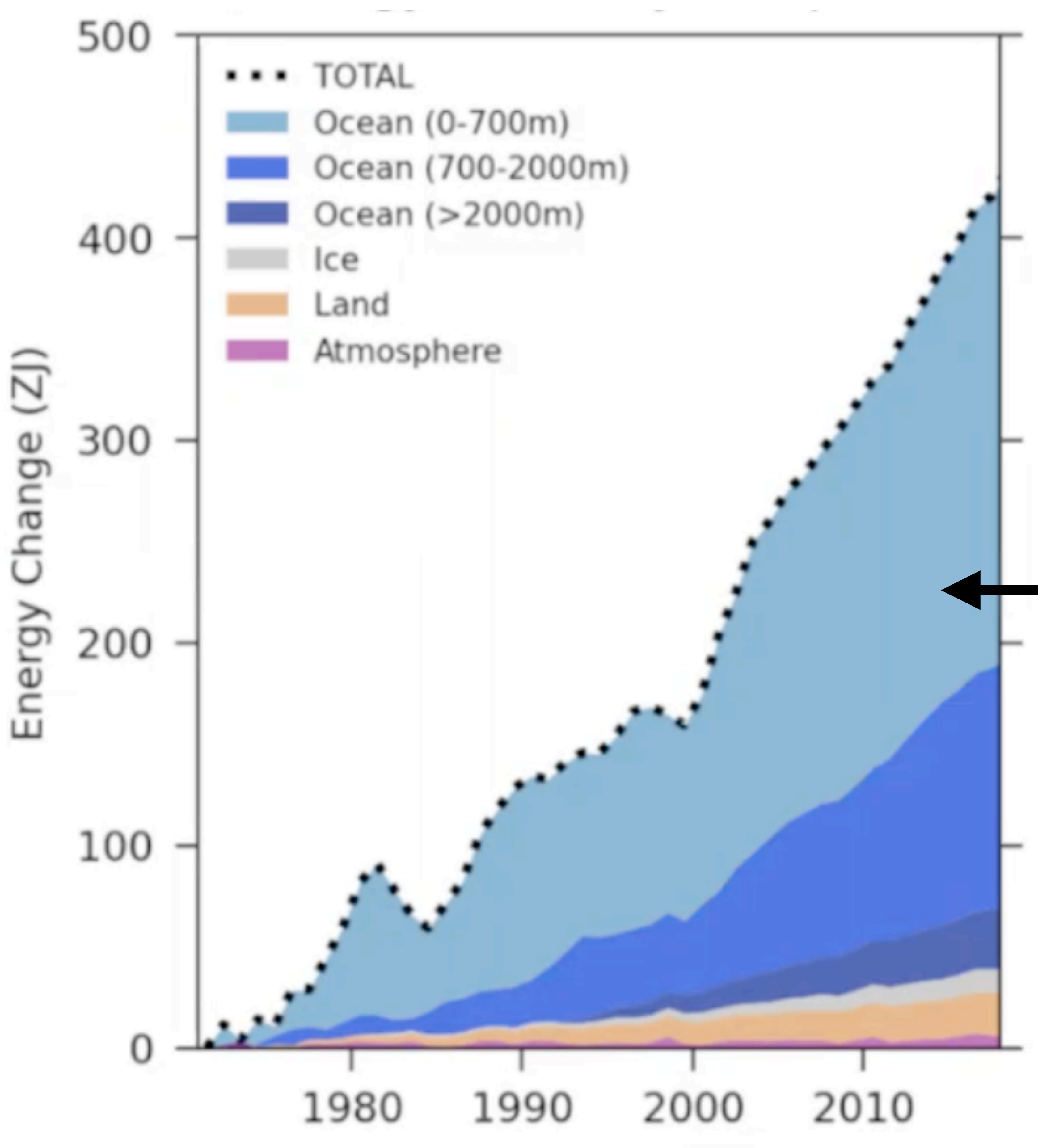
University of Colorado
Boulder

Changes in OHC contribute to extreme climate events



(IPCC AR6)

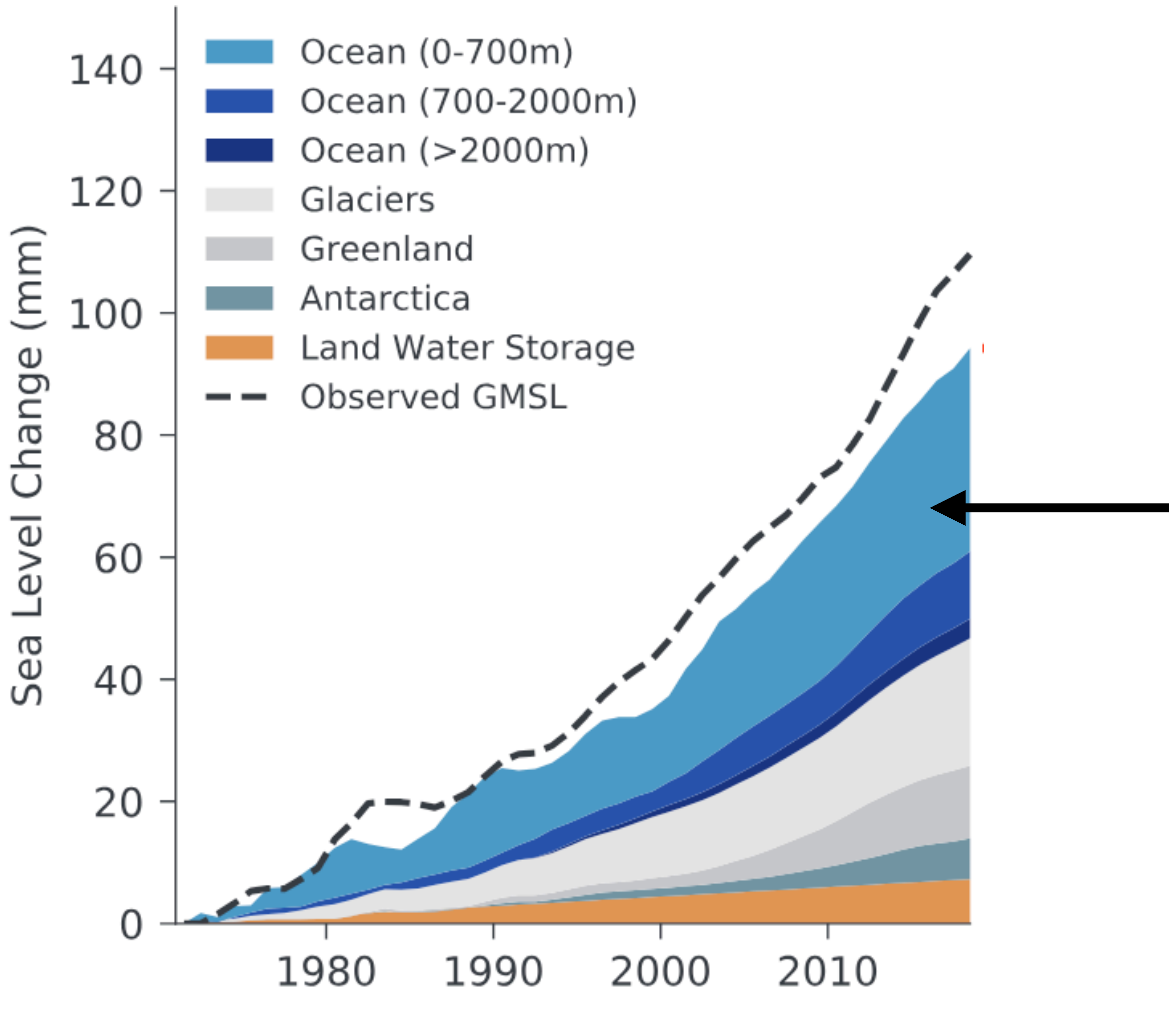
Changes in OHC contribute to extreme climate events



(IPCC AR6)

~91% of the excess heat in the climate system is stored **in the ocean**.

Changes in OHC contribute to extreme climate events

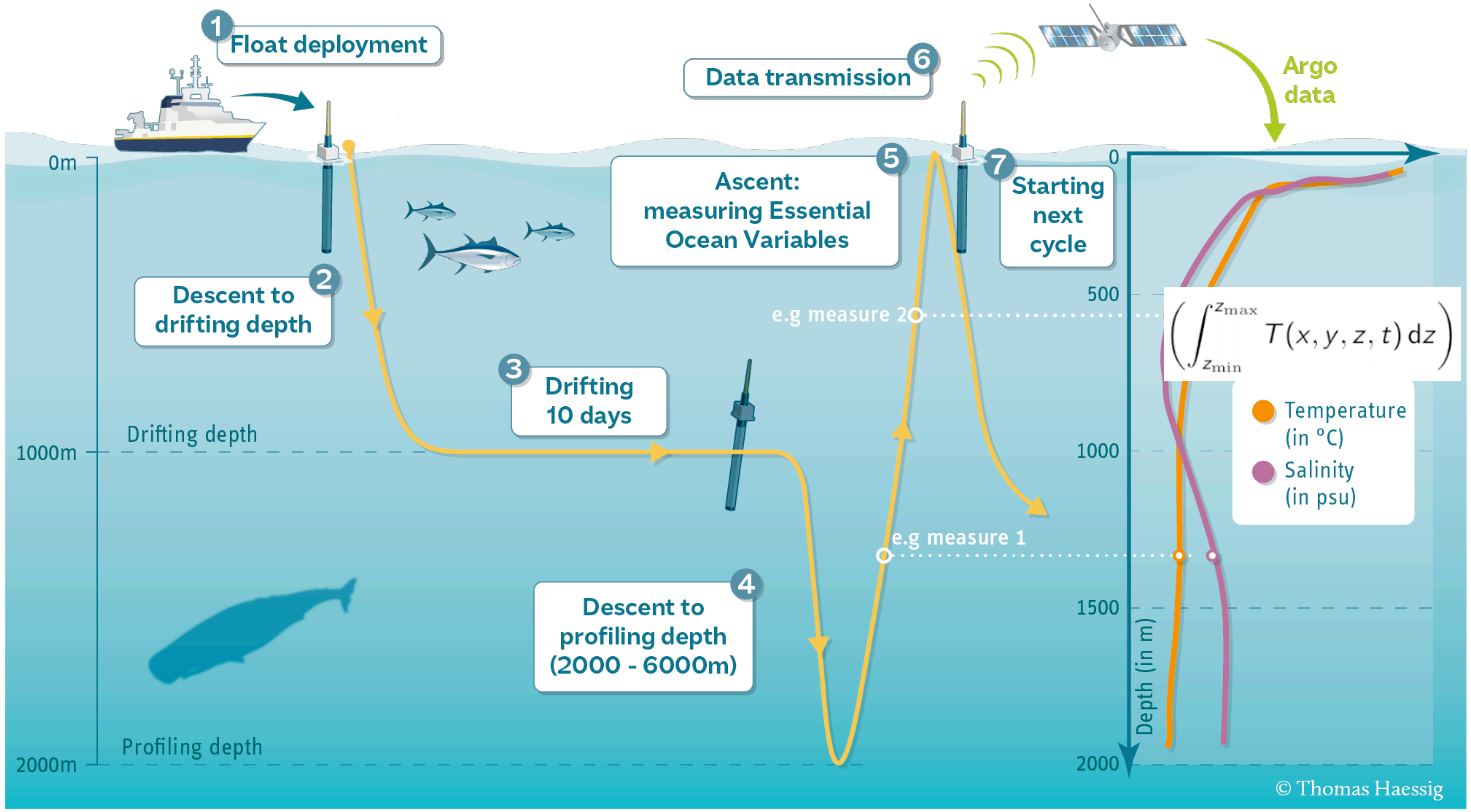


(IPCC AR6)

~50% of sea level rise from 1971-2018 is contributed by **ocean thermal expansion.**

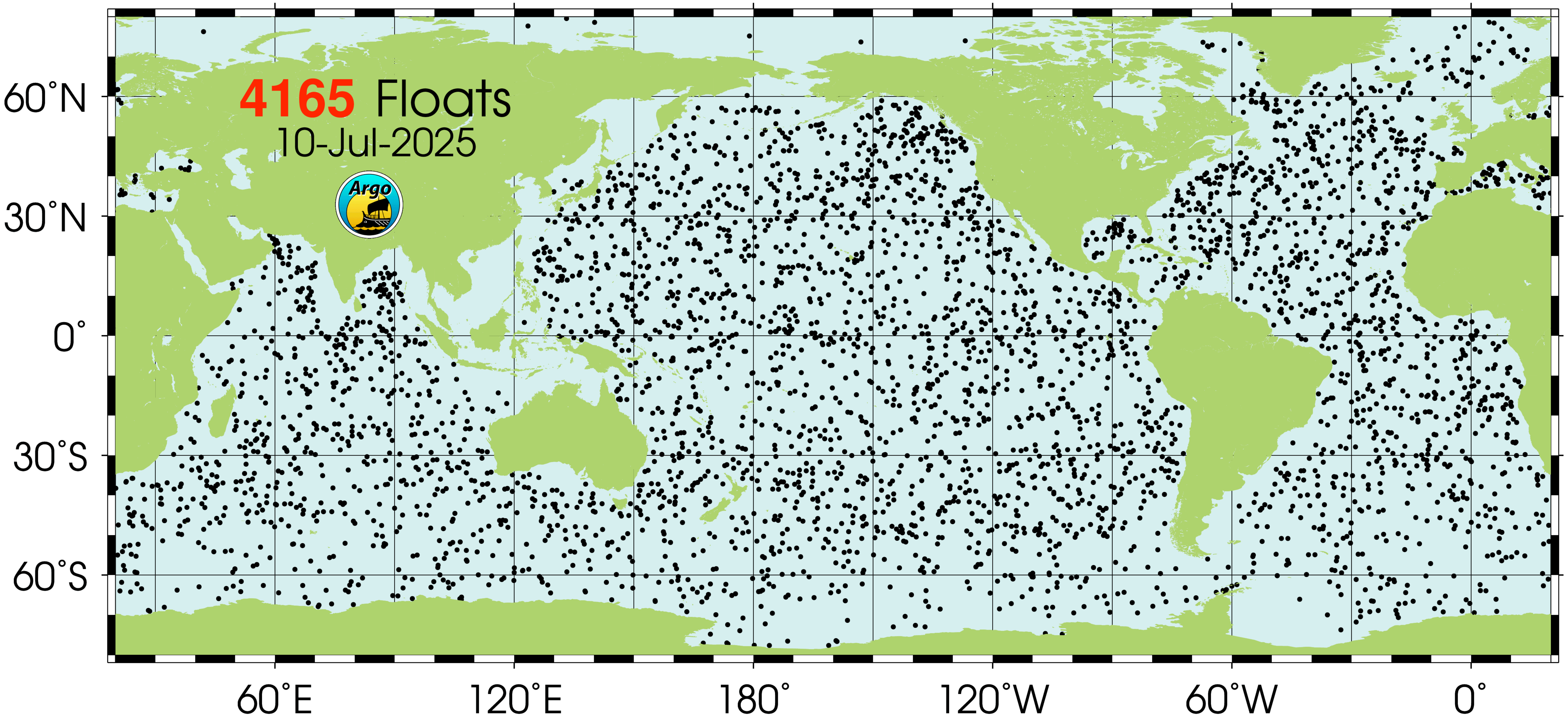
Using Argo data to estimate 15-1850 dbar OHC

$$\mathrm{OHC}(t) = \rho_0 c_{p,0} \int\int \left(\int_{z_{\min}}^{z_{\max}} T(x,y,z,t)\,\mathrm{d}z \right) \mathrm{d}x\,\mathrm{d}y$$



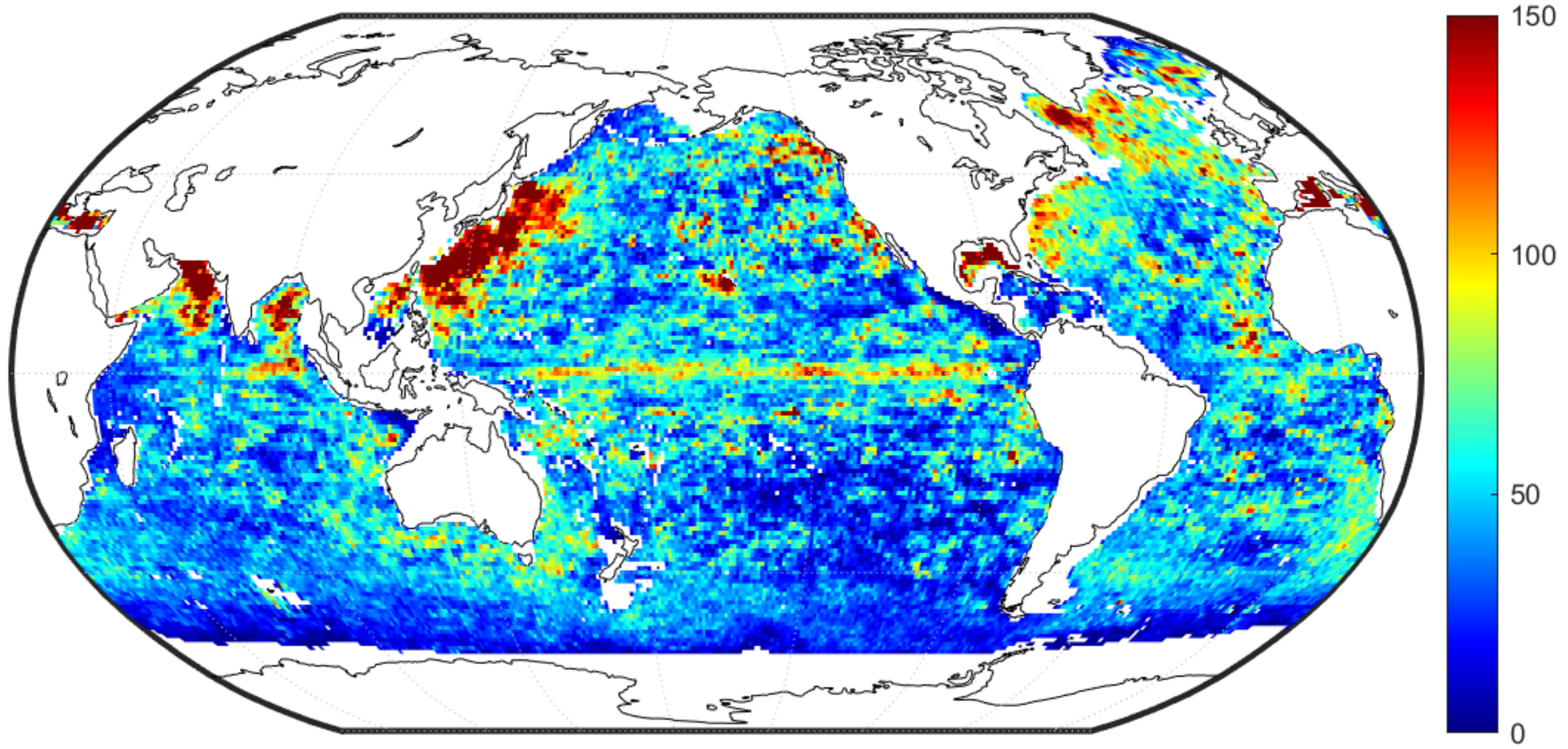
(Argo Program)

Using Argo data to estimate 15-1850 dbar OHC

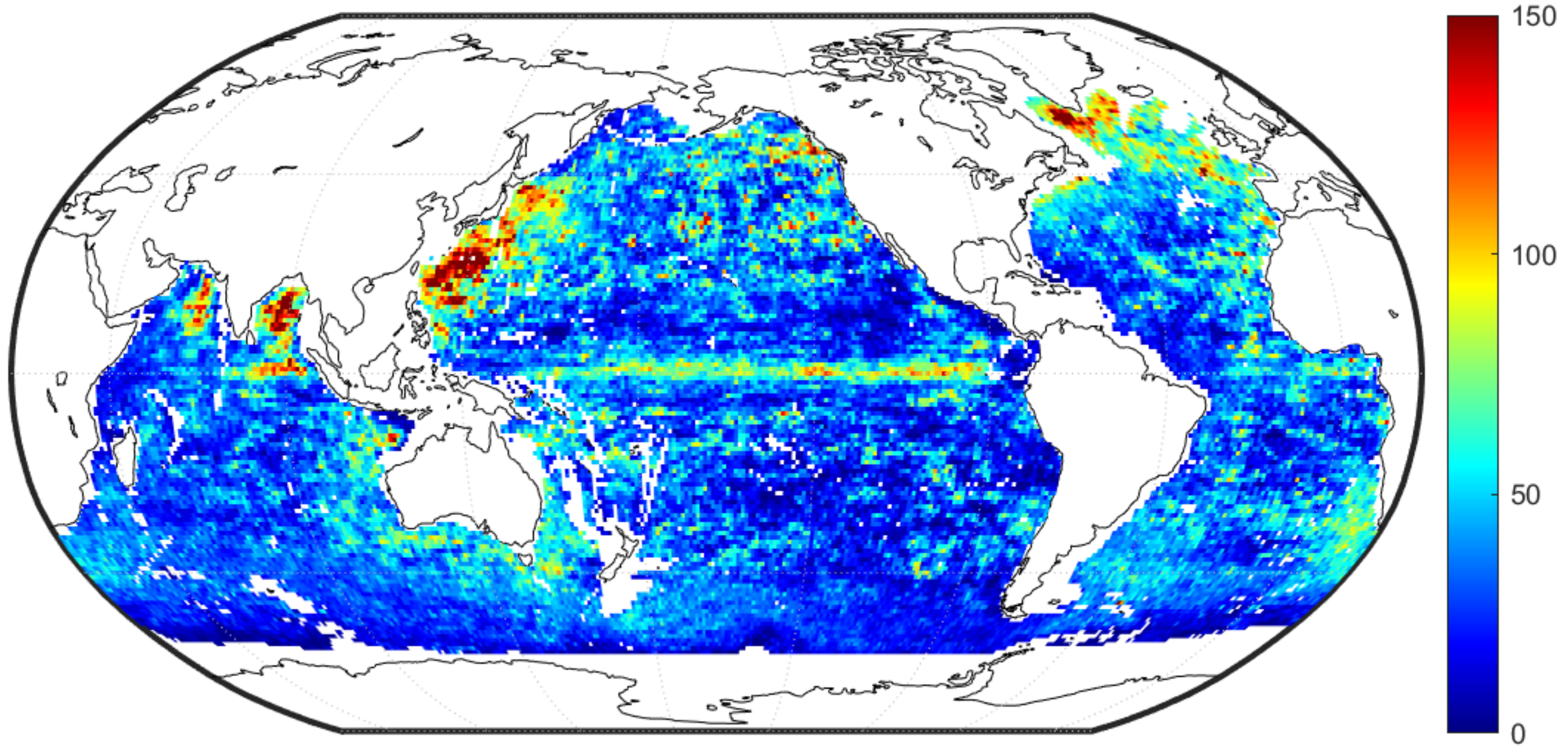


(Argo Program)

Argo data availability (vertical sections)



Top layer profiles (15-975 dbar)



Bottom layer profiles (975-1850 dbar)

Statistical and computational challenges

- **Size:** > 2.5 million Argo profiles (matrix inversion for covariance parameter estimation and kriging is infeasible)
- **Nonstationarity:** Challenging to define a nonstationary covariance function flexible enough to explain variability across entire global ocean
- **Data availability:** How to reliably find uncertainties for global integral of sum of two disjoint sections?

