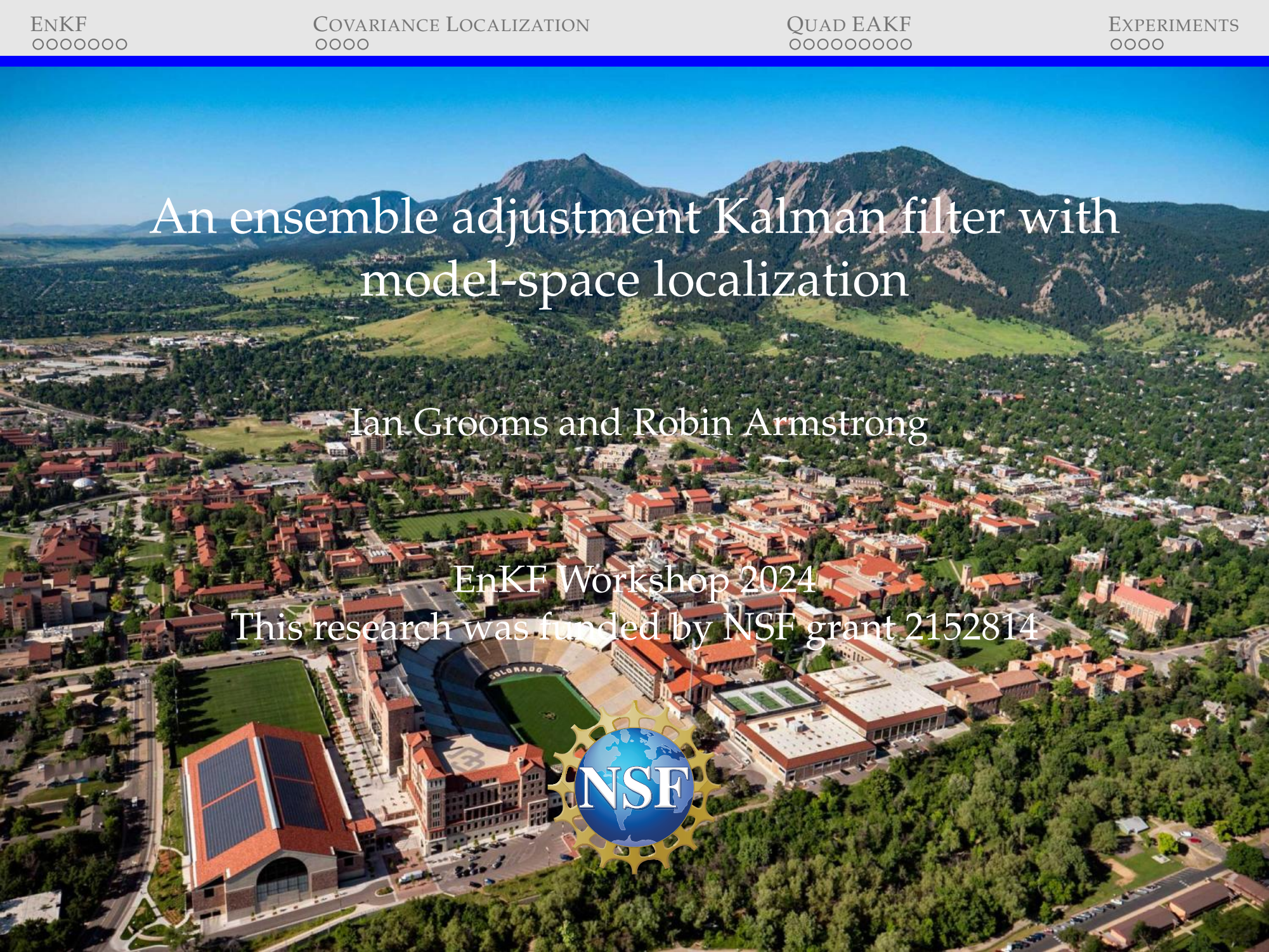
An ensemble adjustment Kalman filter with model-space localization

Ian Grooms and Robin Armstrong

EnKF Workshop 2024

This research was funded by NSF grant 2152814



At its most basic level, an ensemble filter takes in a forecast ensemble and produces an analysis ensemble

$$\left\{\vec{x}_n^f\right\}_{n=1}^N \rightarrow \{\vec{x}_n^a\}_{n=1}^N .$$

All EnKFs and all Particle Filters do this.