OHC modeling framework

$$\widetilde{\text{OHC}}_{z_u}^{z_d}(x,y,t) = \mu(x,y,t) + a(x,y,t) + \varepsilon(x,y,t),$$

1. Compute the **vertical integral** (consider two layers and integrate Argo profiles in each layer)

OHC modeling framework

$$\widetilde{\mathrm{OHC}}_{z_u}^{z_d}(x, y, t) = \mu(x, y, t) + a(x, y, t) + \varepsilon(x, y, t),$$

1. Compute the **vertical integral** (consider two layers and integrate Argo profiles in each layer)

2. Estimate the **mean field** in the horizontal and temporal dimension

OHC modeling framework

$$\widetilde{\text{OHC}}_{z_u}^{z_d}(x,y,t) = \mu(x,y,t) + a(x,y,t) + \varepsilon(x,y,t),$$

1. Compute the **vertical integral** (consider two layers and integrate Argo profiles in each layer)

2. Estimate the **mean field** in the horizontal and temporal dimension

3. Map the **anomalies**

OHC modeling framework

$$\widetilde{\mathrm{OHC}}_{z_u}^{z_d}(x,y,t) = \mu(x,y,t) + a(x,y,t) + \varepsilon(x,y,t),$$

1. Compute the **vertical integral** (consider two layers and integrate Argo profiles in each layer)

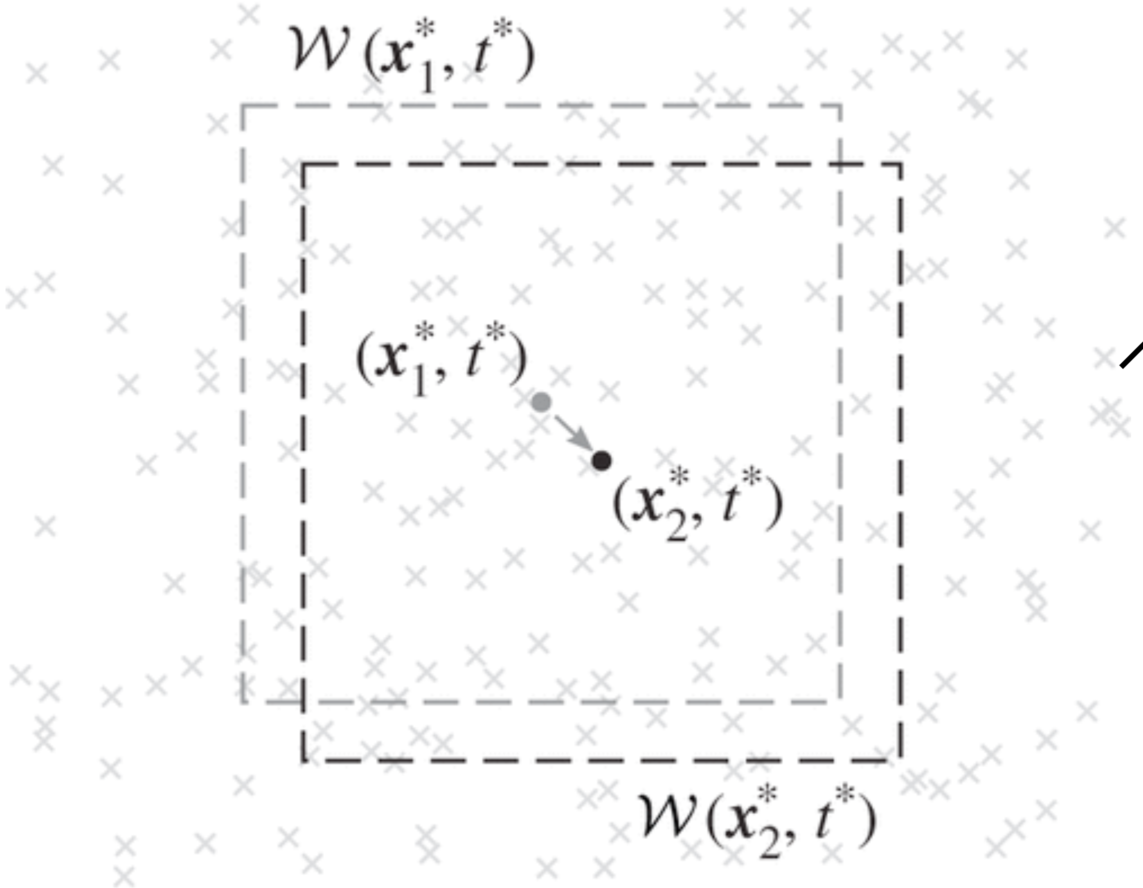
2. Estimate the **mean field** in the horizontal and temporal dimension

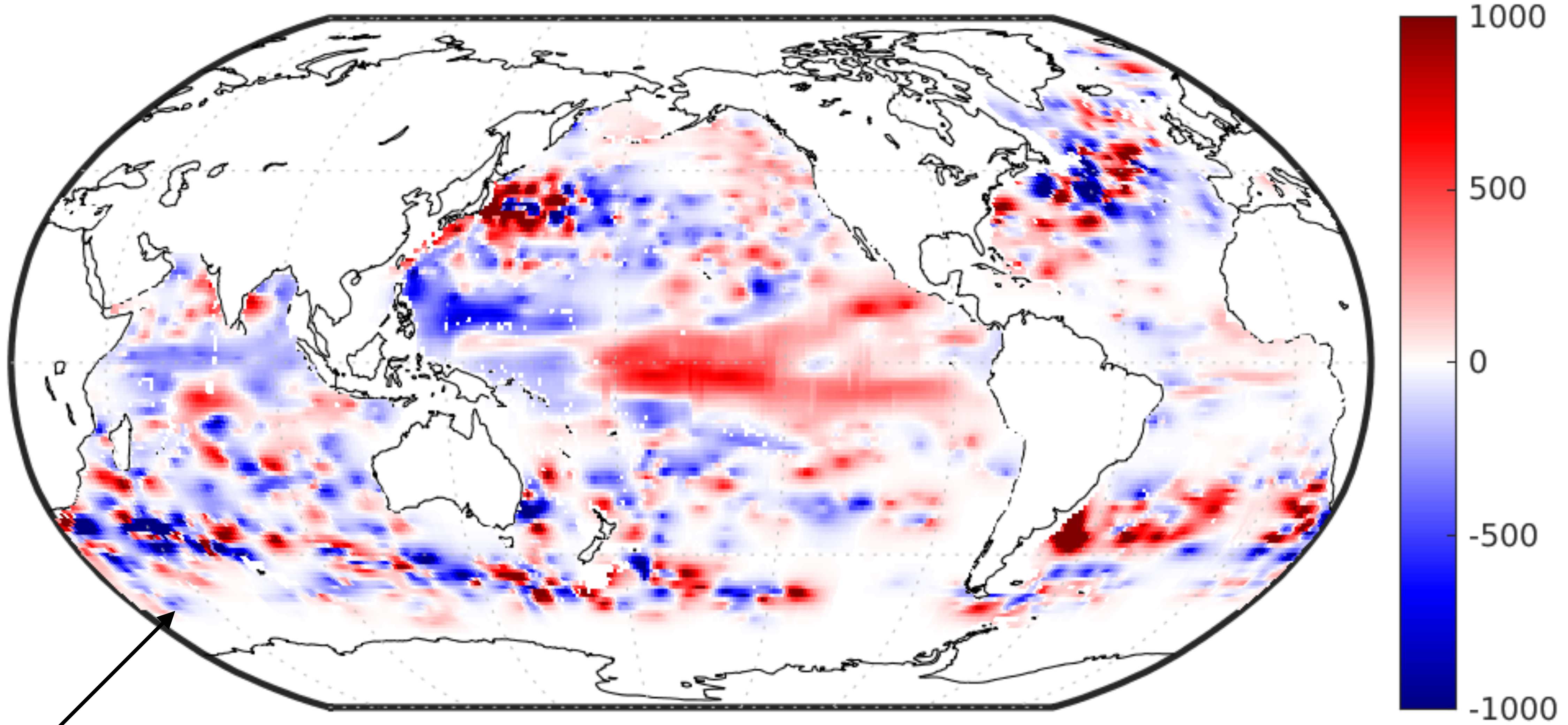
3. Map the **anomalies**
4. Quantify the **uncertainty** of the OHC field

Anomalies: Mapping with local GP regression

1. Model temperature mean field: local least squares regression (Roemmich and Gilson 2009) + linear time trend

2. Model residuals: **locally stationary** Gaussian process (GP) regression (Kuusela and Stein 2018)





Modeled top layer residuals (02/2005)

Anomalies: Mapping with local GP regression

lat/long, date

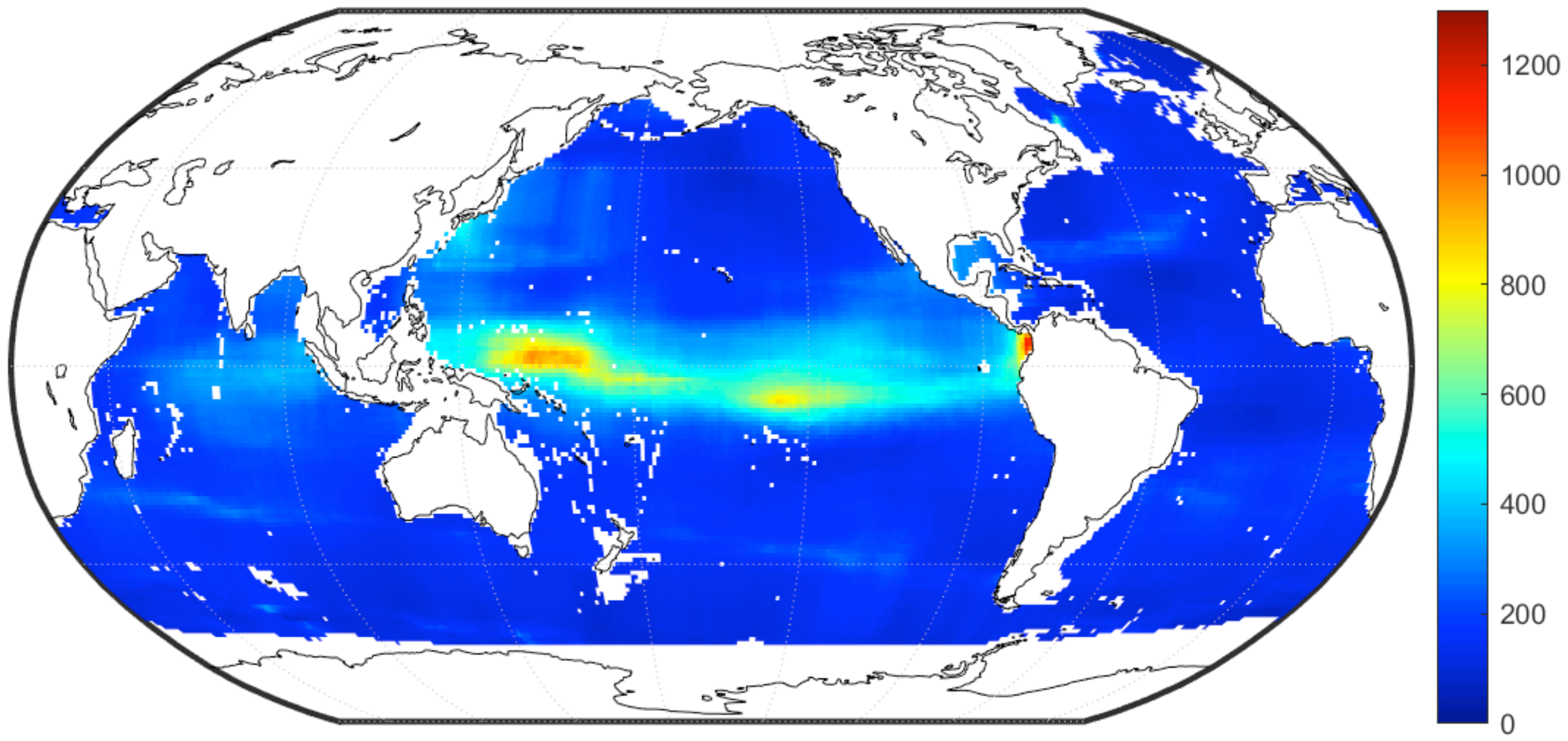
length scales

$$a_i \overset{\text{i.i.d.}}{\sim} \mathrm{GP}(0, k(\boldsymbol{s}_1, t_2, \boldsymbol{s}_2, t_2; \boldsymbol{\theta}_{\mathcal{W}}))$$

$$\epsilon_{i,p} \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2_{\mathcal{W}})$$

$$k = \theta_s \exp(-d(\boldsymbol{s}_1, t_1, \boldsymbol{s}_2, t_2))$$

$$d(\boldsymbol{s}_1, t_1, \boldsymbol{s}_2, t_2) = \sqrt{(\frac{x_1 - x_2}{\theta_{\mathcal{W},\mathrm{lon}}})^2 + (\frac{y_1 - y_2}{\theta_{\mathcal{W},\mathrm{lat}}})^2 + (\frac{t_1 - t_2}{\theta_{\mathcal{W},t}})^2}$$



Length scale MLE (latitude; top layer)

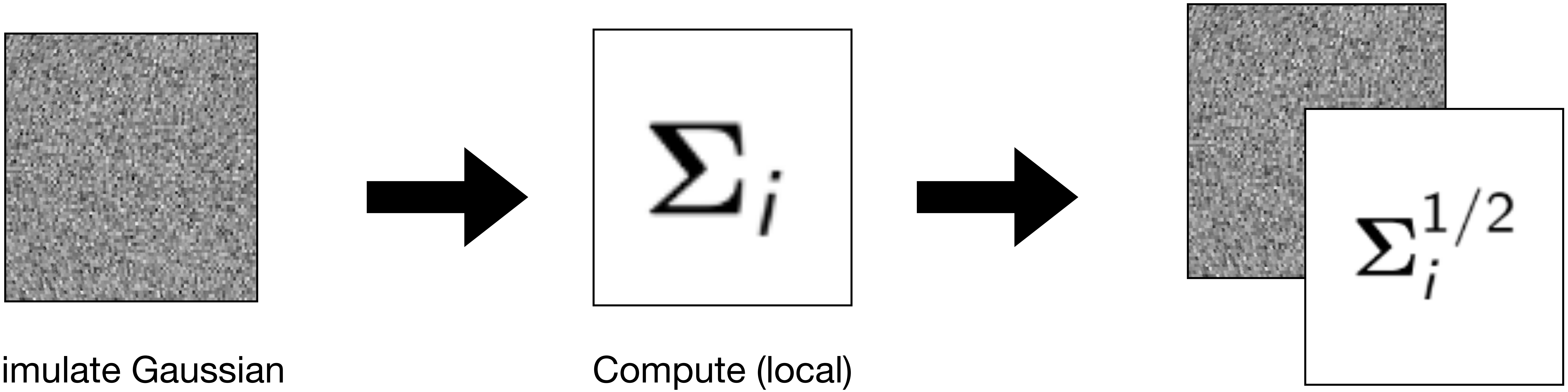
Uncertainties: local conditional simulation

OHC | data - **multivariate normal** with conditional covariance Σ_i

(parameterized by estimated GP variance, length scales)

Local conditional simulations! (extension of Nychka et.al. 2018)

Uncertainties: local conditional simulation

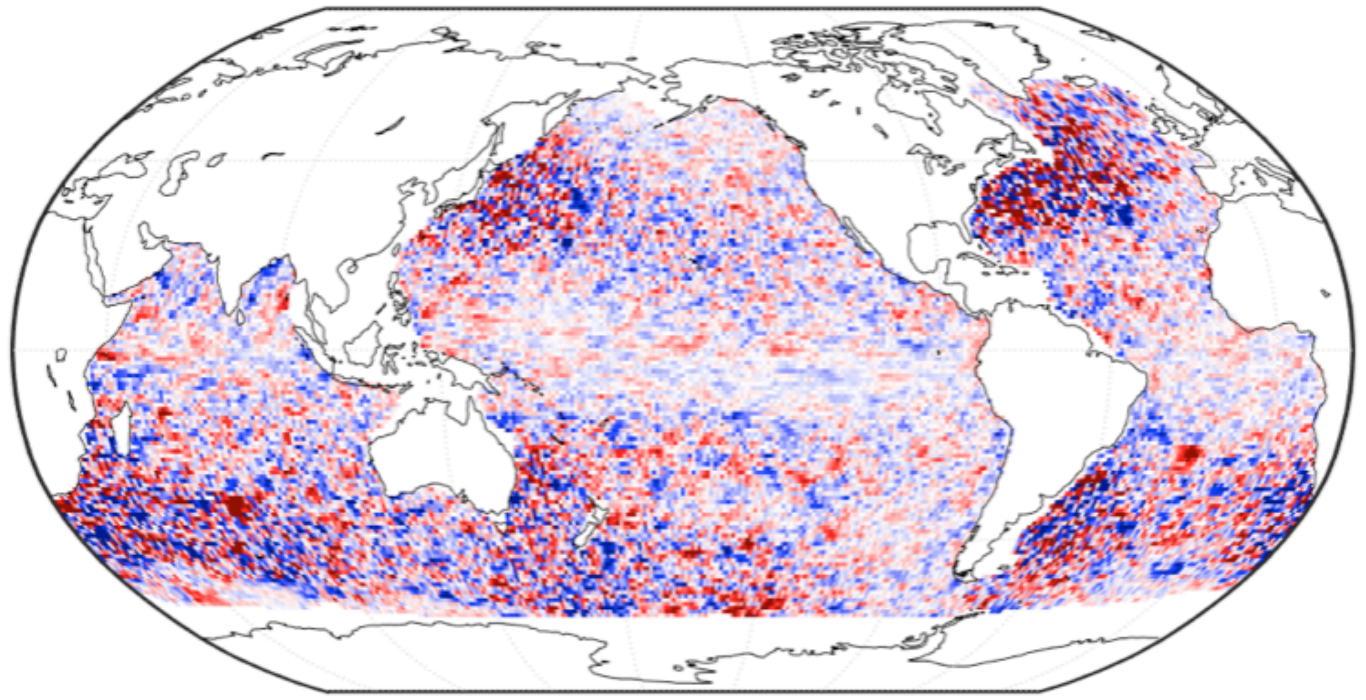


Simulate Gaussian white noise over grid (keep fixed)

Compute (local) conditional covariance matrix

Take symmetric square root and multiply by white noise

Uncertainties: local conditional simulation



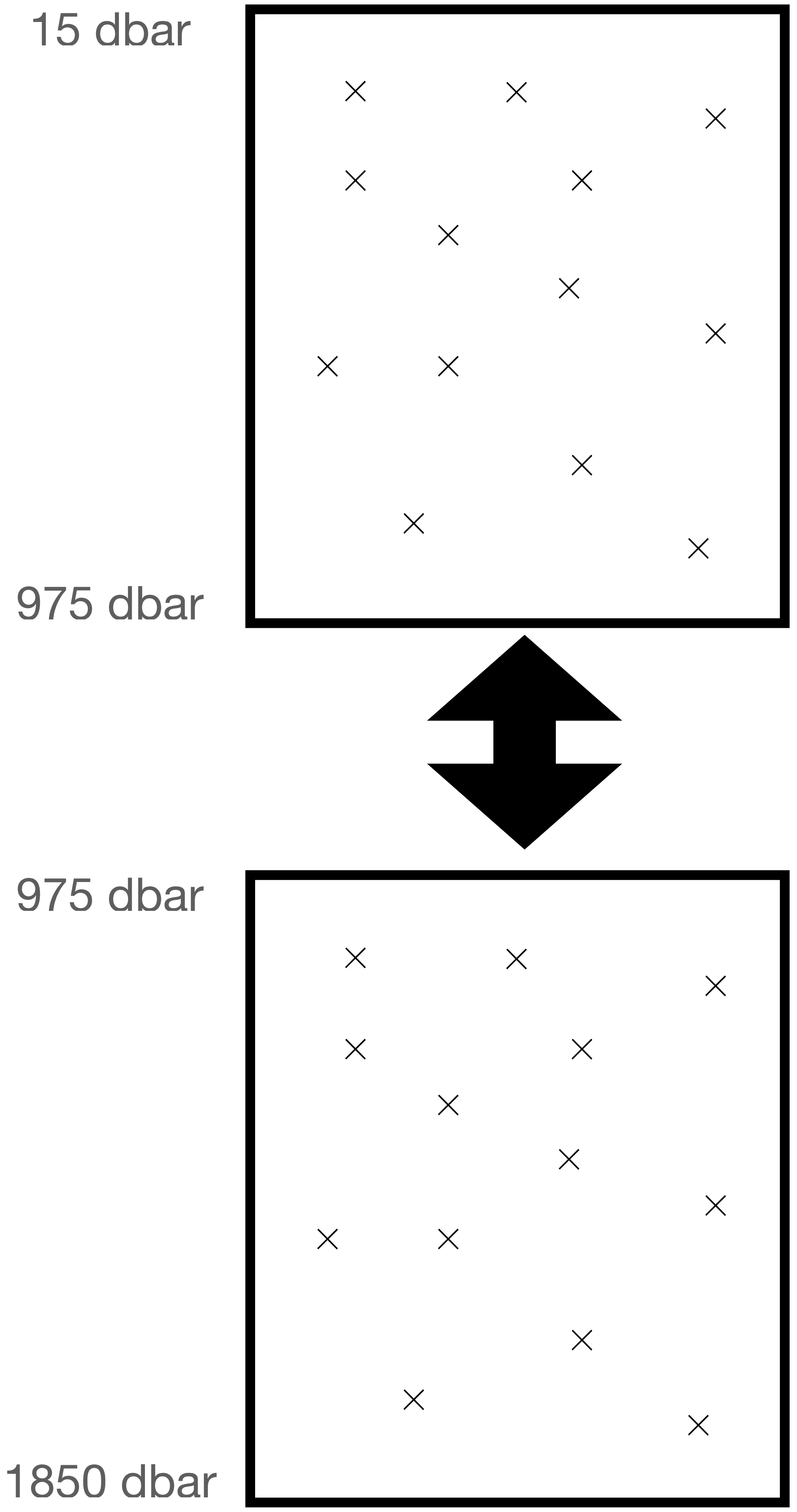
Keep the center point and repeat for all grid points

Repeat for desired number of ensemble members and other layer

Square sum of sample standard deviations

$$(\sqrt{\mathrm{Var}(\mathrm{OHC}_{\mathrm{top}}|\mathrm{data})} + \sqrt{\mathrm{Var}(\mathrm{OHC}_{\mathrm{bot}}|\mathrm{data})})^2$$

Improving total OHC uncertainty estimates

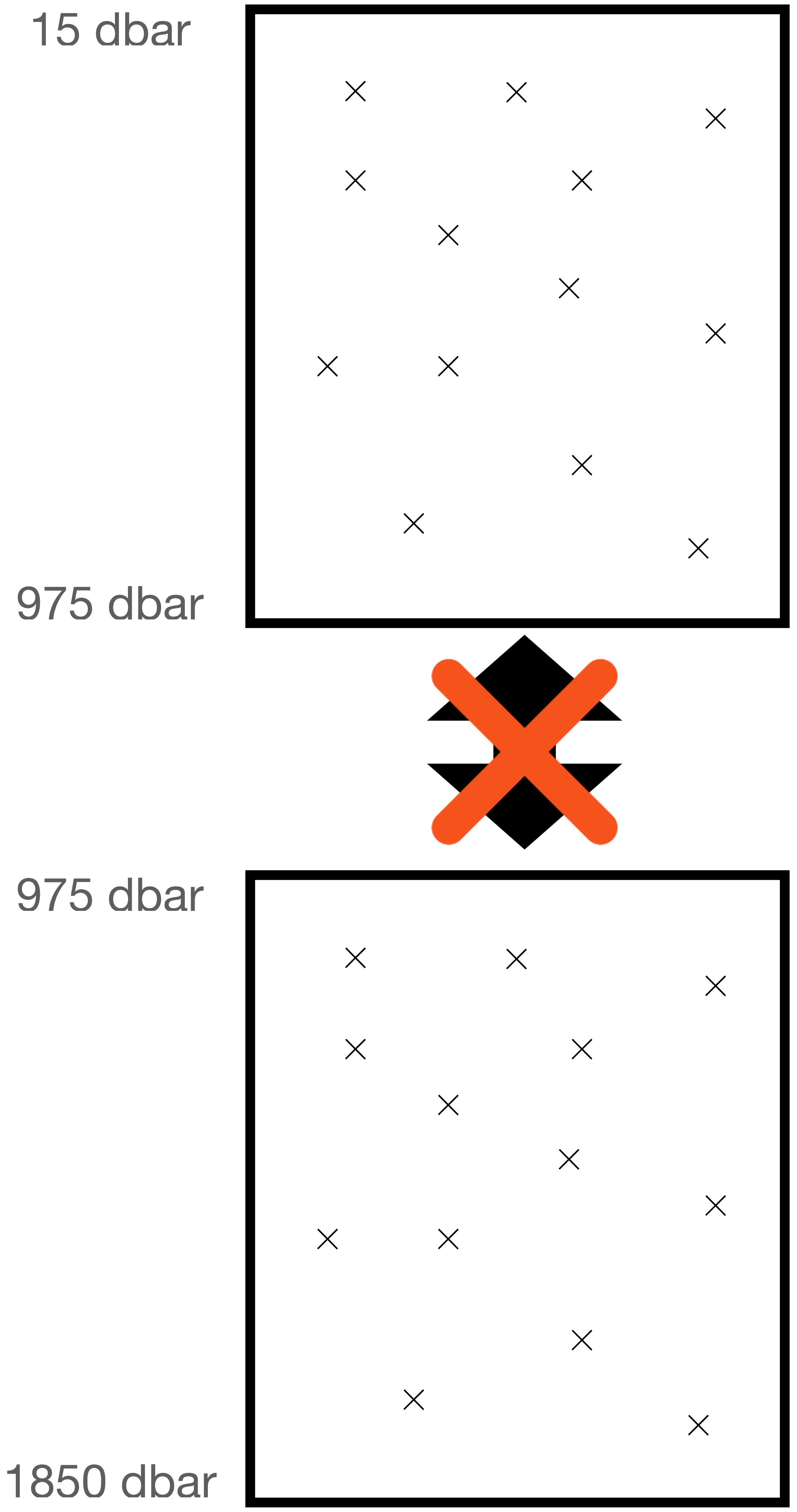


$$\mathrm{Var}(\mathrm{OHC}_{\mathrm{top}}|\mathrm{data})$$

$$2\mathrm{Cov}(\mathrm{OHC}_{\mathrm{top}}, \mathrm{OHC}_{\mathrm{bot}}|\mathrm{data}) \longrightarrow \mathrm{Var}(\mathrm{OHC}_{\mathrm{total}}|\mathrm{data})$$

$$\mathrm{Var}(\mathrm{OHC}_{\mathrm{bot}}|\mathrm{data})$$

Improving total OHC uncertainty estimates



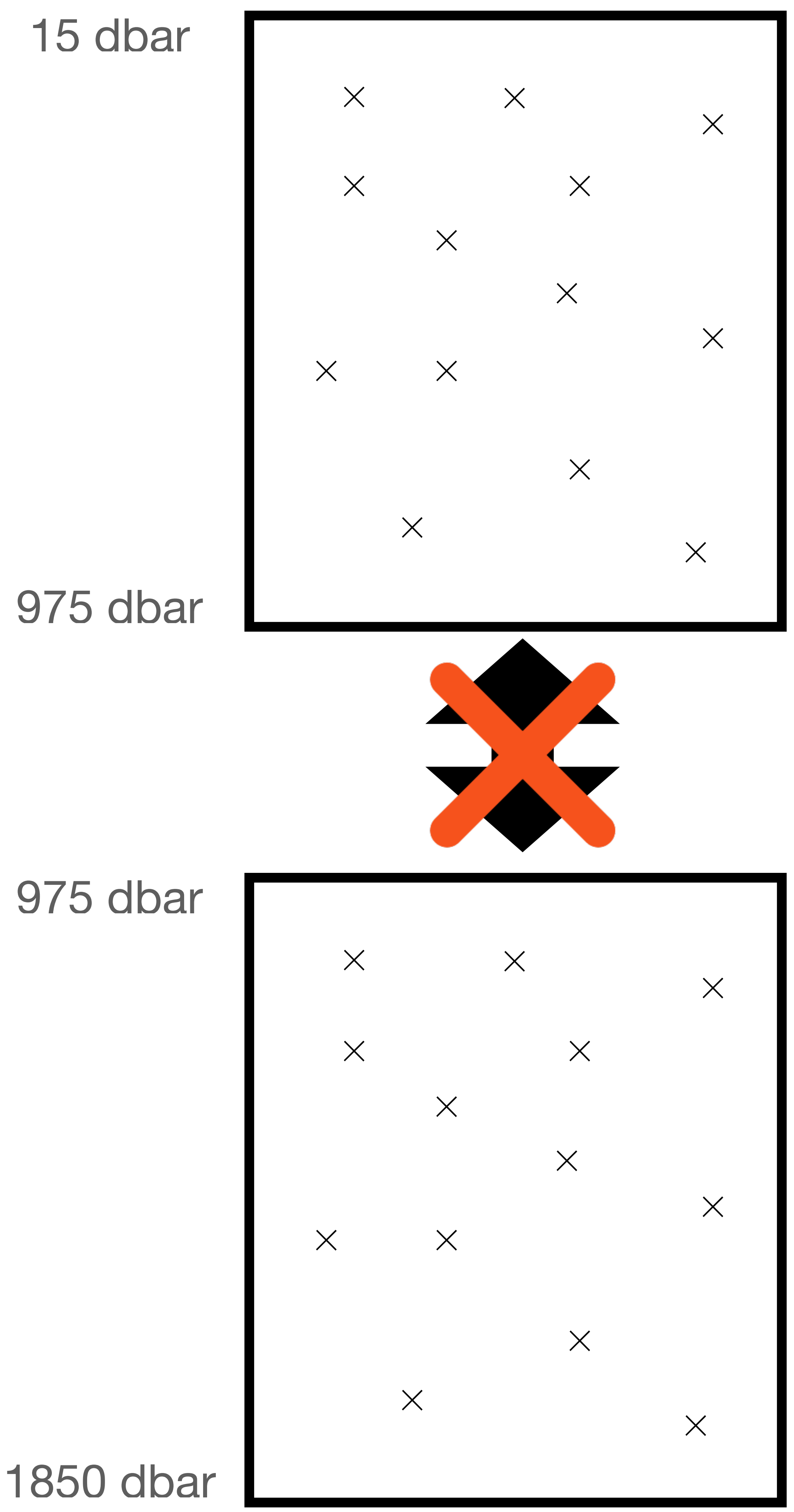
$$\mathrm{Var}(\mathrm{OHC_{top}}|\mathrm{data})$$

$$2\mathrm{Cov}(\mathrm{OHC_{top}}, \mathrm{OHC_{bot}}|\mathrm{data})$$

$$\mathrm{Var}(\mathrm{OHC_{bot}}|\mathrm{data})$$

$$\mathrm{Var}(\mathrm{OHC_{total}}|\mathrm{data})$$

Improving total OHC uncertainty estimates



$$\mathrm{Var}(\mathrm{OHC}_{\mathrm{top}}|\mathrm{data})$$

$$2\mathrm{Cov}(\mathrm{OHC}_{\mathrm{top}}, \mathrm{OHC}_{\mathrm{bot}}|\mathrm{data})$$

$$\mathrm{Var}(\mathrm{OHC}_{\mathrm{bot}}|\mathrm{data})$$

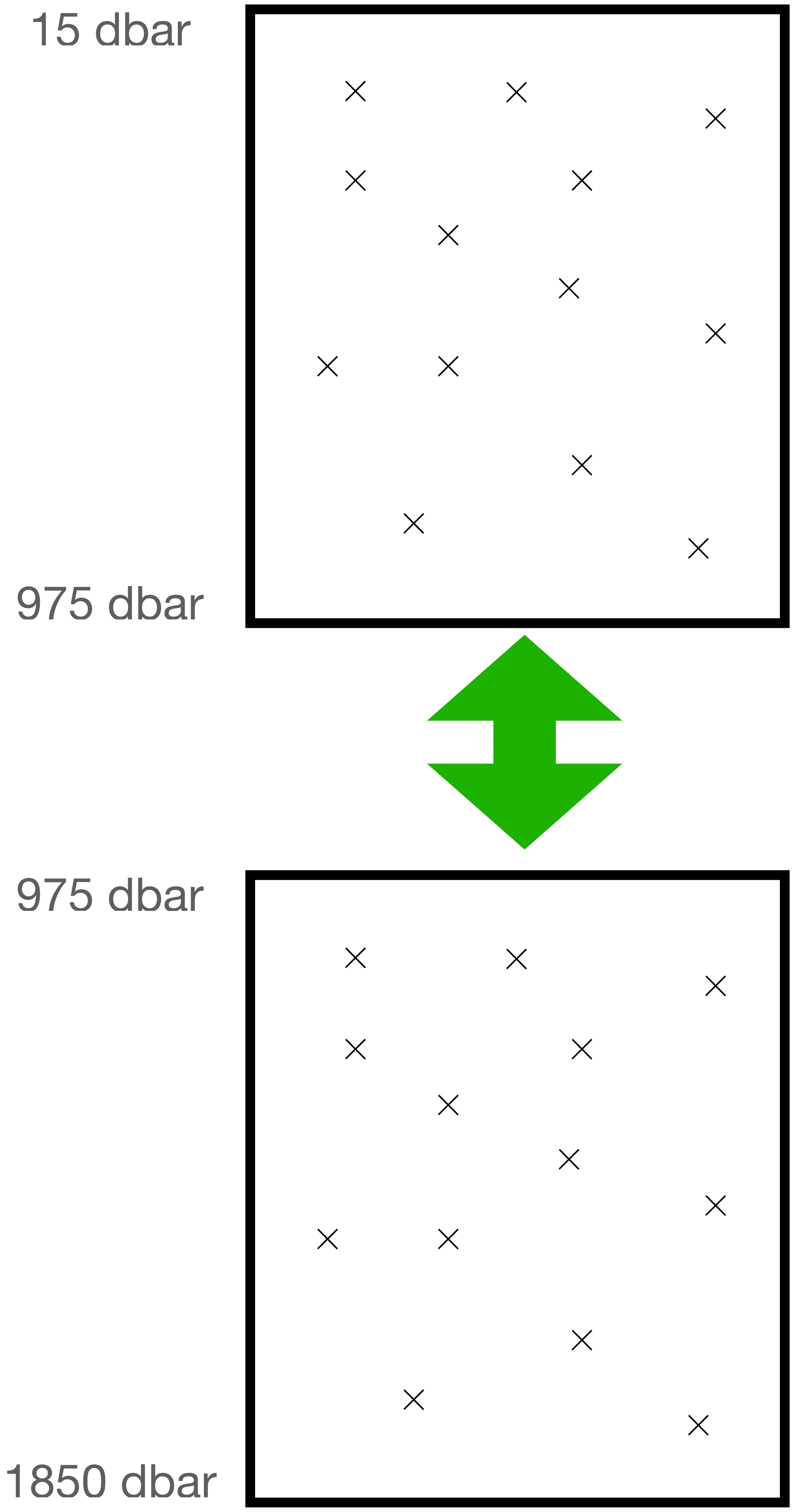
$$\mathrm{Var}(\mathrm{OHC}_{\mathrm{total}}|\mathrm{data})$$

(conservative upper bound)

$$\left(\sqrt{\mathrm{Var}(\mathrm{OHC}_{\mathrm{top}}|\mathrm{data})} + \sqrt{\mathrm{Var}(\mathrm{OHC}_{\mathrm{bot}}|\mathrm{data})}\right)^2$$

Squaring the sum of the standard deviations for each layer **overestimates** the uncertainties of the total OHC.

Improving total OHC uncertainty estimates



$$\mathrm{Var}(\mathrm{OHC}_{\mathrm{top}}|\mathrm{data})$$

$$2\mathrm{Cov}(\mathrm{OHC}_{\mathrm{top}}, \mathrm{OHC}_{\mathrm{bot}}|\mathrm{data})$$

$$\mathrm{Var}(\mathrm{OHC}_{\mathrm{bot}}|\mathrm{data})$$

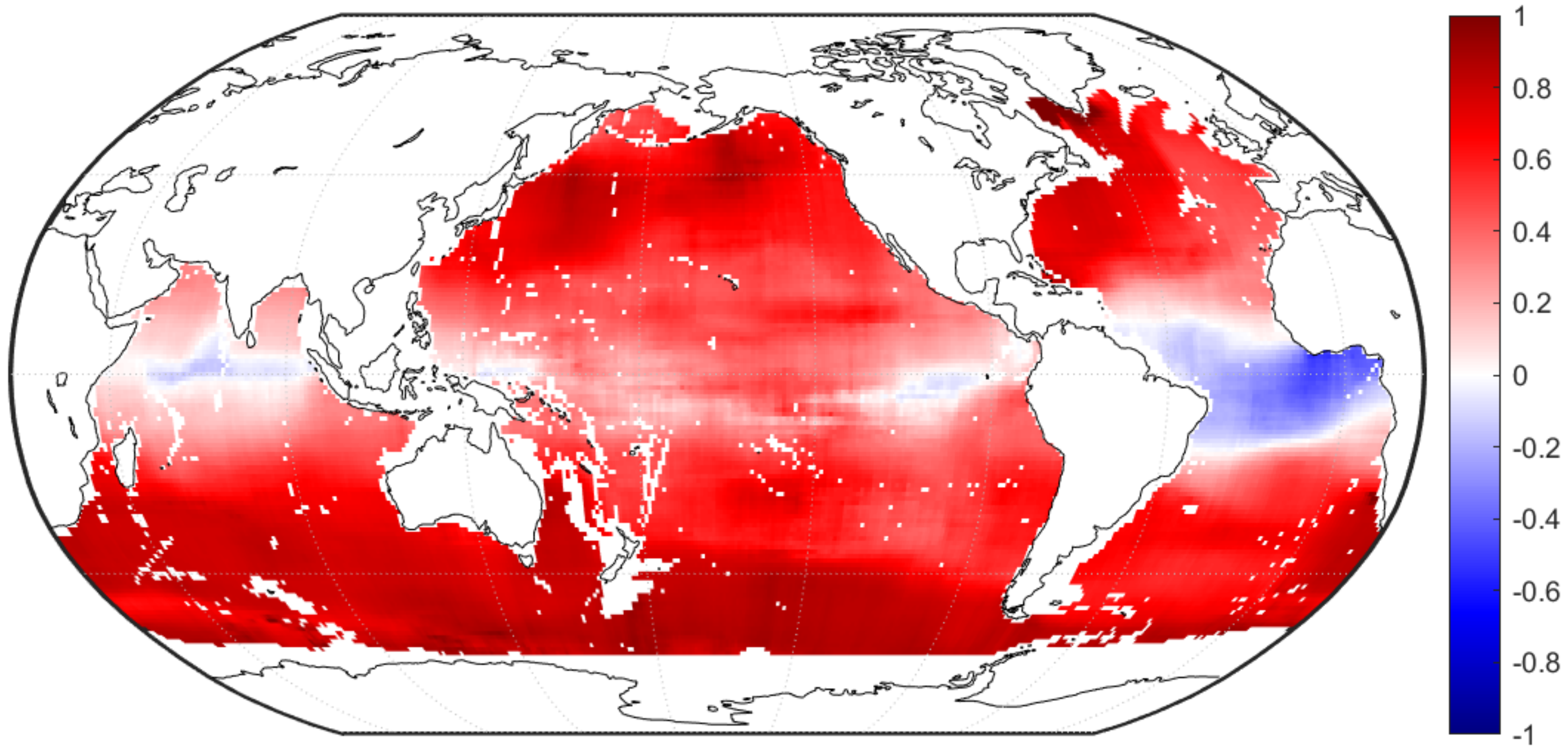
$$\mathrm{Var}(\mathrm{OHC}_{\mathrm{total}}|\mathrm{data})$$

We can improve the uncertainty estimates by also modeling the **correlation.**

Bivariate extension: local GP model

$$\begin{bmatrix} a_{\text{up}} \\ a_{\text{mid}} \end{bmatrix}_i \overset{\text{i.i.d.}}{\sim} \text{GP}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \boldsymbol{K}(\boldsymbol{s}_1, \boldsymbol{t}_1, \boldsymbol{s}_2, \boldsymbol{t}_2; \boldsymbol{\theta}_{\mathcal{W}})\right)$$

$$\begin{bmatrix} \varepsilon_{\text{up}} \\ \varepsilon_{\text{mid}} \end{bmatrix}_i \overset{\text{i.i.d.}}{\sim} \text{MVN}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}\right)$$

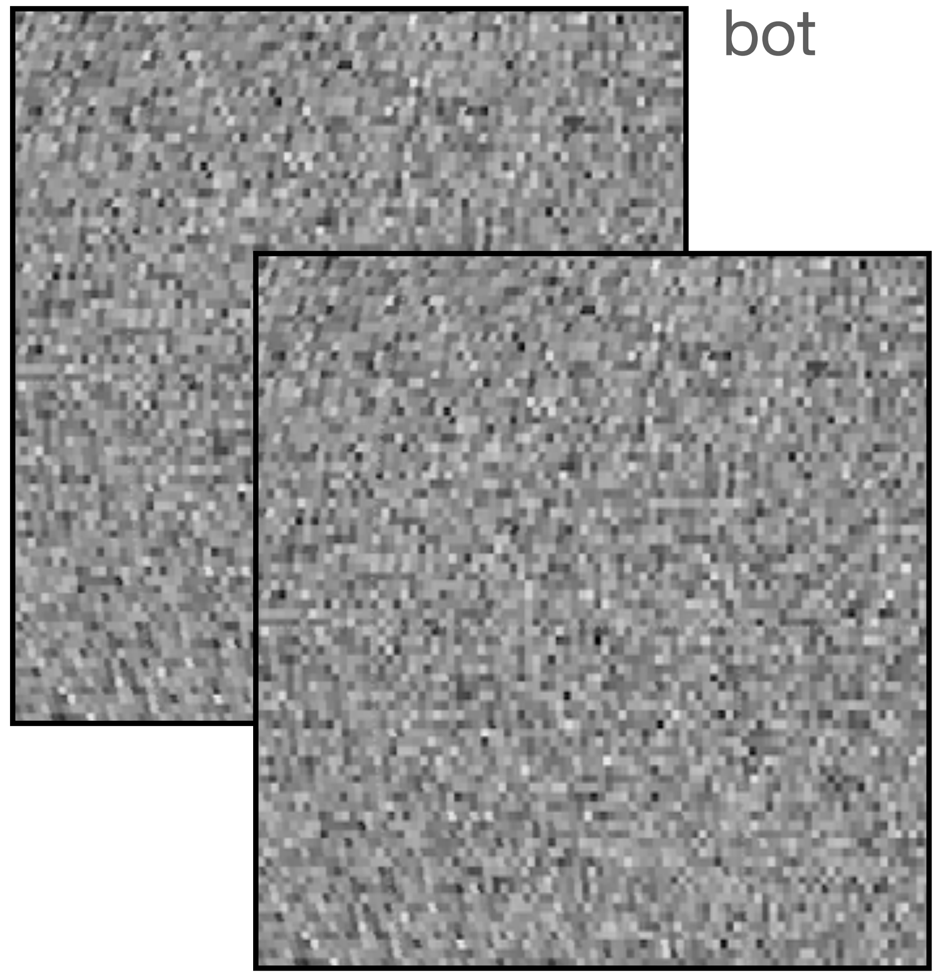


Estimated cross-correlation

$$\boldsymbol{K}_{12}(\boldsymbol{z_1}, \boldsymbol{z_2}; \boldsymbol{\theta}_{\mathcal{W}}) = \beta \frac{\delta_1\delta_2}{|\boldsymbol{\Theta}_{12}|^{1/2}} \exp\left(-\sqrt{(\boldsymbol{z_1} - \boldsymbol{z_2})^{\mathrm{T}} \boldsymbol{\Theta}_{12}^{-1} (\boldsymbol{z_1} - \boldsymbol{z_2})}\right)$$

Cross-covariance
(Kleiber and Nychka 2012)

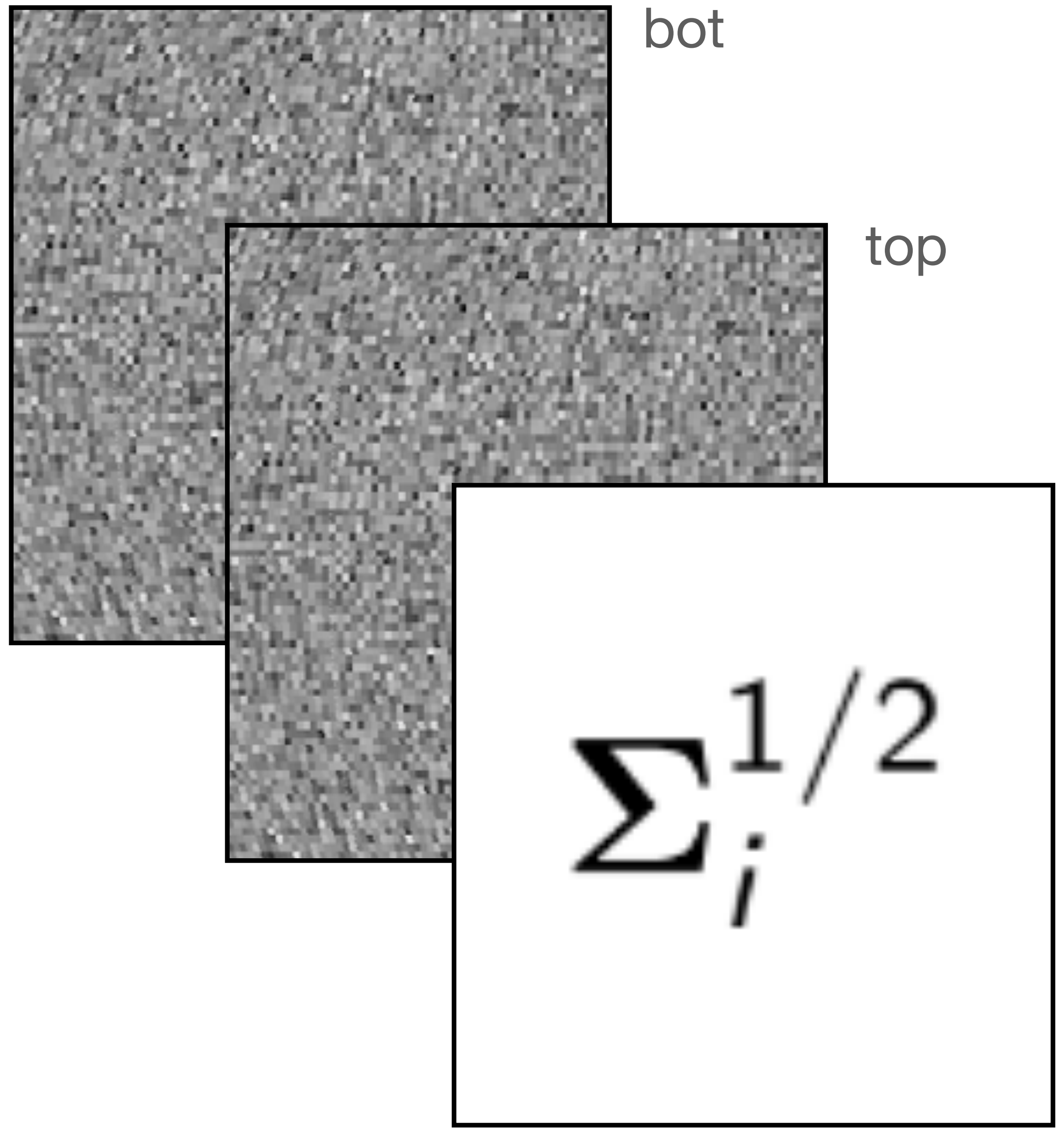
Bivariate extension: local conditional simulation