EnKFs are based on two fundamental approximations. **First**: the joint distribution of $\vec{X}$ and $\vec{Y}$ is approximated as Gaussian.

$$\begin{pmatrix} \vec{X} \\ \vec{Y} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \vec{\mu}_x \\ \vec{\mu}_y \end{pmatrix}, \begin{bmatrix} \mathbf{\Sigma}_{xx} & \mathbf{\Sigma}_{xy} \\ \mathbf{\Sigma}_{yx} & \mathbf{\Sigma}_{yy} \end{bmatrix} \right).$$

Then $\vec{X}|(\vec{Y} = \vec{y})$ is also Gaussian, with mean and covariance

$$\mathbb{E}\left[\vec{X}|(\vec{Y} = \vec{y})\right] = \vec{\mu}_x + \mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-1}(\vec{y} - \vec{\mu}_y),$$

$$\mathrm{Cov}\left[\vec{X}|(\vec{Y} = \vec{y})\right] = \mathbf{\Sigma}_{xx} - \mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-1}\mathbf{\Sigma}_{yx}.$$

For the rest of the talk we will assume Gaussianity.

EnKFs commonly assume an observation model of the form

$$\vec{Y} = \vec{H} + \vec{\epsilon} \quad \vec{H} := \vec{h}(\vec{X})$$

where $\vec{\epsilon}$ is independent of $\vec{X}$ and has zero mean (wlog) and covariance $\mathbf{R}$.

$$\Rightarrow \begin{pmatrix} \vec{X} \\ \vec{Y} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \vec{\mu}_x \\ \vec{\mu}_h \end{pmatrix}, \begin{bmatrix} \mathbf{\Sigma}_{xx} & \mathbf{\Sigma}_{xh} \\ \mathbf{\Sigma}_{hx} & \mathbf{\Sigma}_{hh} + \mathbf{R} \end{bmatrix} \right).$$

$$\mathbb{E}\left[\vec{X} | (\vec{Y} = \vec{y})\right] = \vec{\mu}_x + \mathbf{\Sigma}_{xh} \left(\mathbf{\Sigma}_{hh} + \mathbf{R}\right)^{-1} (\vec{y} - \vec{\mu}_h),$$

$$\mathrm{Cov}\left[\vec{X} | (\vec{Y} = \vec{y})\right] = \mathbf{\Sigma}_{xx} - \mathbf{\Sigma}_{xh} \left(\mathbf{\Sigma}_{hh} + \mathbf{R}\right)^{-1} \mathbf{\Sigma}_{hx}.$$

You can do EnKFs without the standard observation model, but the rest of this talk will use it.

EnKFs are based on two fundamental approximations. **Second**: the parameters of the joint Gaussian are estimated from the ensemble.

To estimate means and (cross-)covariance involving $\vec{X}$ and/or $\vec{h}(\vec{X})$ you need a joint ensemble $\left\{\vec{x}_n^f, \vec{h}_n^f\right\}_{n=1}^N$.

EnKFs are based on two fundamental approximations. **Second**: the parameters of the joint Gaussian are estimated from the ensemble.

To estimate means and (cross-)covariance involving $\vec{X}$ and/or $\vec{h}(\vec{X})$ you need a joint ensemble $\left\{\vec{x}_n^f, \vec{h}_n^f\right\}_{n=1}^N$.

OK, we made some approximations. How does this help us move from forecast ensemble to analysis ensemble?

Suppose that you have $\vec{X}_f \sim \mathcal{N}(\vec{\mu}_x, \mathbf{\Sigma}_{xx})$ and you want to convert it to $\vec{X}_a \sim \mathcal{N}(\vec{\mu}_a, \mathbf{C})$. How can you do it?

Suppose that you have $\vec{X}_f \sim \mathcal{N}(\vec{\mu}_x, \mathbf{\Sigma}_{xx})$ and you want to convert it to $\vec{X}_a \sim \mathcal{N}(\vec{\mu}_a, \mathbf{C})$. How can you do it?

$$\vec{X}_a = \vec{\mu}_a + \mathbf{C}^{1/2}\mathbf{\Sigma}_{xx}^{-1/2}\left(\vec{X}_f - \vec{\mu}_x\right).$$

Subtract the old mean, change the covariance, add the new mean, done.