Simulate Gaussian white noise over grid (keep fixed)

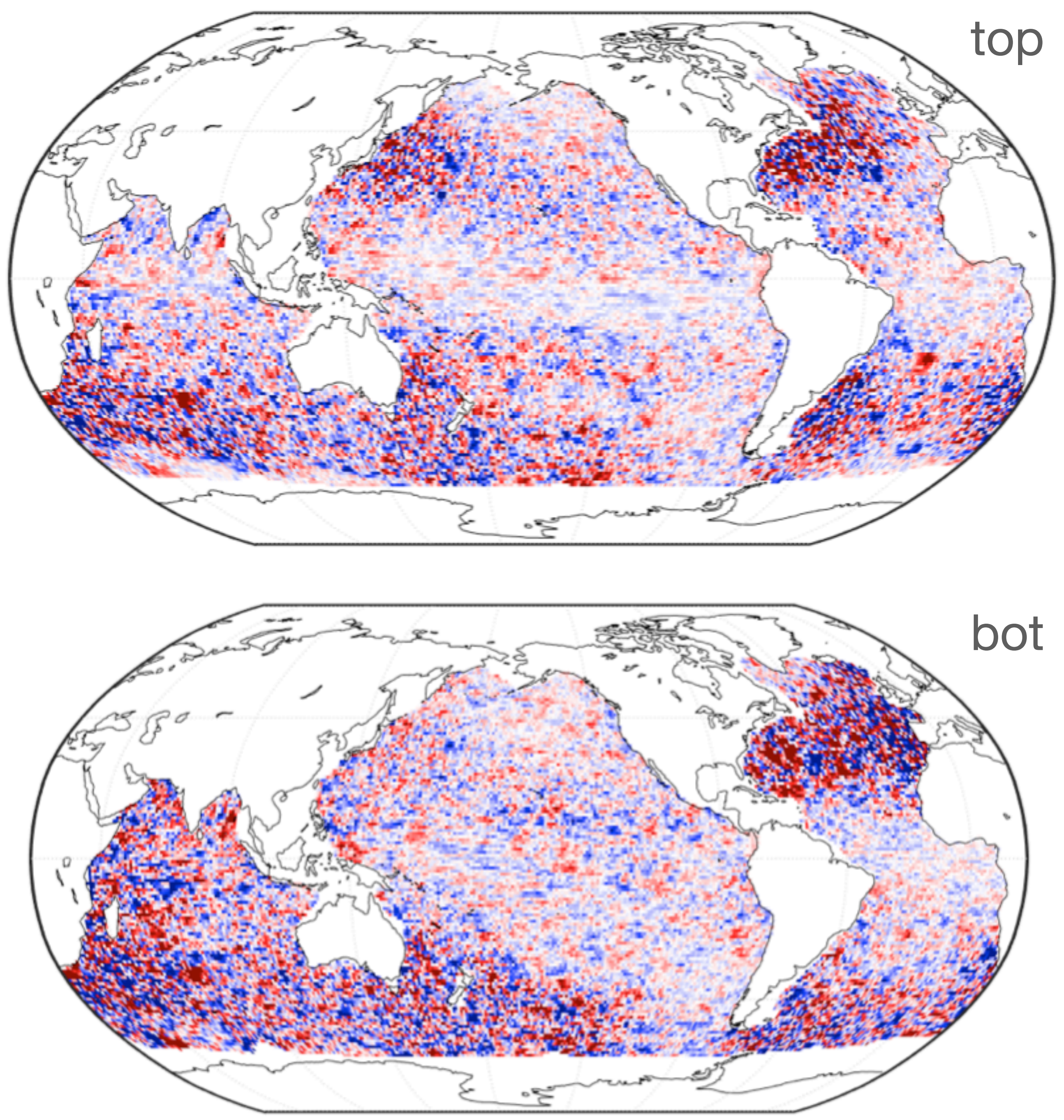
$$\Sigma_i$$

Compute (local) conditional covariance matrix now with cross-correlation included



Take symmetric square root and multiply by white noise

Bivariate extension: local conditional simulation



Keep the center point and repeat for all grid points

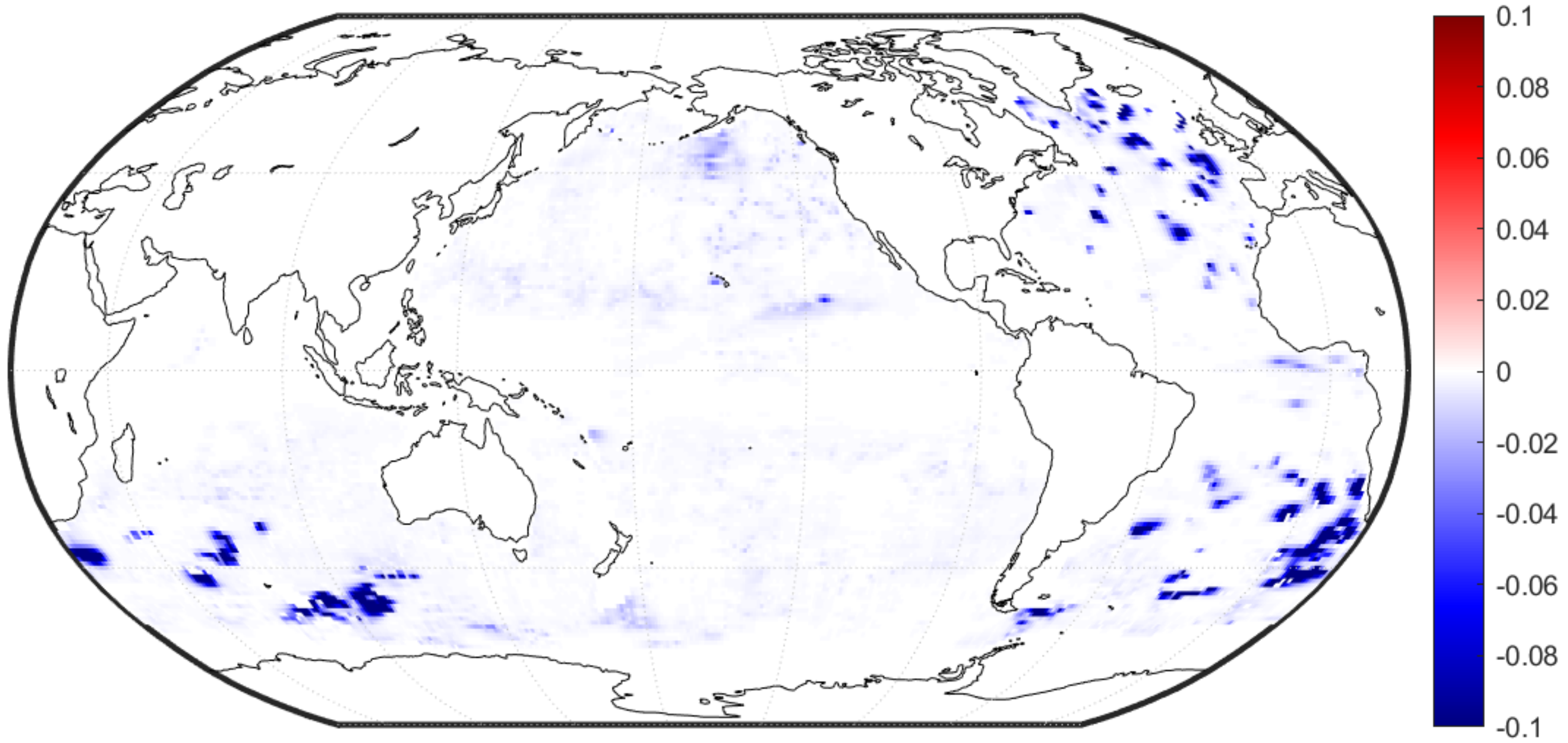
Repeat for desired number of ensemble members

Sample variance of top + bottom layer integrated ensemble members is estimate of

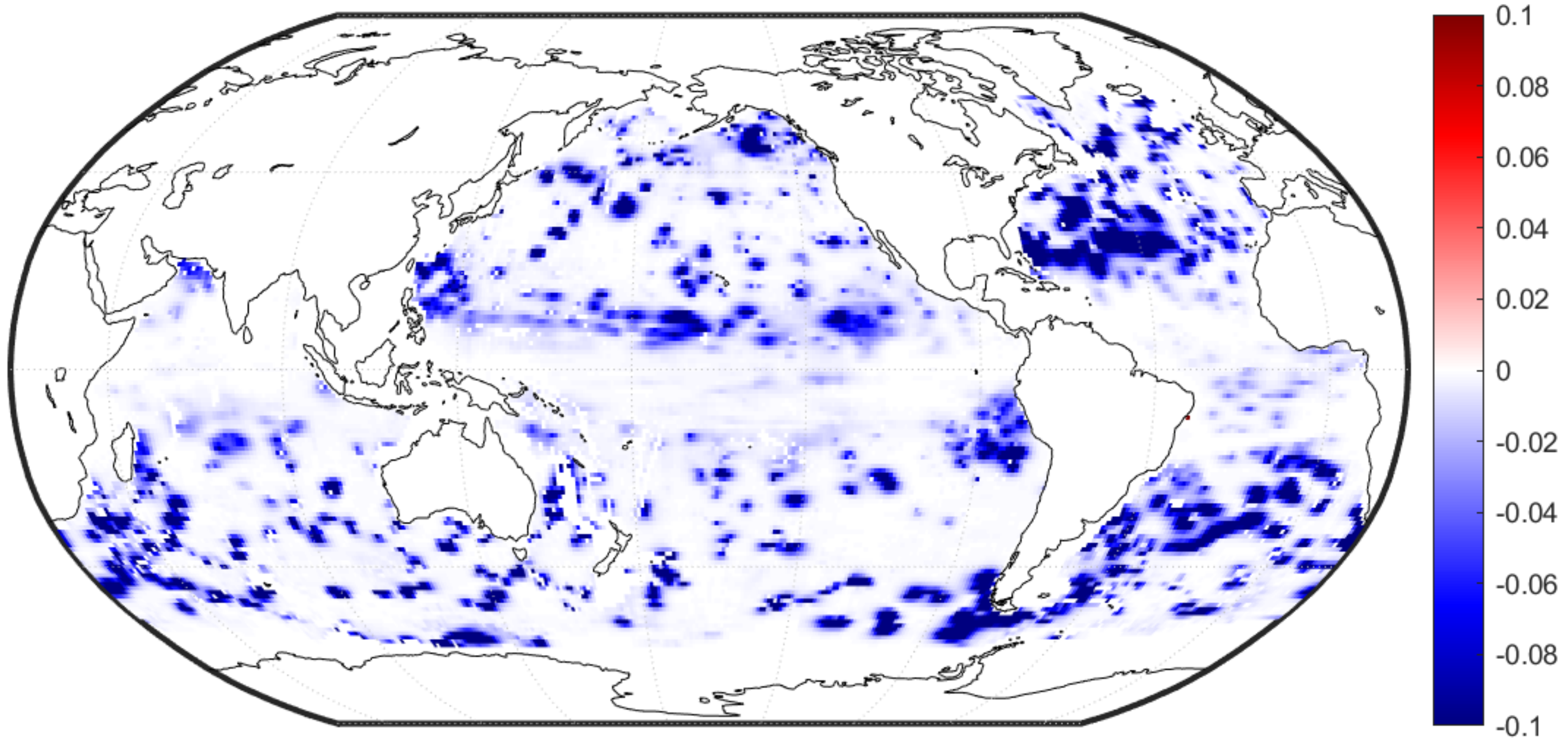
$$\mathrm{Var}(\mathrm{OHC}_{\mathrm{total}} | \mathrm{data})$$

Bivariate extension: predictive variance comparison

$$\frac{\text{bivariate kriging variance - univariate kriging variance}}{\text{univariate kriging variance}}$$ (02/2010)

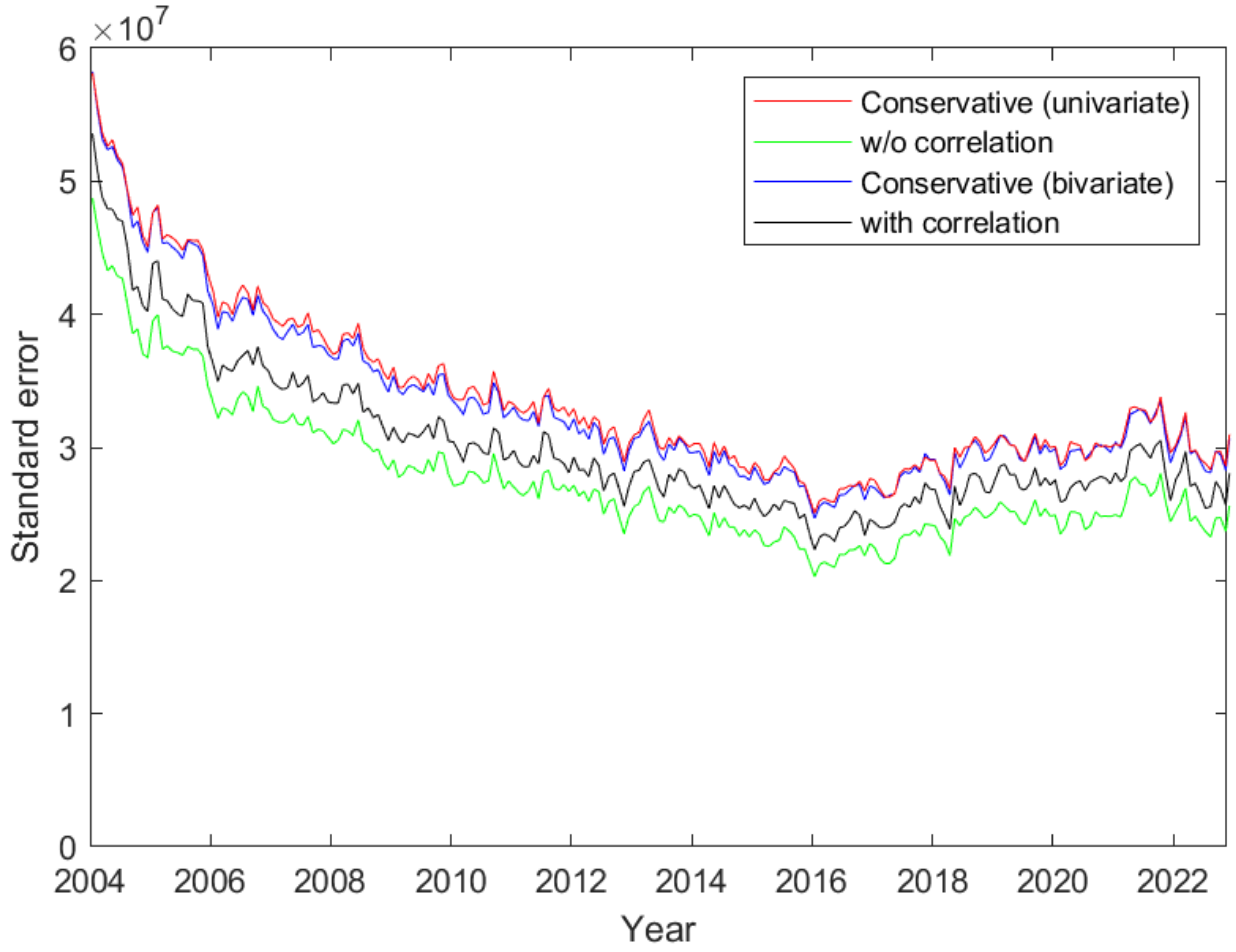


Top layer



Bottom layer

Bivariate extension: global OHC uncertainty comparison



(w/o correlation)

$$\sqrt{\mathrm{Var}(\mathrm{OHC}_{\mathrm{top}}|\mathrm{data}) + \mathrm{Var}(\mathrm{OHC}_{\mathrm{bot}}|\mathrm{data})}$$

(conservative)

$$\sqrt{\mathrm{Var}(\mathrm{OHC}_{\mathrm{top}}|\mathrm{data})} + \sqrt{\mathrm{Var}(\mathrm{OHC}_{\mathrm{bot}}|\mathrm{data})}$$

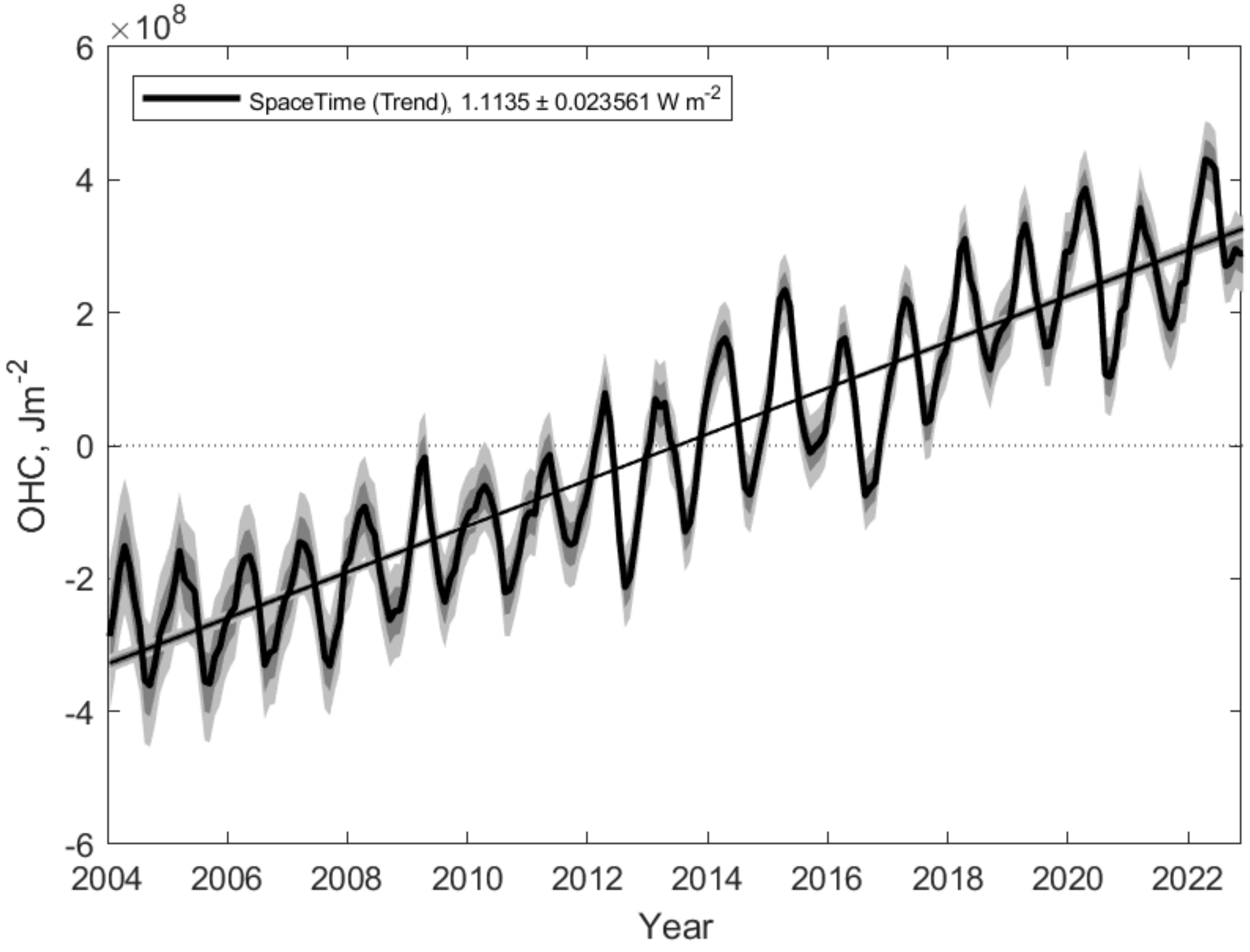
(with correlation)

$$\sqrt{\mathrm{Var}(\mathrm{OHC}_{\mathrm{total}}|\mathrm{data})}$$

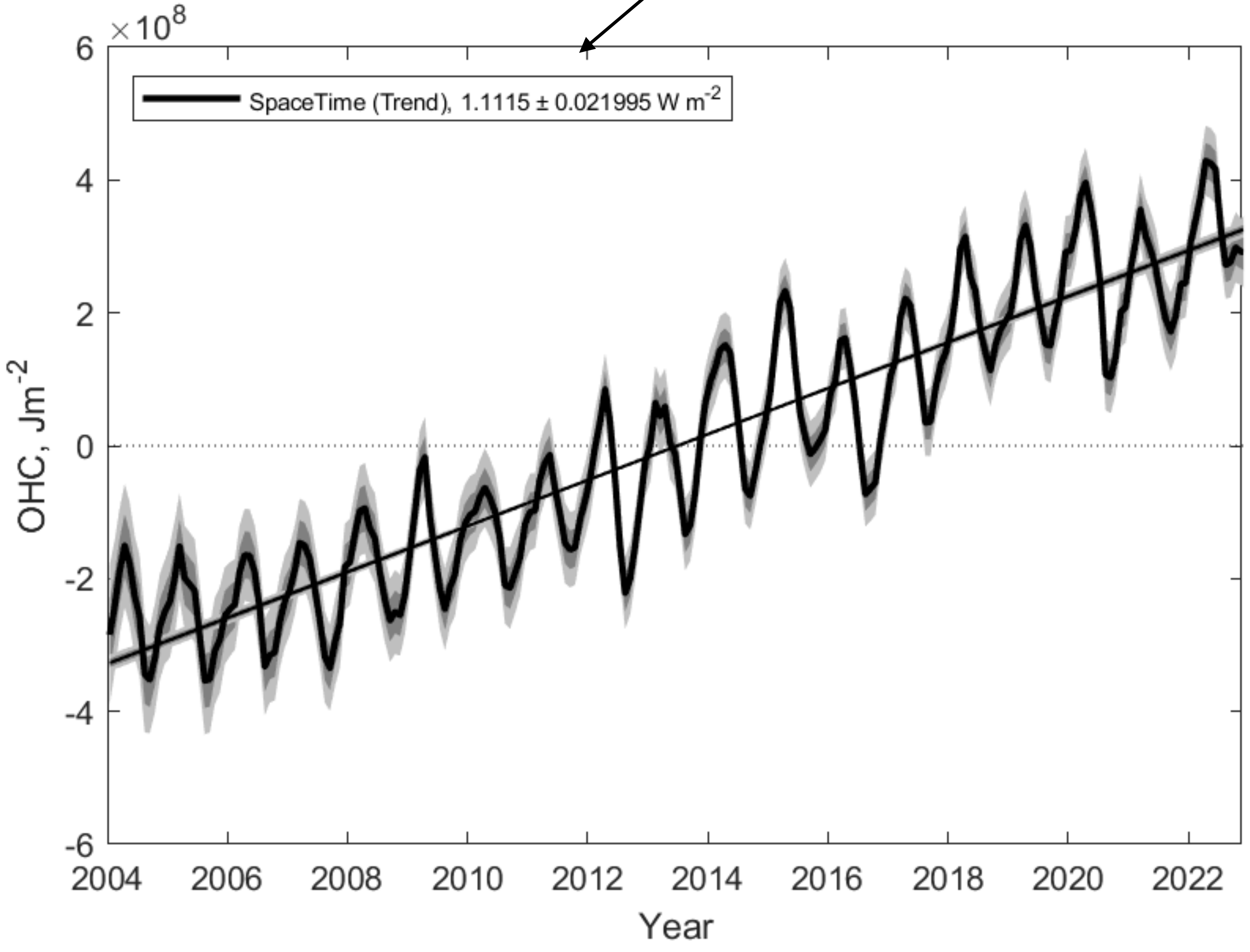
When we model the correlation, the uncertainties are **smaller** than the conservative and **larger** than those w/o the correlation.