Suppose that you have $\vec{X}_f \sim \mathcal{N}(\vec{\mu}_x, \mathbf{\Sigma}_{xx})$ and you want to convert it to $\vec{X}_a \sim \mathcal{N}(\vec{\mu}_a, \mathbf{C})$. How can you do it?

$$\vec{X}_a = \vec{\mu}_a + \mathbf{C}^{1/2}\mathbf{\Sigma}_{xx}^{-1/2}\left(\vec{X}_f - \vec{\mu}_x\right).$$

Subtract the old mean, change the covariance, add the new mean, done.

The EAKF (Anderson, 2001) is an ensemble approximation to this.

Let $\mathbf{X}_f$ be the forecast ensemble perturbation matrix and let $\mathbf{Y}$ be the forecast ensemble perturbation matrix for $\vec{H} = \vec{h}(\vec{X})$ (note the bad notation).

Three kinds of deterministic/square-root EnKFs:

- 'Adjustment': $\mathbf{X}_a = \mathbf{A}\mathbf{X}_f$ (e.g. EAKF: $\mathbf{A} = \mathbf{C}^{1/2}\mathbf{\Sigma}_{xx}^{-1/2}$)
- 'Transform': $\mathbf{X}_a = \mathbf{X}_f\mathbf{T}$ (e.g. SEIK, ETKF)
- 'Modified Gain': $\mathbf{X}_a = \mathbf{X}_f - \mathbf{G}\mathbf{Y}$ (Serial ESRF, GETKF)

We will be interested in the exact adjustment

$$\mathbf{A} = \left(\mathbf{C}\boldsymbol{\Sigma}_{xx}^{-1}\right)^{1/2}$$

where $\mathbf{C}$ is the exact posterior covariance and $\boldsymbol{\Sigma}_{xx}$ is the exact prior covariance.

With the standard observation model and linear obs the exact adjustment matrix is

$$\mathbf{A} = \left(\mathbf{I} + \boldsymbol{\Sigma}_{xx}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)^{-1/2}$$

To use this we insert an approximation $\mathbf{B}$ to $\mathbf{\Sigma}_{xx}$ into the formula for $\mathbf{A}$.

Sampling errors corrupt the ensemble approximation. Localization is a set of techniques to mitigate these errors.

If each element of $\vec{X}$ **and** of $\vec{H}$ has a spatial location, then there are several options for localization:

- Gain/Modified Gain localization
- R-inflation
- 'Local analysis' (cf. LEnKF, LETKF)

If each element of $\vec{X}$ has a spatial location **but** elements of $\vec{H}$ do not, then you have to resort to **covariance regularization**.

Satellite radiance observations are a canonical example: They report some kind of weighted integral along their line of sight.

Covariance regularization often takes the form of a Schur product:

$$\mathbf{\Sigma}_{xx} \approx \mathbf{T} \circ \mathbf{B}_e$$

$\mathbf{T}$ is the 'taper' (localization) matrix; $\mathbf{B}_e$ is the raw ensemble covariance.

This talk does not address how to choose $\mathbf{T}$; I assume it is given.

Bishop et al. (2017) add covariance localization to the GETKF as follows:

$$
\begin{aligned}
\mathbf{T} &\approx & \mathbf{T}_{1/2}\mathbf{T}_{1/2}^T & \quad \text{Optimal} \\
\mathbf{T} \circ \mathbf{B}_e &\approx & (\mathbf{T}_{1/2}\mathbf{T}_{1/2}^T) \circ \mathbf{B}_e & \quad \text{Not Optimal} \\
(\mathbf{T}_{1/2}\mathbf{T}_{1/2}^T) \circ \mathbf{B}_e &= & \mathbf{B} = \mathbf{L}\mathbf{L}^T & \quad \text{Exact} \\
\mathbf{A} &\approx & \left(\mathbf{I} + \mathbf{L}\mathbf{L}^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)^{-1/2} & \quad \text{Not Optimal} \\
\mathbf{A}\mathbf{X}_f &\approx & \hat{\mathbf{A}}\mathbf{X}_f = \mathbf{X}_f - \hat{\mathbf{G}}\mathbf{Y} & \quad \text{Not Optimal}
\end{aligned}
$$

The first approximation is optimal via an SVD, but the final error in the analysis ensemble is very indirectly related to the error in the fundamental approximation.