Bivariate extension: global OHC uncertainty comparison

~13% smaller



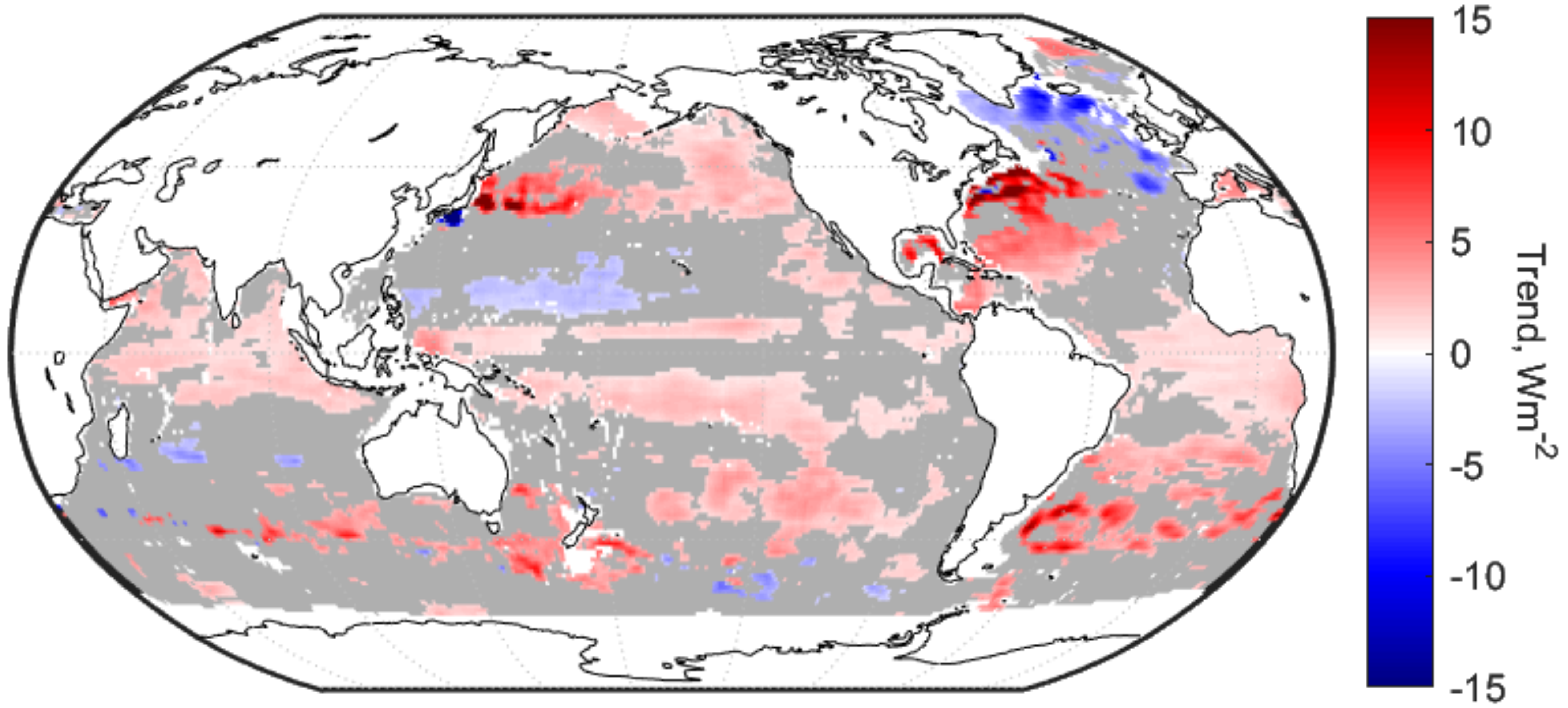
Univariate



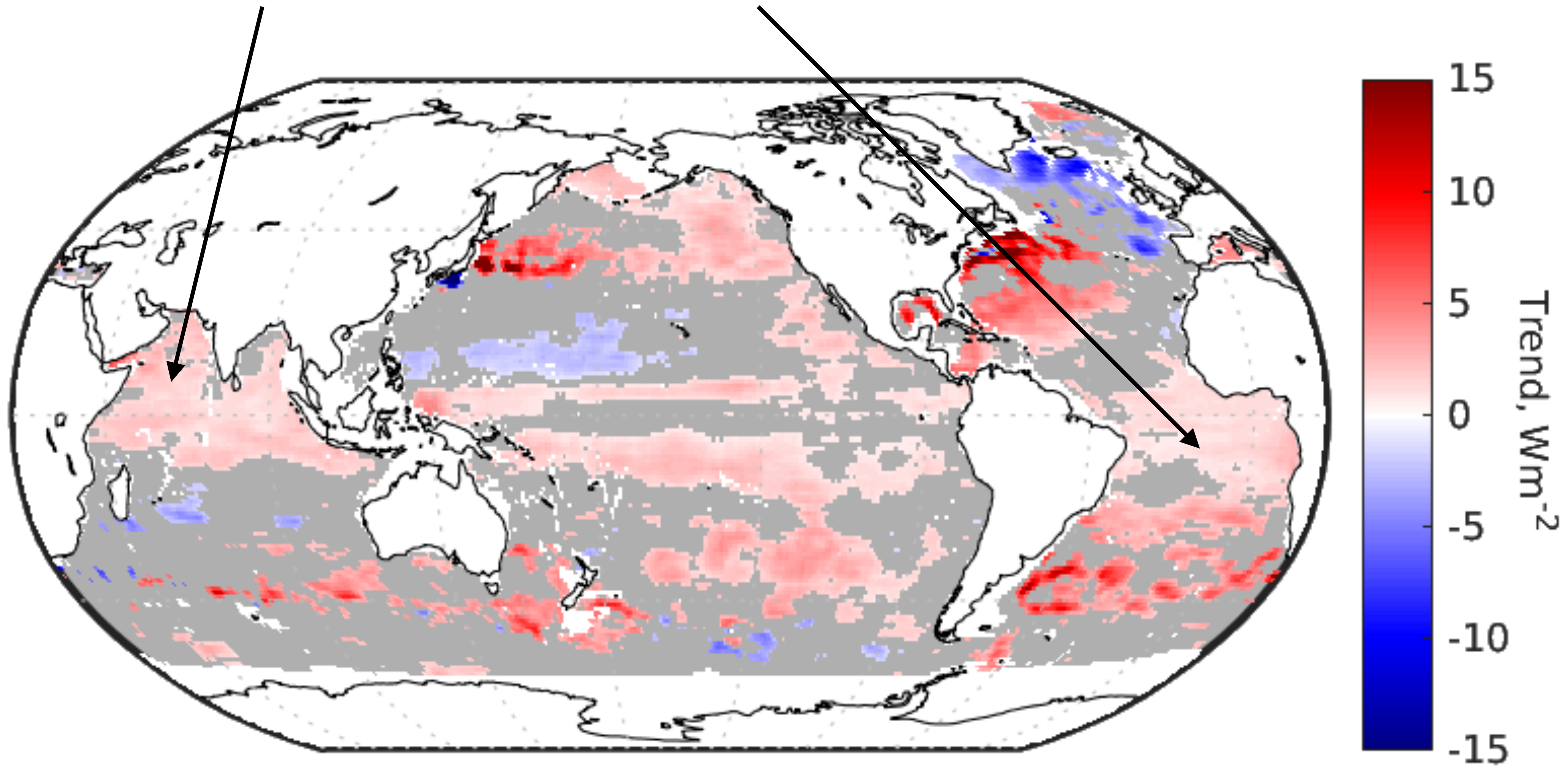
Bivariate

Bivariate extension: regional OHC uncertainty comparison

More significant regions

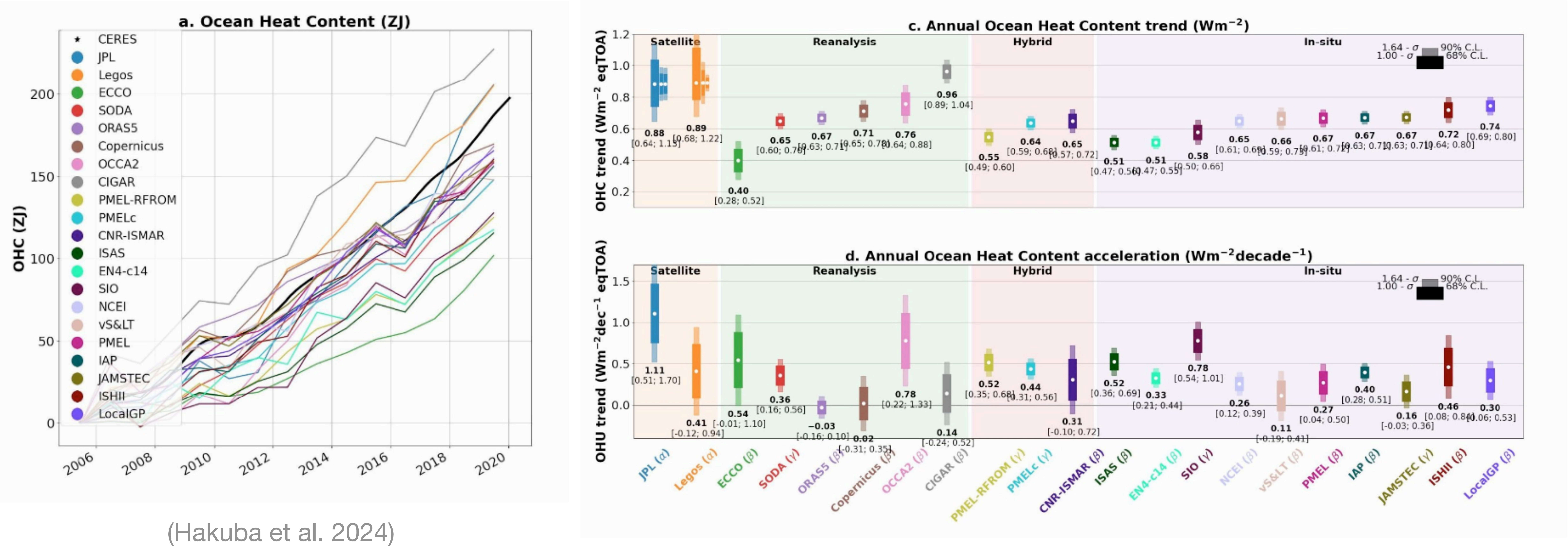


Univariate

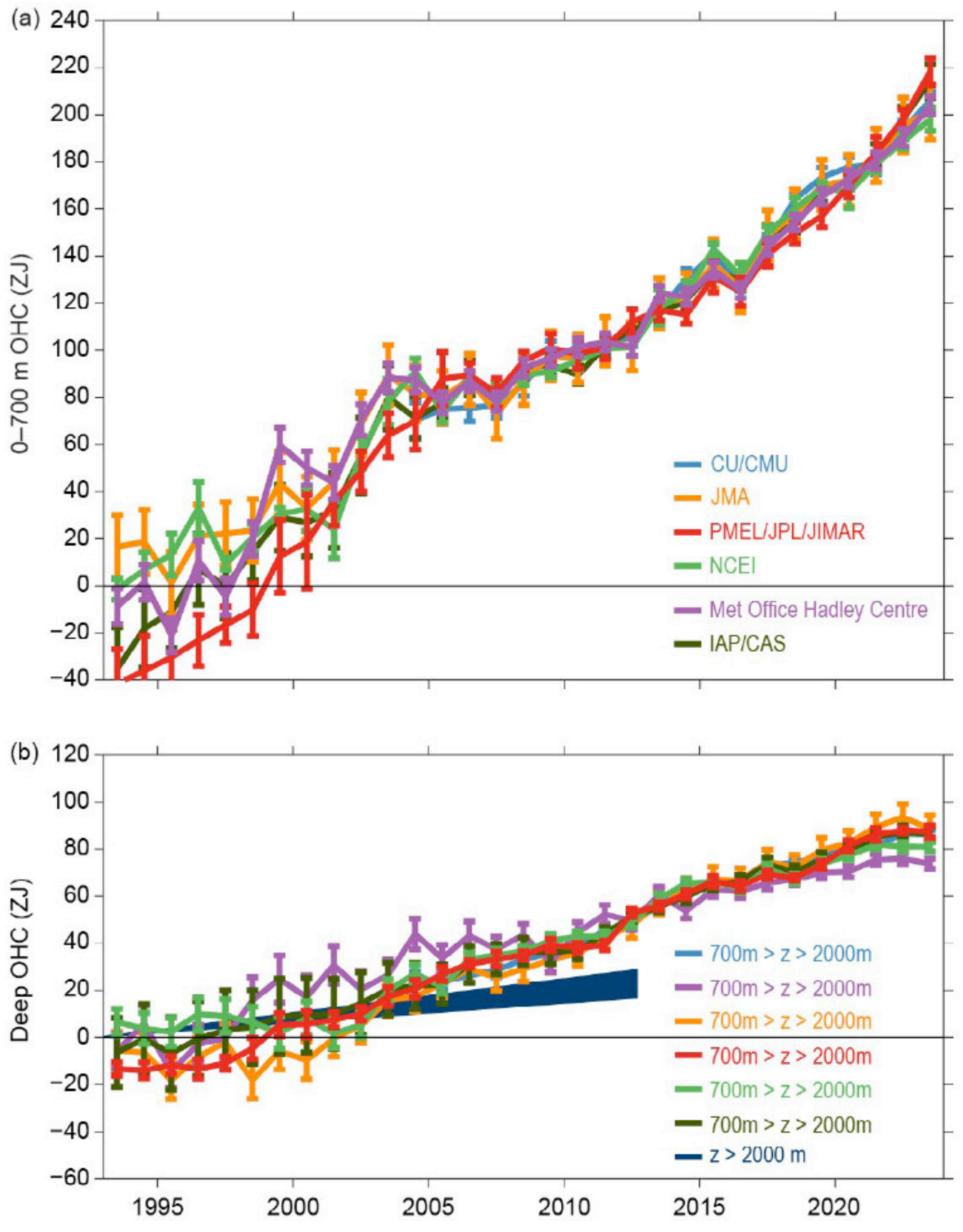


Bivariate

Our framework in action: Intercomparison efforts



(Hakuba et al. 2024)



(Johnson et al. 2024)

We are in the State of the Climate in 2023 report!
(blue line)

Local methods still have computational challenges

For each window, we numerically optimize the GP log-likelihood over years t:

$$\log(\mathcal{L}_{\mathcal{W}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_\epsilon | \boldsymbol{y}_t)) = -\frac{1}{2} \sum_{t \in D_t} [\log \det(\boldsymbol{K}_t(\boldsymbol{\theta}_f) + \boldsymbol{\Sigma}_\epsilon(\boldsymbol{\theta}_\epsilon)) + \boldsymbol{y}_t^{\mathrm{T}} (\boldsymbol{K}_t(\boldsymbol{\theta}_f) + \boldsymbol{\Sigma}_\epsilon(\boldsymbol{\theta}_\epsilon))^{-1} \boldsymbol{y}_t + n_t \log(2\pi)]$$

Local methods still have computational challenges

For each window, we numerically optimize the GP log-likelihood over years t:

$$\log(\mathcal{L}_{\mathcal{W}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_\epsilon | \boldsymbol{y}_t)) = -\frac{1}{2} \sum_{t \in D_t} [\log \det(\boldsymbol{K}_t(\boldsymbol{\theta}_f) + \boldsymbol{\Sigma}_\epsilon(\boldsymbol{\theta}_\epsilon)) + \boldsymbol{y}_t^{\mathrm{T}} (\boldsymbol{K}_t(\boldsymbol{\theta}_f) + \boldsymbol{\Sigma}_\epsilon(\boldsymbol{\theta}_\epsilon))^{-1} \boldsymbol{y}_t + n_t \log(2\pi)]$$

$\mathcal{O}(n_t^3)$

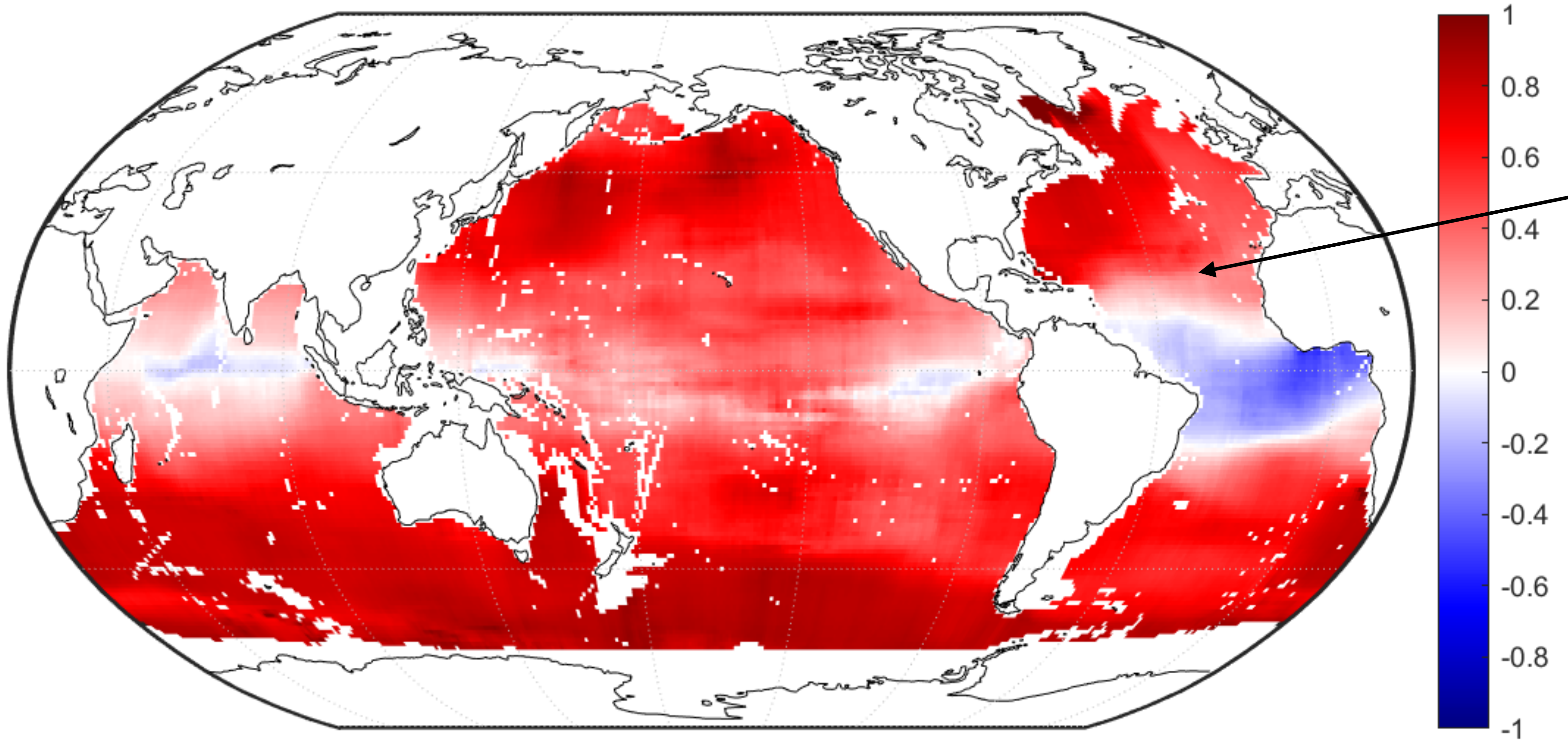
Local methods still have computational challenges

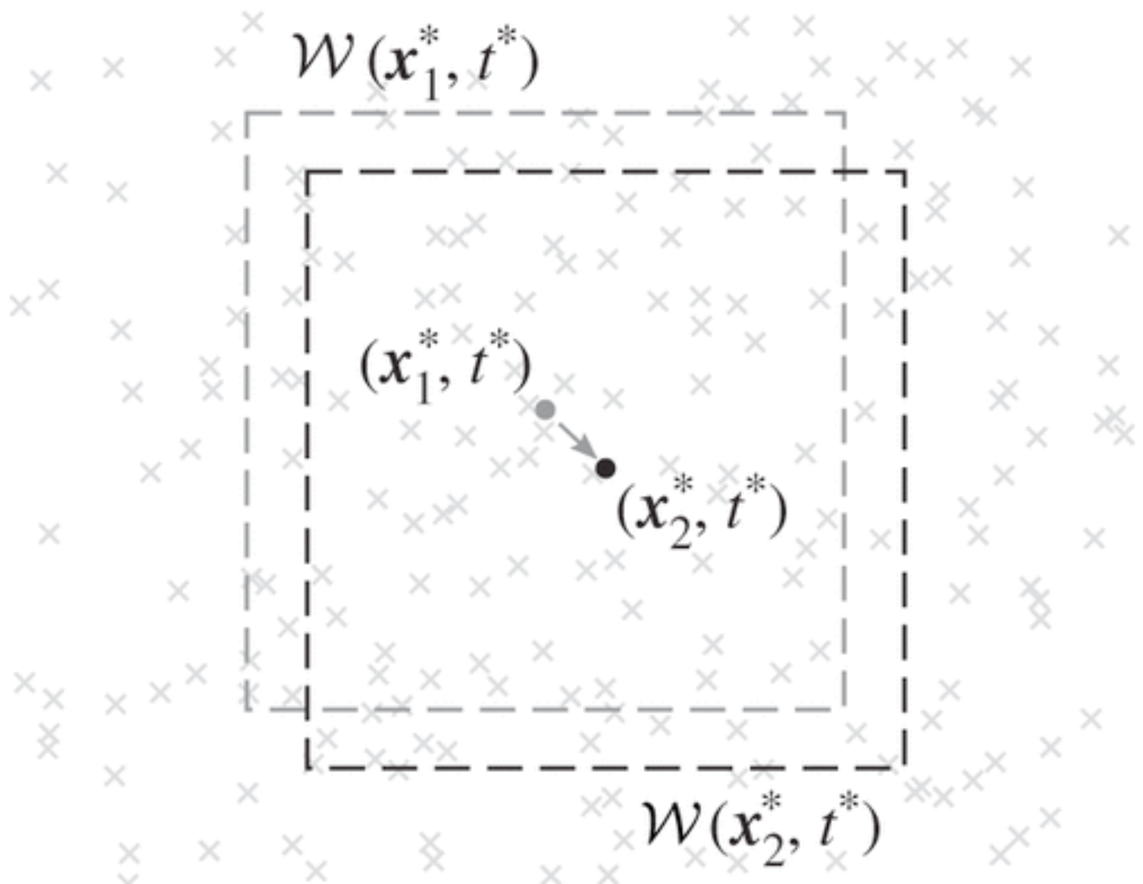
For each window, we numerically optimize the GP log-likelihood over years t:

$$\log(\mathcal{L}_{\mathcal{W}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_\epsilon | \boldsymbol{y}_t)) = -\frac{1}{2} \sum_{t \in D_t} [\log \det(\boldsymbol{K}_t(\boldsymbol{\theta}_f) + \boldsymbol{\Sigma}_\epsilon(\boldsymbol{\theta}_\epsilon)) + \boldsymbol{y}_t^{\mathrm{T}} (\boldsymbol{K}_t(\boldsymbol{\theta}_f) + \boldsymbol{\Sigma}_\epsilon(\boldsymbol{\theta}_\epsilon))^{-1} \boldsymbol{y}_t + n_t \log(2\pi)]$$

$\mathcal{O}(n_t^3)$

(ex. cross-correlation)





~30,000 windows!

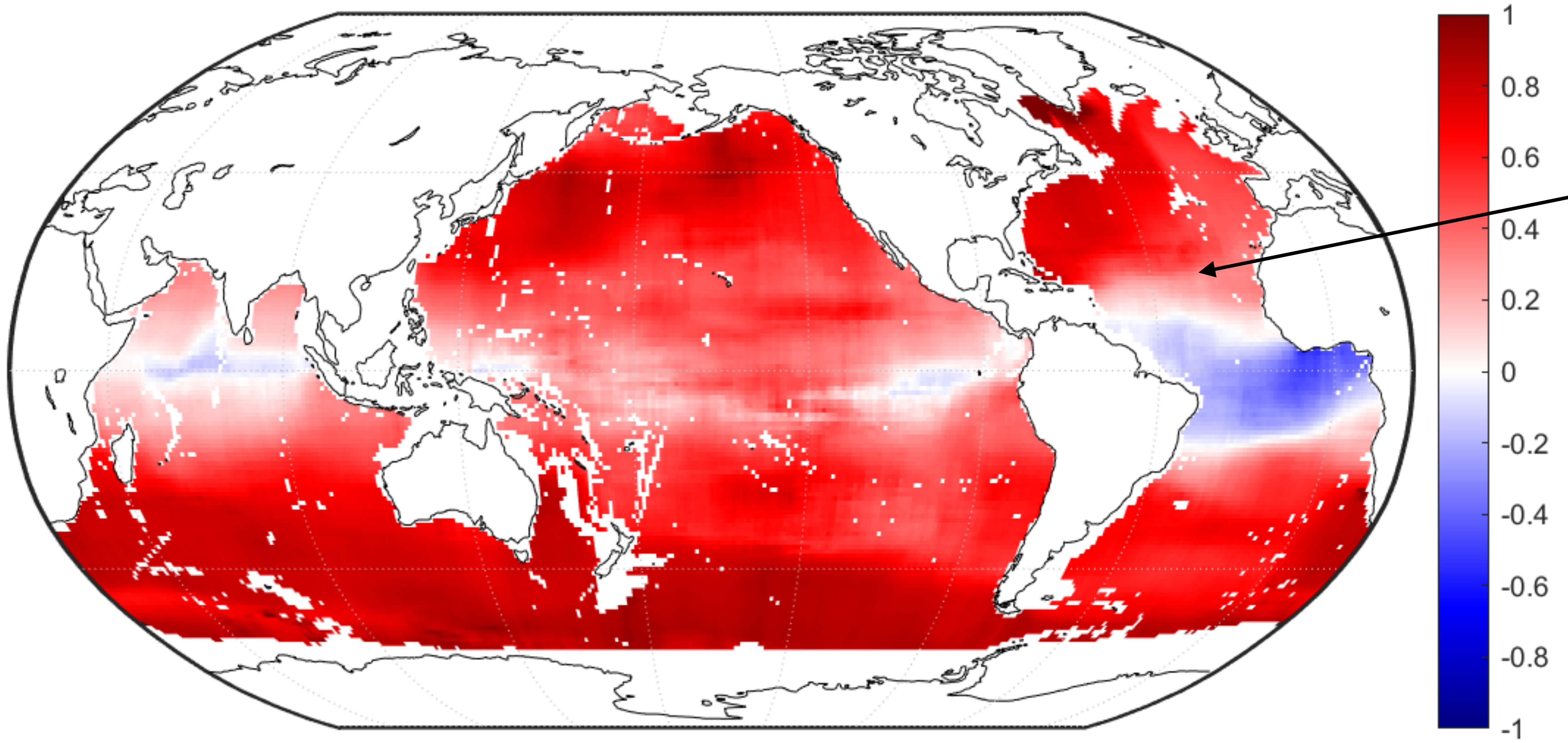
Local methods still have computational challenges

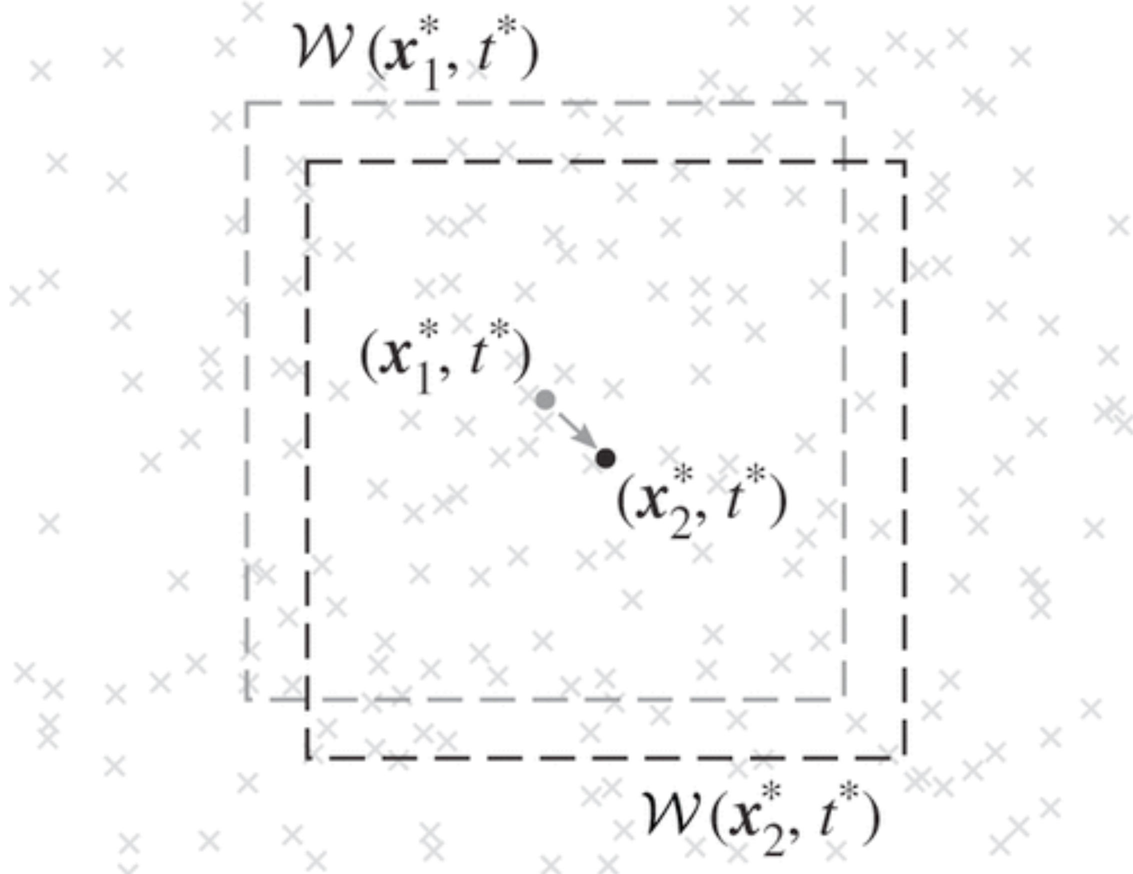
For each window, we numerically optimize the GP log-likelihood over years t:

$$\log(\mathcal{L}_{\mathcal{W}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_\epsilon | \boldsymbol{y}_t)) = -\frac{1}{2} \sum_{t \in D_t} [\log \det(\boldsymbol{K}_t(\boldsymbol{\theta}_f) + \boldsymbol{\Sigma}_\epsilon(\boldsymbol{\theta}_\epsilon)) + \boldsymbol{y}_t^{\mathrm{T}} (\boldsymbol{K}_t(\boldsymbol{\theta}_f) + \boldsymbol{\Sigma}_\epsilon(\boldsymbol{\theta}_\epsilon))^{-1} \boldsymbol{y}_t + n_t \log(2\pi)]$$

$\mathcal{O}(n_t^3)$

(ex. cross-correlation)





~30,000 windows!

*Parallelizable, but still ~2 days on a supercomputer

Computational improvements: neural likelihood

We propose to make the parameter estimation process more efficient via **neural likelihood** (Walchessen et al. 2024)

Idea: Train a neural network classifier $h(\boldsymbol{y}_t, \boldsymbol{\theta})$ so that $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}_t)$ is proportional to $\frac{h(\boldsymbol{y}_t,\boldsymbol{\theta})}{1-h(\boldsymbol{y}_t,\boldsymbol{\theta})}$

(which we can then evaluate the classifier to obtain the likelihood surface)

Neural likelihood is **amortized** - we train once and evaluation is vectorized, so parameter estimation will be much faster than numerical MLE

Computational improvements: neural likelihood

How do we choose the classifier?

Walchessen et al. 2024 use a convolutional neural network (CNN),
but Argo observations **are not on a regular grid!**

Computational improvements: neural likelihood

How do we choose the classifier?

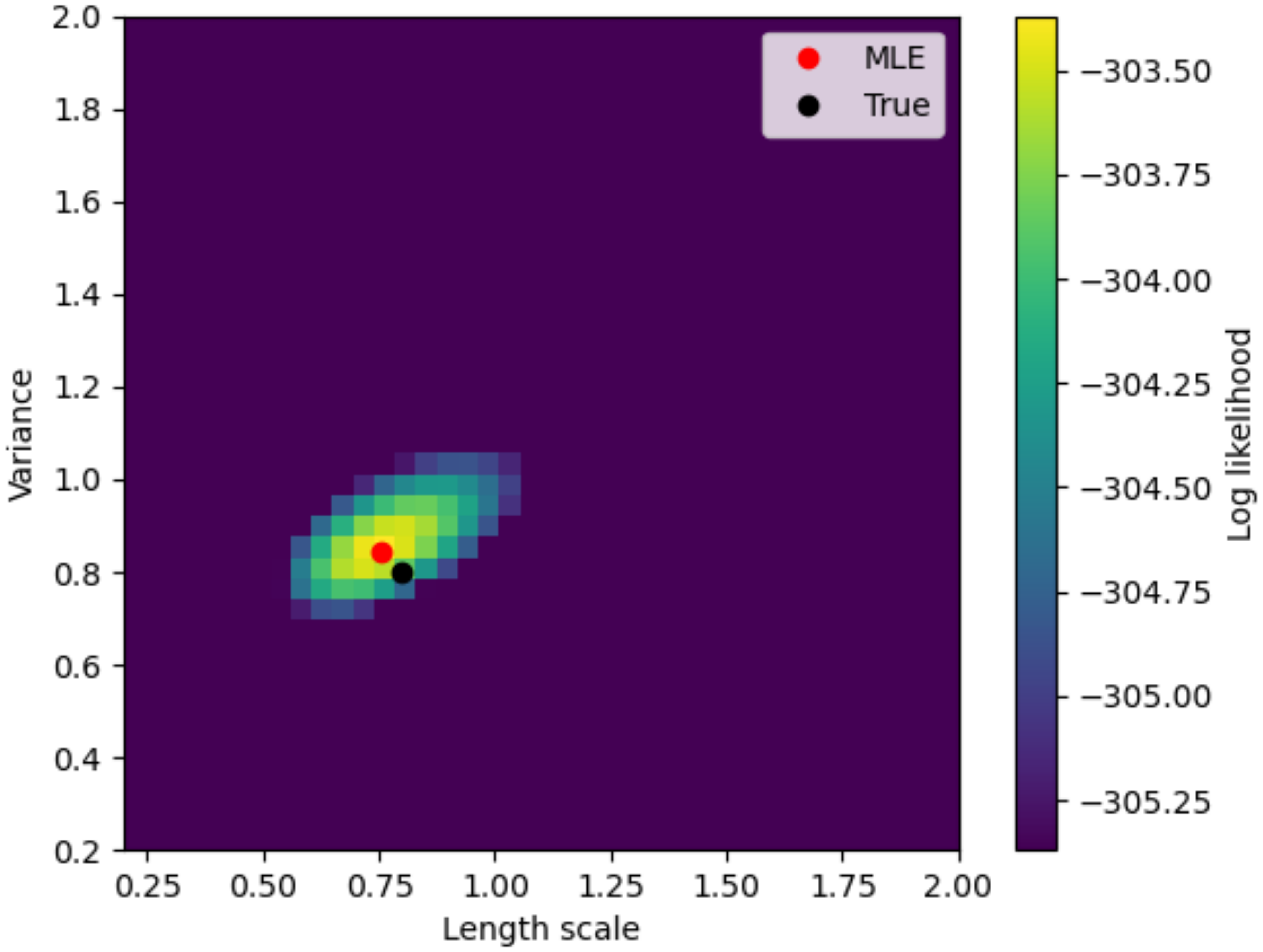
Walchessen et al. 2024 use a convolutional neural network (CNN),
but Argo observations **are not on a regular grid!**

We replace the CNN with a GNN, which can handle irregular inputs.

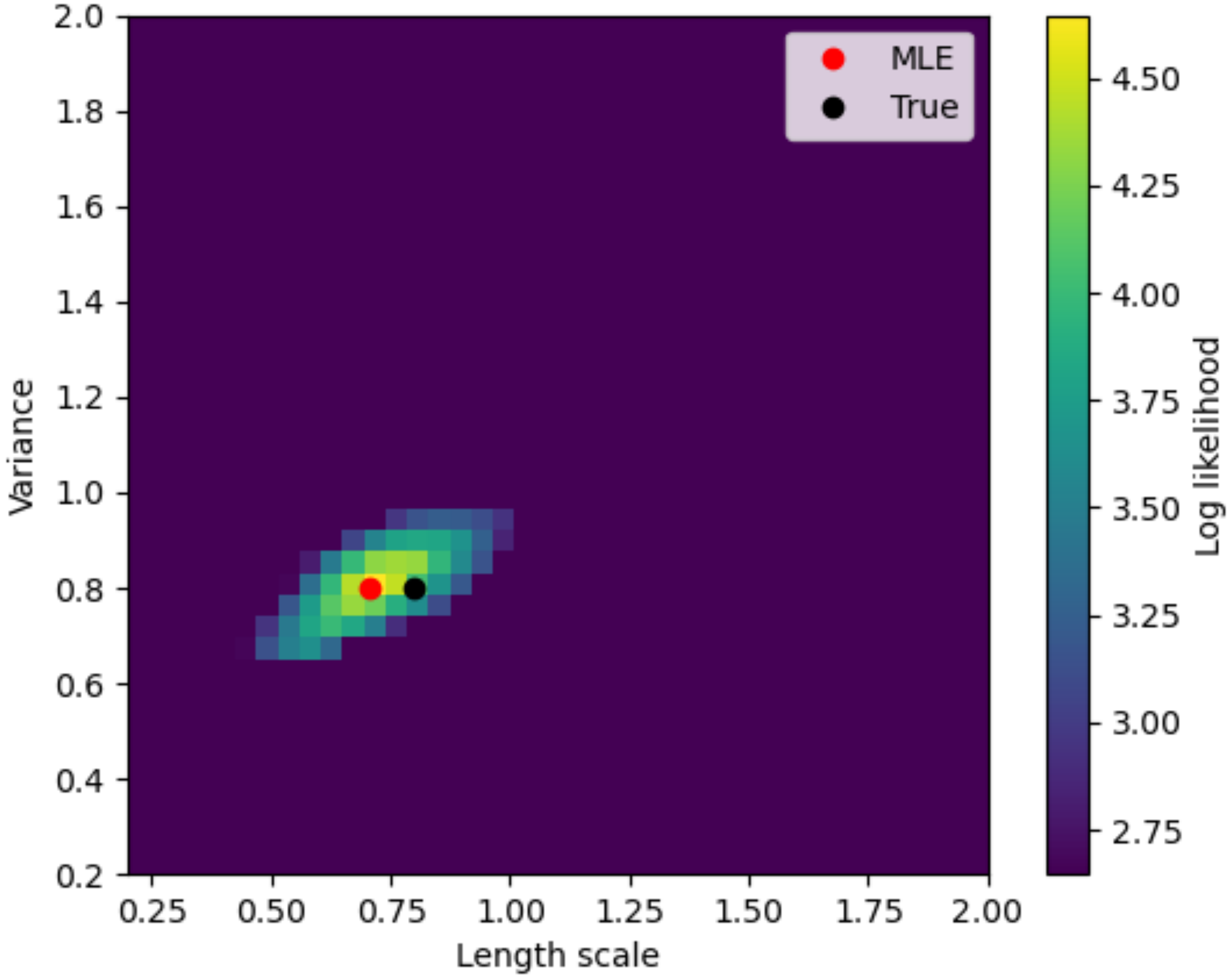
GNNs have been used for neural Bayes estimation (Sainsbury-Dale et al. (2025))

Preliminary results: likelihood surfaces

Simple test case: isotropic Gaussian process with variance = 0.8 and length scale = 0.8
(network trained on 30,000 point patterns sampled from a uniform distribution)