Farchi & Bocquet (2019) moved the approximation one step closer to the final result:

$$\begin{aligned} \mathbf{T} \circ \mathbf{B}_e &\approx \mathbf{L}\mathbf{L}^T && \text{Near Optimal} \\ \mathbf{A} &\approx \left(\mathbf{I} + \mathbf{L}\mathbf{L}^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)^{-1/2} && \text{Not Optimal} \\ \mathbf{A}\mathbf{X}_f &\approx \hat{\mathbf{A}}\mathbf{X}_f = \mathbf{X}_f - \hat{\mathbf{G}}\mathbf{Y} && \text{Not Optimal} \end{aligned}$$

The first approximation is near optimal via a sketched SVD, but the final error in the analysis ensemble is still only indirectly related to the error in the fundamental approximation.

This is more costly than modulated-ensemble GETKF because you have to do an (approximate) SVD at each new assimilation cycle, but also more accurate for a fixed rank of $\mathbf{L}$.

This talk introduces an ensemble adjustment Kalman filter with covariance localization:

Quad EAKF

"The EAKF that never skips leg day"

Quad EAKF **does not** use a low-rank approximation to the localized ensemble covariance.

We use the same adjustment formula as the ETKF/GETKF but make different approximations

$$\mathbf{A} = \left(\mathbf{I} + \mathbf{\Sigma}_{xx}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)^{-1/2} \approx \left(\mathbf{I} + (\mathbf{T} \circ \mathbf{B}_e)\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)^{-1/2} := \hat{\mathbf{A}}$$

How can you actually compute the update of the perturbations?

$$\mathbf{X}_a = \left(\mathbf{I} + (\mathbf{T} \circ \mathbf{B}_e)\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)^{-1/2}\mathbf{X}_f$$

Computing matrix square roots is not cheap or easy, especially for this matrix which has the same size as the ensemble covariance.

Steward et al. (2018) use Krylov subspace methods to get a reduced-dimension problem where they can compute/apply the matrix square root using direct SVD/eigenvalue methods.

Quad-EAKF begins by expressing the inverse square root as an integral of the form

$$x^{-1/2} = \int \frac{w(t)}{s(t) + x} \mathrm{d}t$$

There are many possible combinations of functions w and s that work here, e.g.

$$x^{-1/2} = \frac{2}{\pi} \int_0^\infty \frac{1}{t^2 + x} \mathrm{d}t.$$

We actually use a method based on contour integration in the complex plane.

Quad-EAKF approximation #1: Quadrature

$$x^{-1/2} = \int \frac{w(t)}{s(t) + x} \mathrm{d}t \approx \sum_{q=1}^{Q} \frac{w_q}{s_q + x}$$

Where t_q are the quadrature nodes, $s_q = s(t_q)$, and w_q are the quadrature weights (not necessarily $w_q = w(t_q)$).

The accuracy of this approximation depends on x, on w and s, on the quadrature scheme, and on the number of quadrature nodes Q. But it can be controlled and driven to zero if desired.

Applying this quadrature to the Quad-EAKF ensemble update yields

$$\left(\mathbf{I} + (\mathbf{T} \circ \mathbf{B}_e)\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)^{-1/2}\mathbf{X}_f \approx \sum_{q=1}^{Q} w_q \left((s_q+1)\mathbf{I} + \mathbf{B}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\mathbf{X}_f$$

So we can approximate the square root by solving Q linear systems where the coefficient matrix is positive-definite, but not symmetric.

Lack of symmetry is a problem. We can't afford direct methods.

The matrix is self-adjoint with respect to the $\mathbf{B}^{-1}$ inner product, but the corresponding CG algorithm requires applying $\mathbf{B}^{-1}$, which is prohibitive.