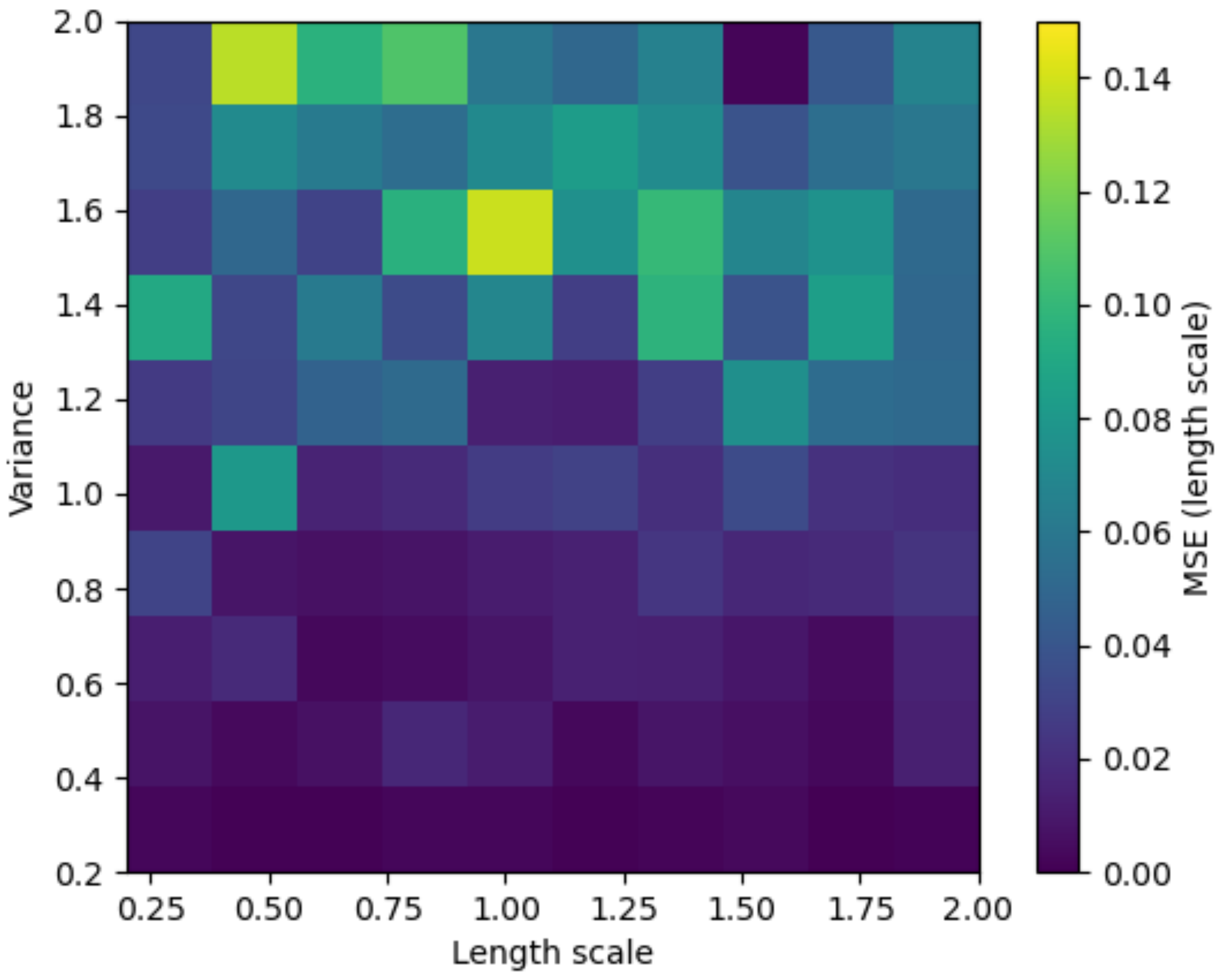


True likelihood surface

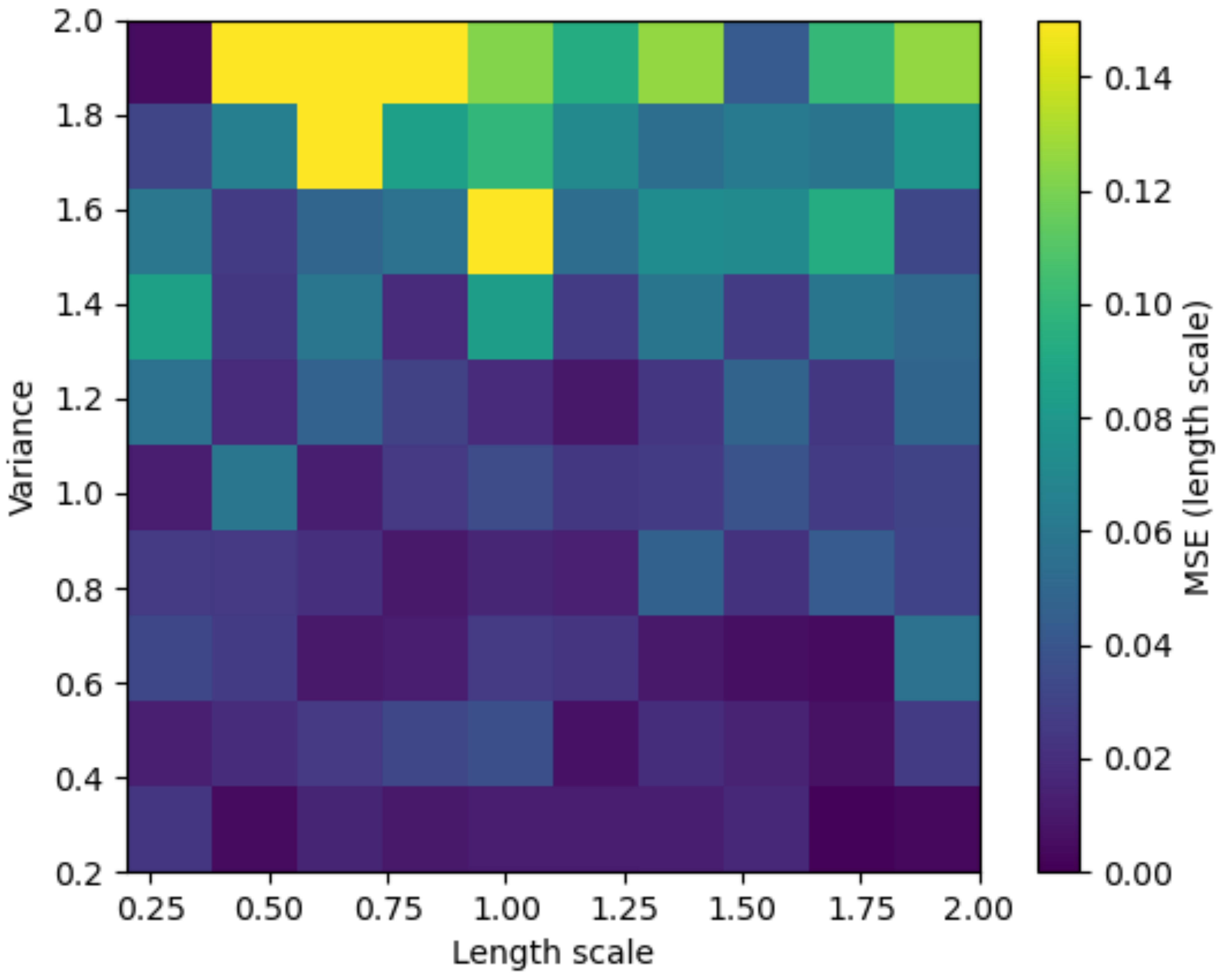


Neural likelihood surface

Preliminary results: length scale estimation error

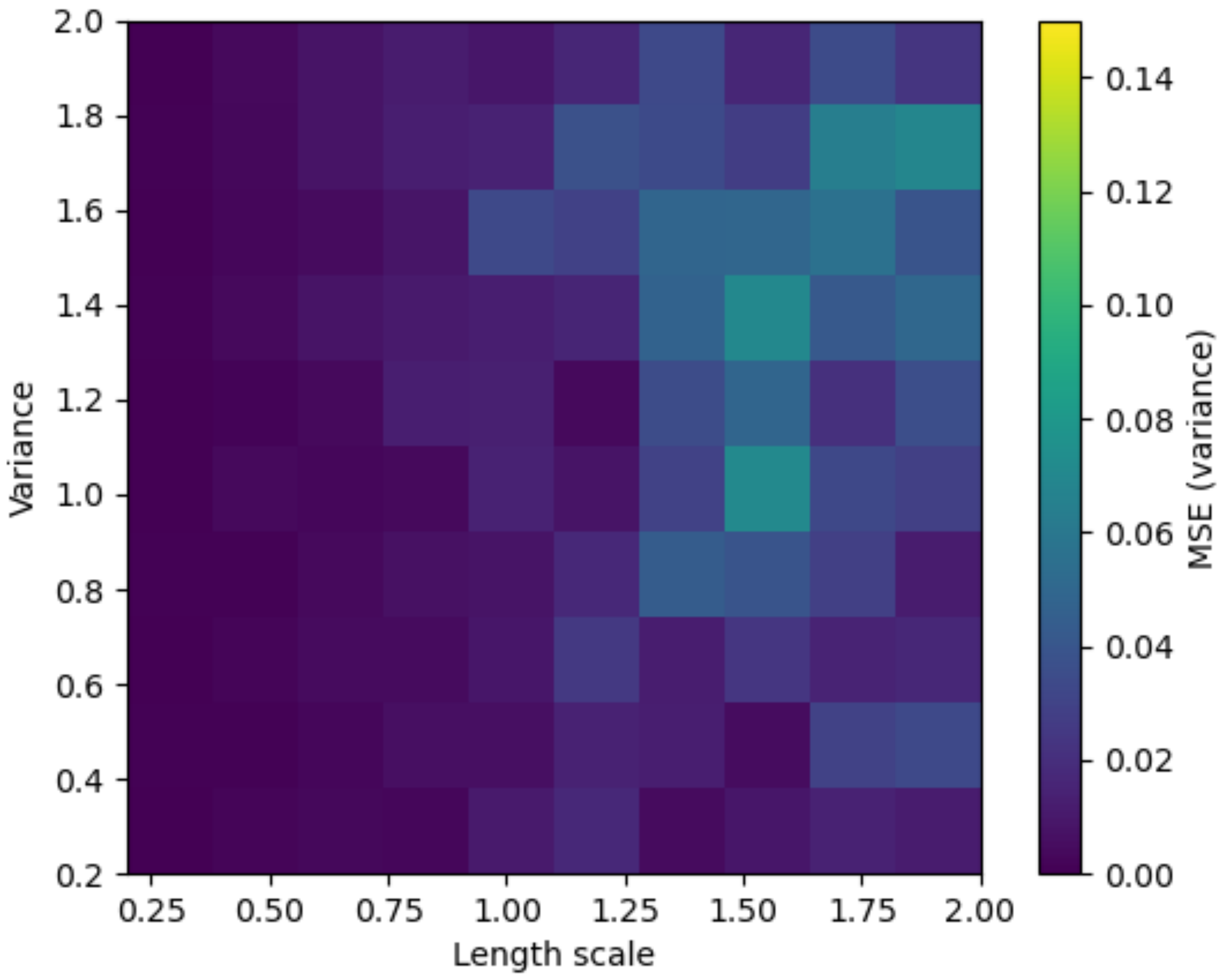


True MLE error

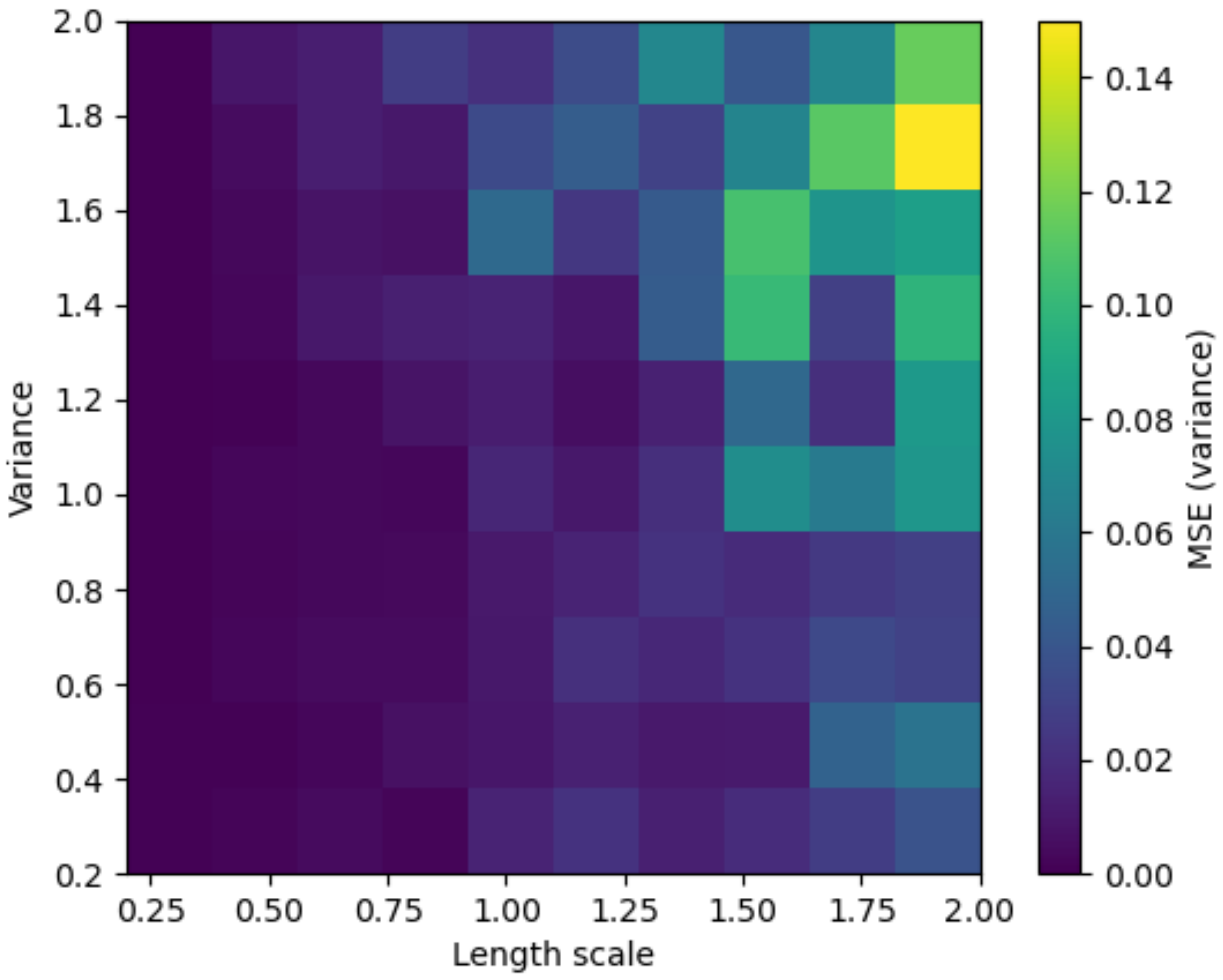


Neural MLE error

Preliminary results: variance estimation error



True MLE error



Neural MLE error

Ongoing work and implications

Ongoing work:

- Training the network using Argo sampling patterns
- Extending to the anisotropic case

We hope these computational improvements will facilitate **further extensions and scientific applications** that were previously infeasible:

- 4D mapping (e.g., including depth in the covariance model)
- Increasing the mapped field resolution
- Including other data sources (e.g., satellite and mooring data)
- Joint mapping of multiple fields (e.g., with sea surface height and sea surface temperature)

Questions/Comments?

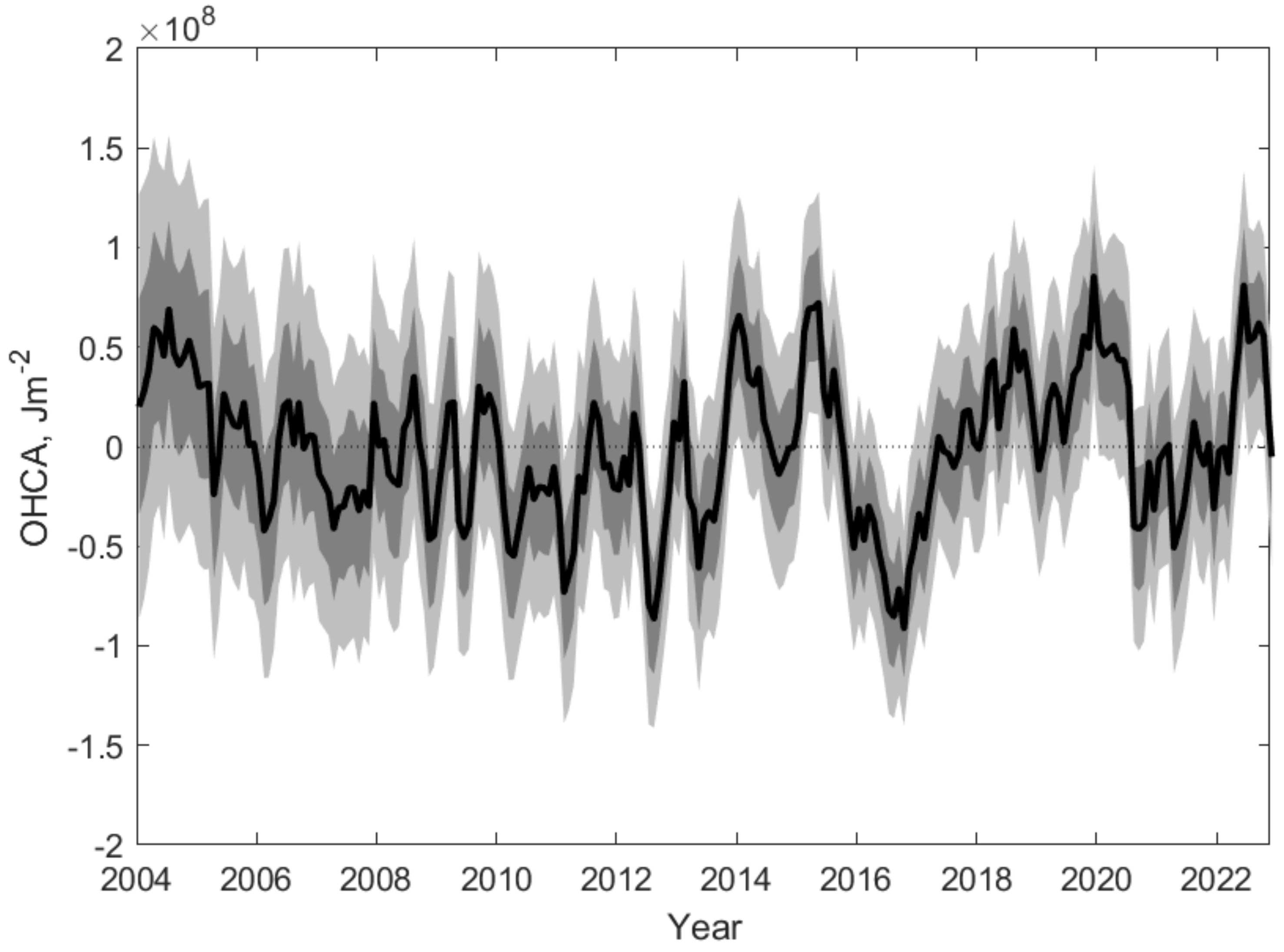
Contact: thea@stat.cmu.edu

Data product

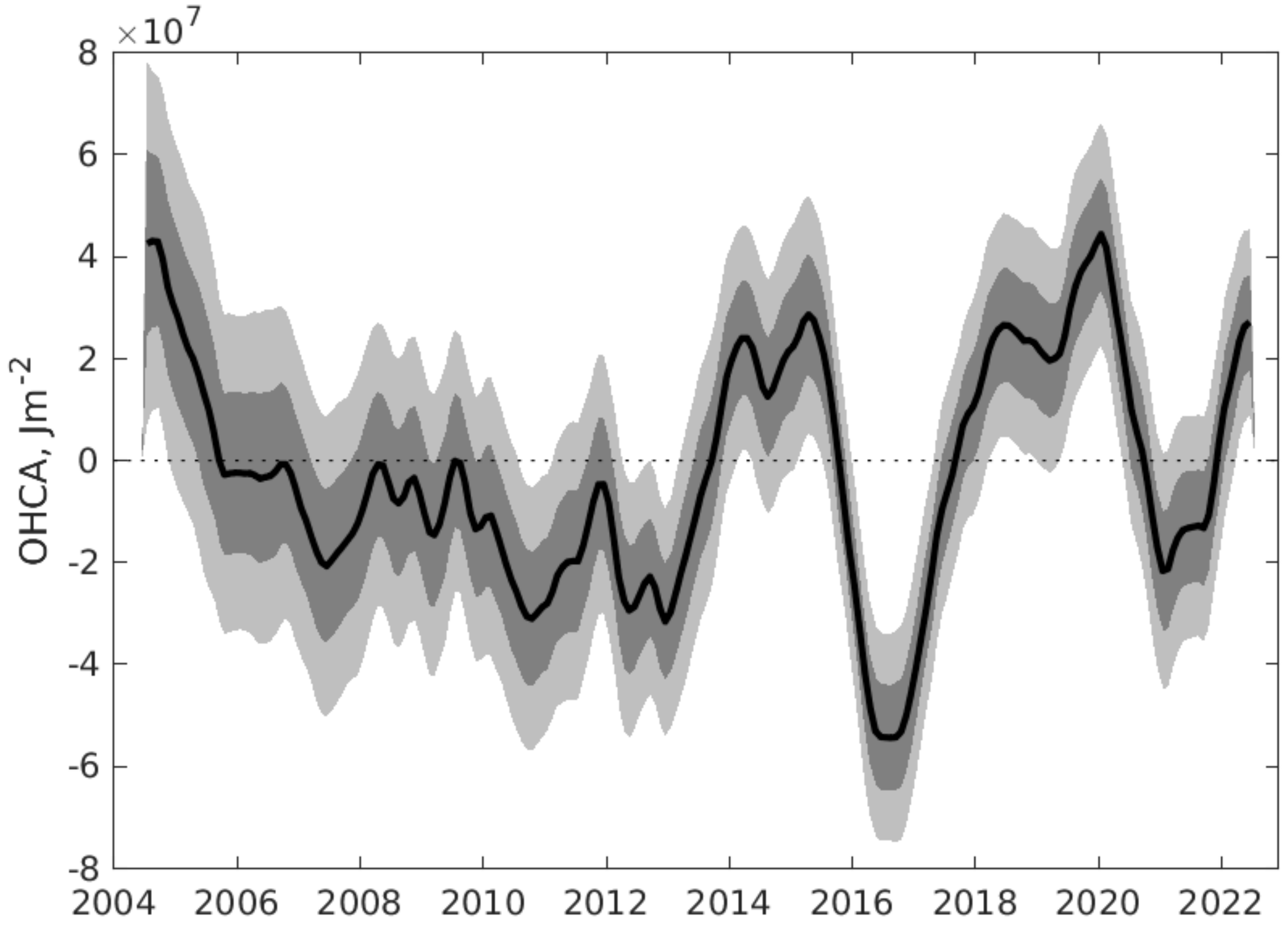
Intercomparison paper

Appendix

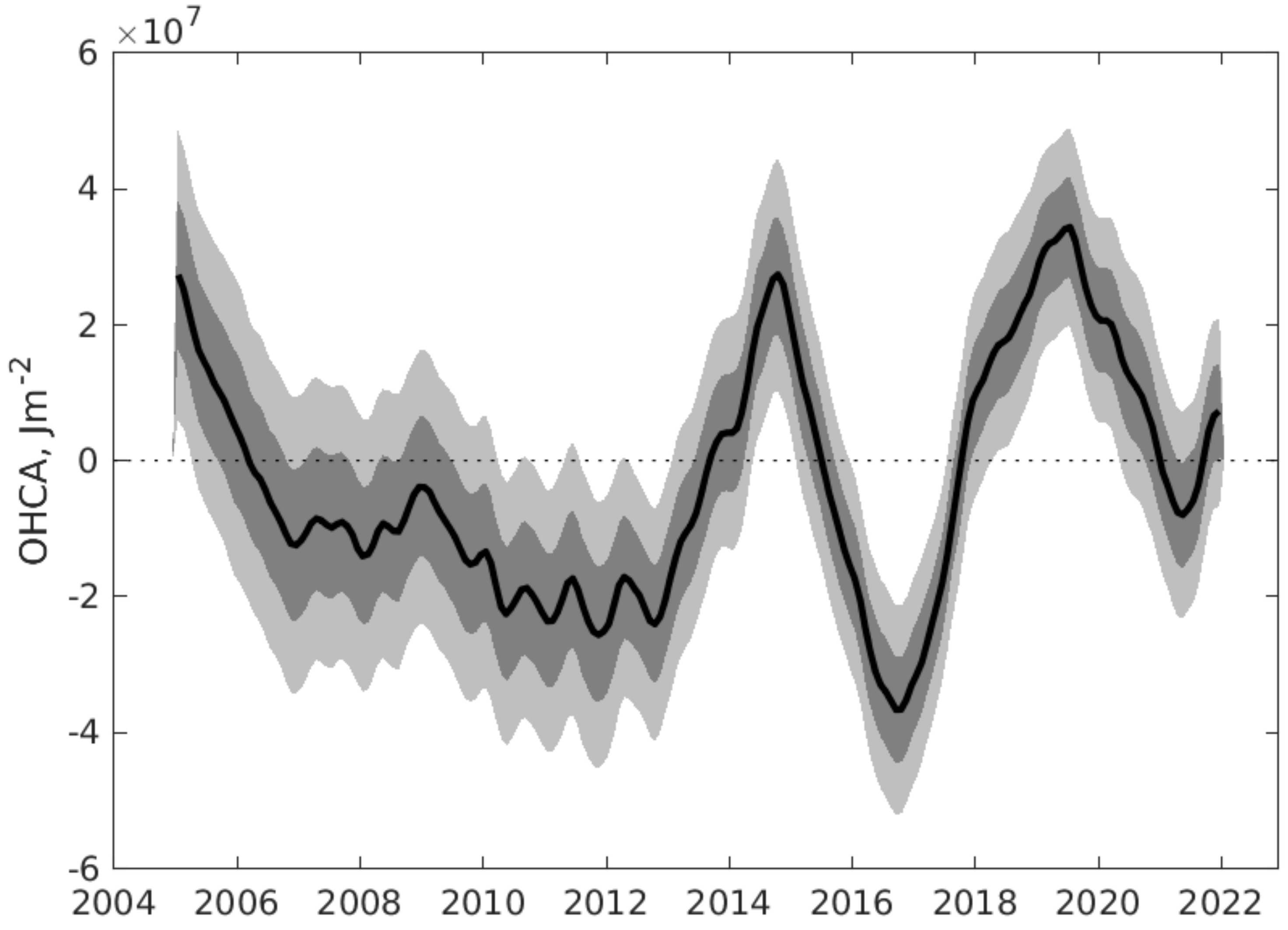
Uncertainties: Global OHC anomalies



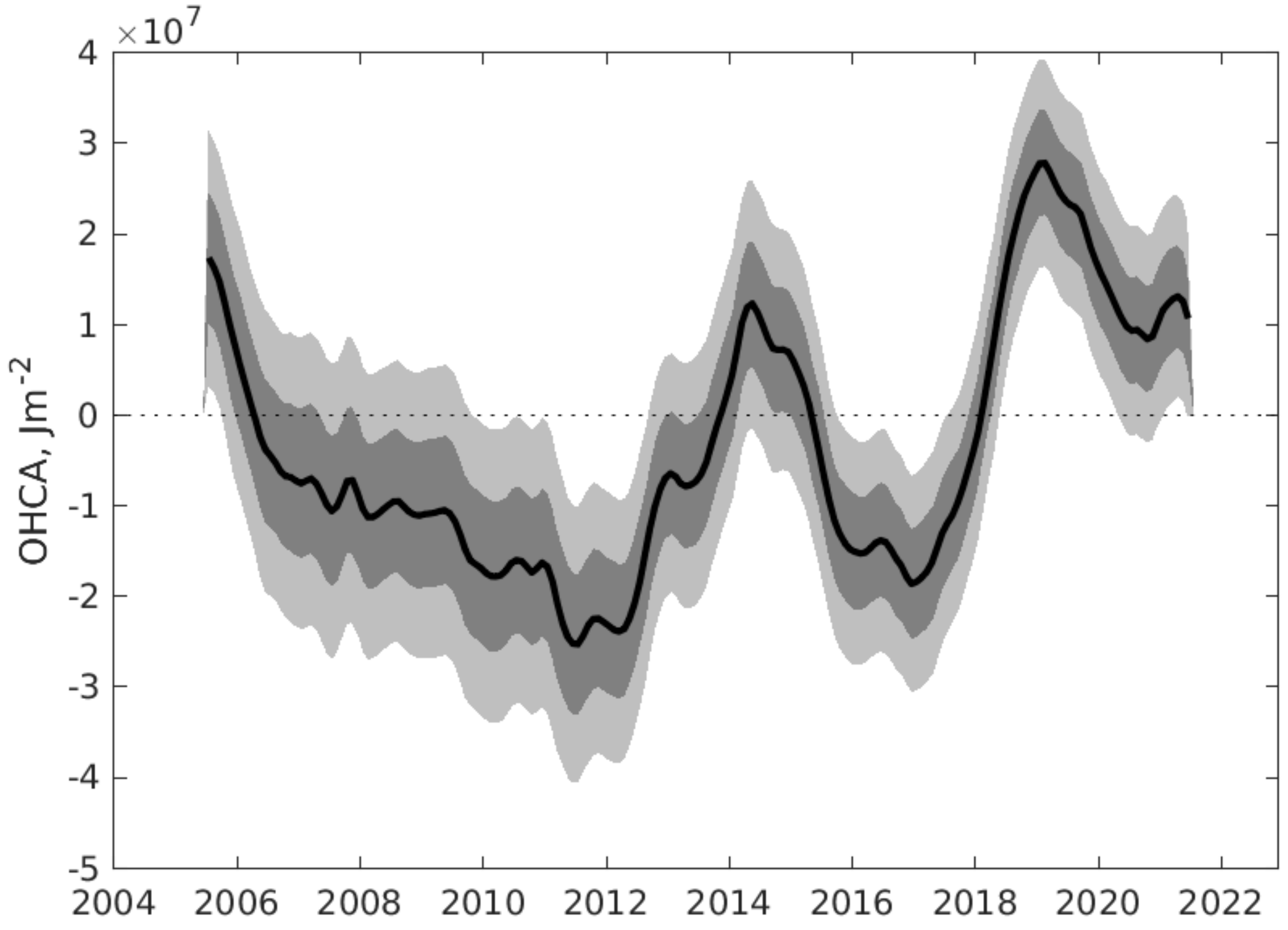
Monthly



12-month
moving average

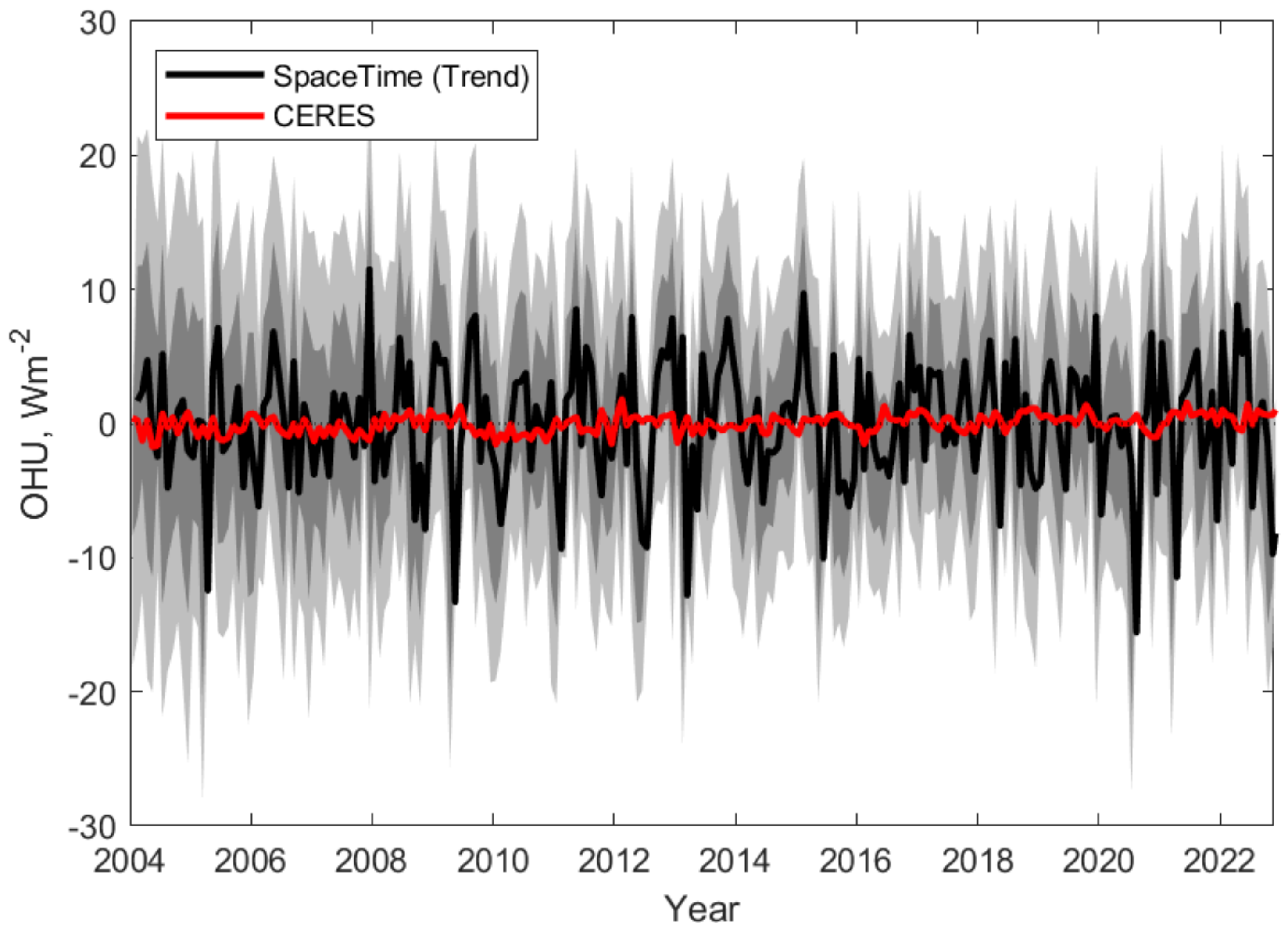


24-month
moving average

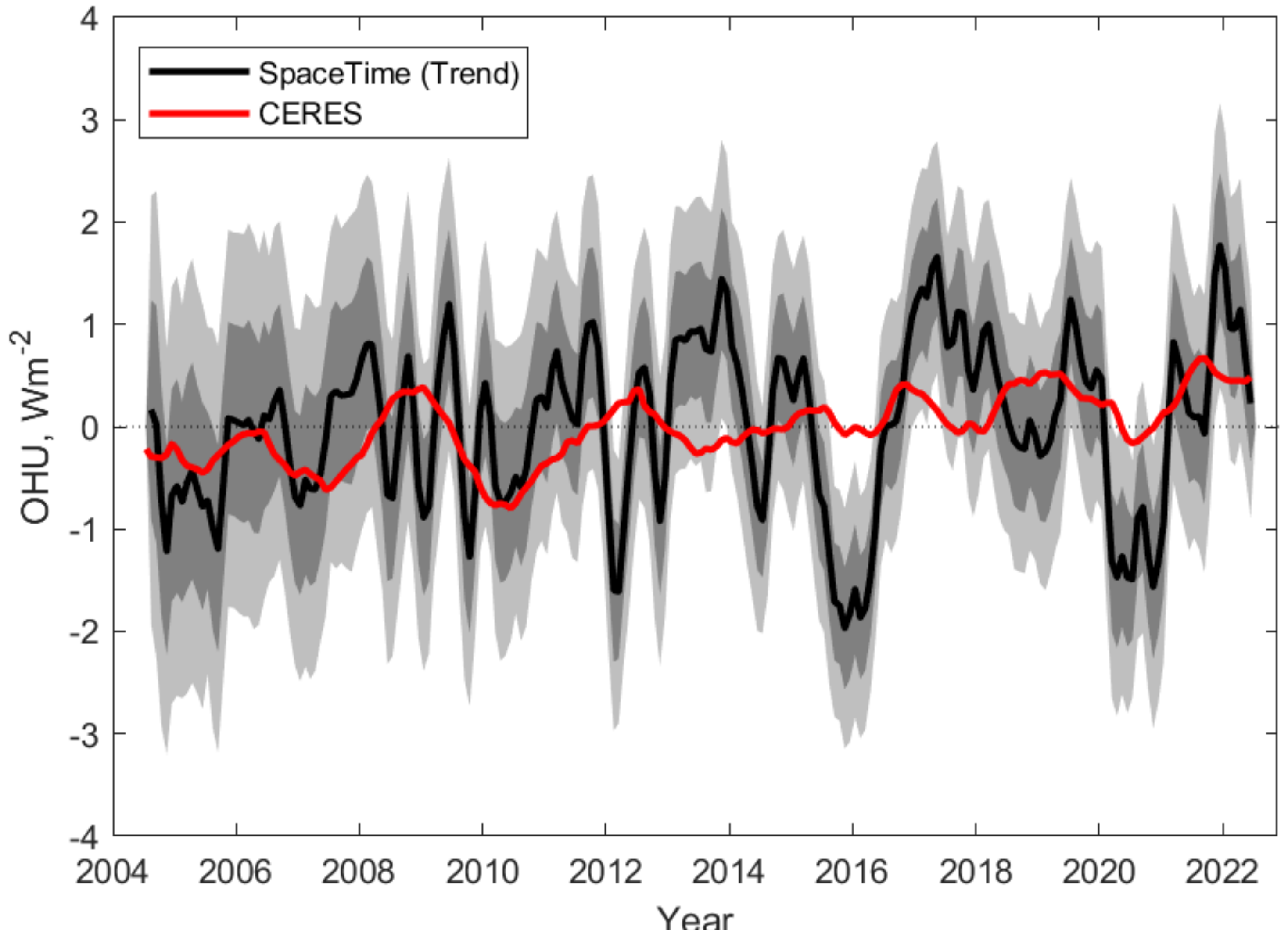


36-month
moving average

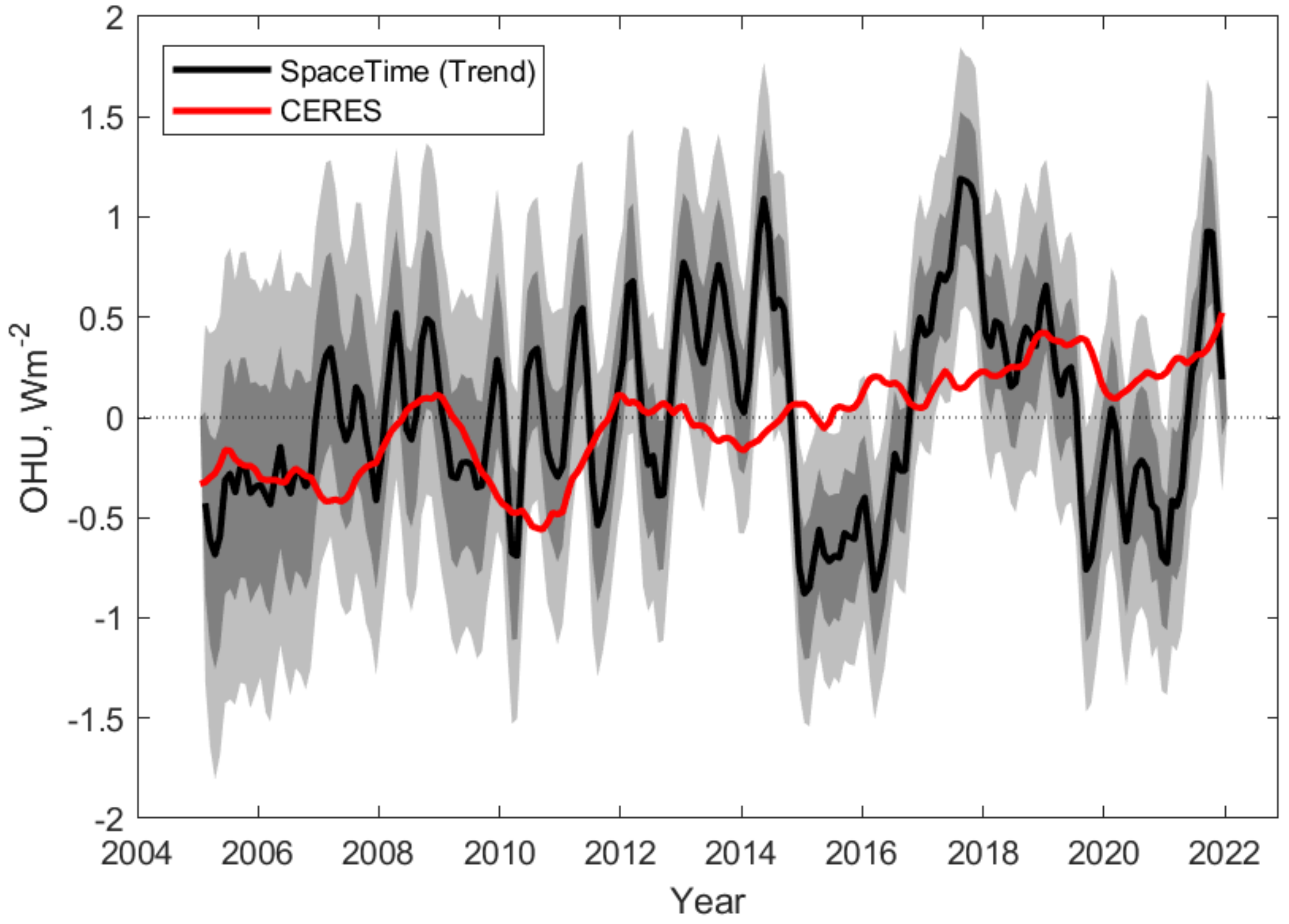
Uncertainties: Global ocean heat uptake (OHU) anomalies



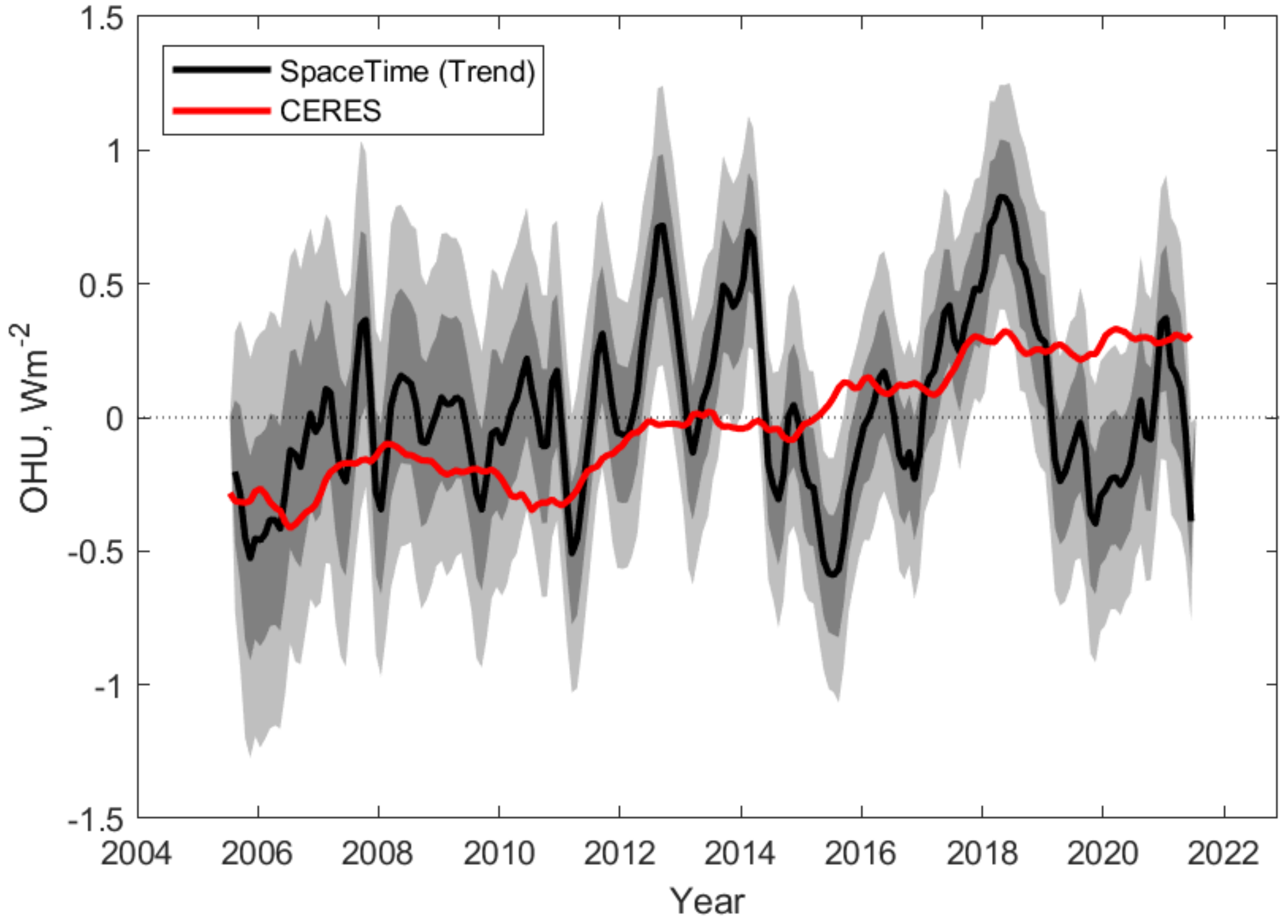
Monthly



12-month
moving average

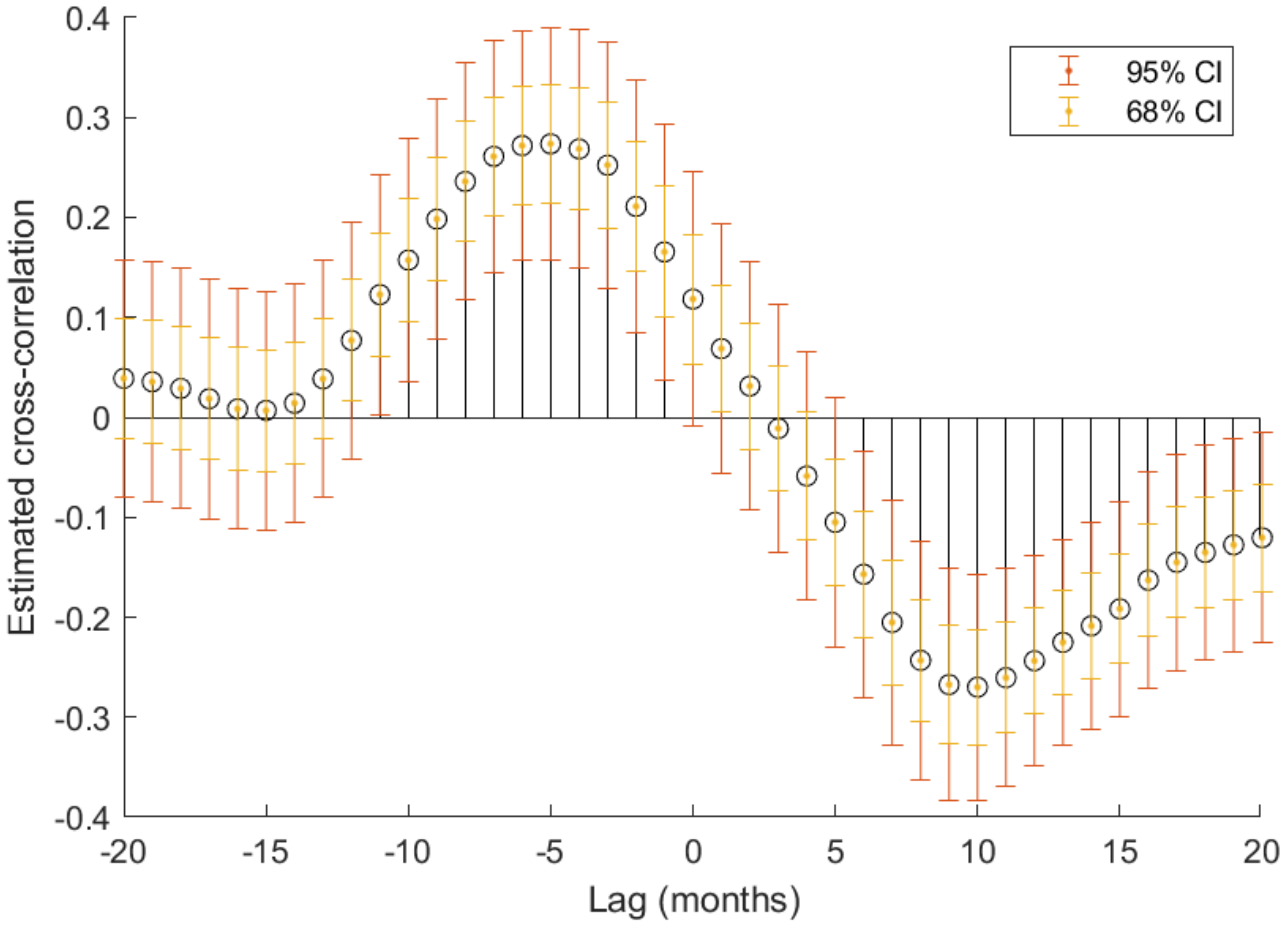


24-month
moving average



36-month
moving average

Uncertainties: Correlation between top layer OHC anomalies and ONI



GNN training details

- Spatial domain: [-10, 10] x [-10, 10]
- Simulated parameters: 3000 on [0, 2.5] x [0, 2.5], generated using Latin Hypercube sampling
- Point patterns: 10 per parameter pair (30000 total), generated from a uniform distribution

GNN training details

We will implement a parameter estimation procedure:

1. Generate a candidate parameter vector and subsample the local windows (right)

2. Simulate realizations from the local GP model at the locations in the subsampled windows

3. Repeat to form a training dataset

4. Train the classifier **(GNN + FCN)** to discern between the training dataset and its permuted version

5. Evaluate to obtain the likelihood surface for all years

6. Sum over all years and find the maximum