Re-label:

$$\mathbf{H} \leftarrow \mathbf{R}^{-1/2}\mathbf{H}$$

Apply Sherman-Morrison-Woodbury:

$$\left((s_q+1)\mathbf{I} + \mathbf{B}\mathbf{H}^T\mathbf{H}\right)^{-1} =$$

$$(s_q+1)^{-1}\mathbf{I} - (s_q+1)^{-2}\mathbf{B}\mathbf{H}^T\left(\mathbf{I} + (s_q+1)^{-1}\mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{H}$$

A few more simplifications leads to...

$$\left(\mathbf{I} + (\mathbf{T} \circ \mathbf{B}_e)\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)^{-1/2}\mathbf{X}_f \approx$$

$$\mathbf{X}_f - \mathbf{B}\mathbf{H}^T \sum_{q=1}^{Q} \frac{w_q}{s_q+1}\left((s_q+1)\mathbf{I} + \mathbf{H}\mathbf{B}\mathbf{H}^T\right)^{-1}\mathbf{Y}$$

This is an update using a modified gain, like the serial ESRF and the GETKF where the approximation to the modified Kalman gain is

$$\hat{\mathbf{G}} = \mathbf{B}\mathbf{H}^T \sum_{q=1}^{Q} \frac{w_q}{s_q+1}\left((s_q+1)\mathbf{I} + \mathbf{H}\mathbf{B}\mathbf{H}^T\right)^{-1}.$$

$$\hat{\mathbf{G}} = \mathbf{B}\mathbf{H}^T \sum_{q=1}^{Q} \frac{w_q}{s_q + 1} \left((s_q + 1)\mathbf{I} + \mathbf{H}\mathbf{B}\mathbf{H}^T \right)^{-1}.$$

For each ensemble member we need to solve Q linear systems where the coefficient matrix is symmetric and positive definite.

Use CG! But it gets better: The linear systems all have the same Krylov subspace, so the cost of solving all Q systems is only slightly more than the cost of solving one of them.

You can either do this in parallel for all ensemble members, or you can do all members simultaneously using block-CG.

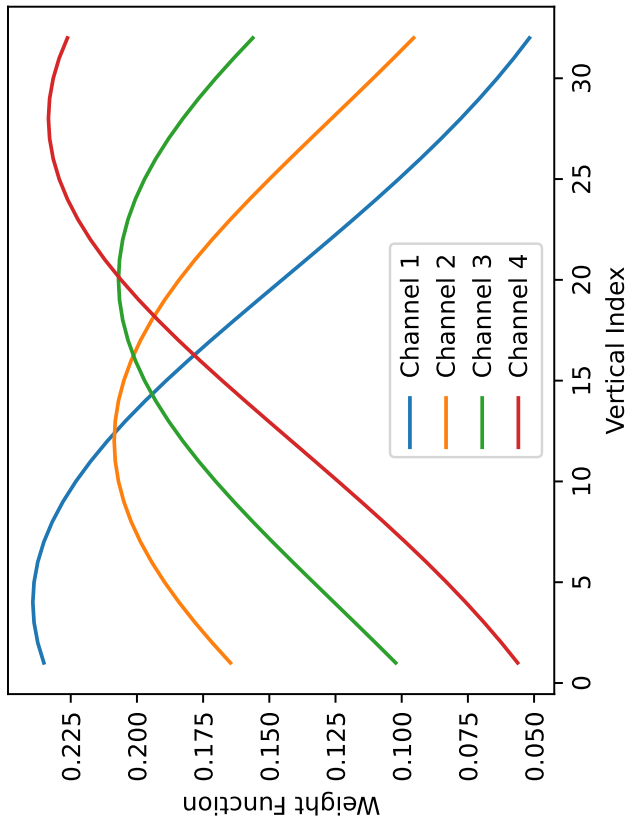
We use the efficient algorithm (noted by Farchi & Bocquet) to apply $\mathbf{B} = \mathbf{T} \circ \mathbf{B}_e$ to vectors.

Quad-EAKF Summary:

- 'From the left' ensemble update, i.e. adjustment; also equivalent to a modified-gain update
- **First approximation:** In the formula for the exact adjustment matrix, replace the exact covariance with the localized ensemble covariance
- **Second approximation:** Trade a single matrix square root for Q linear system solves using quadrature
- **Third approximation:** Solve iteratively using CG

We use the stacked Lorenz-96 model created by Farchi & Bocquet (2019): 32 layers, 40 variables each

We observe 8 columns (every 5^{th}). Each column produces 4 different weighted-average observations:

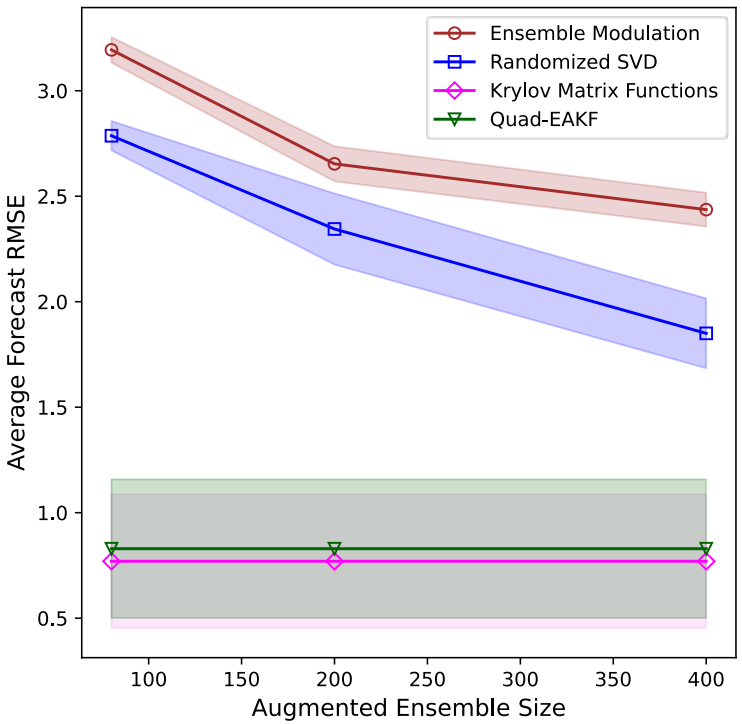


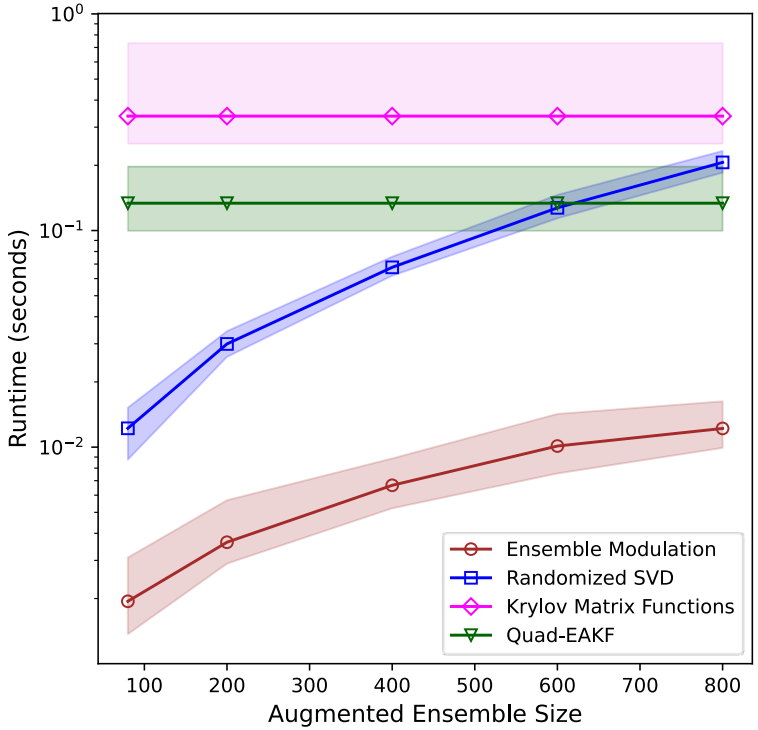
We compare

- GETKF with modulation following Bishop et al. (2017)
- Sketched SVD following Farchi & Bocquet (2019)
- Krylov square-root following Steward et al. (2018)
- Quad-EAKF: $Q = 20$ and 20 CG iterations

We use the same localization for all methods: Gaspari-Cohn with horizontal radius 10 and vertical radius 8.

Ensemble size 40 for all methods, not counting modulation. All methods use RTPS with 1% relaxation.


Ensemble Modulation
Randomized SVD
Krylov Matrix Functions
Quad-EAKF
Average Forecast RMSE
3.0
2.5
2.0
1.5
1.0
0.5
100
150
200
250
300
350
400
Augmented Ensemble Size


10^0
10^{-1}
10^{-2}
Runtime (seconds)
Ensemble Modulation
Randomized SVD
Krylov Matrix Functions
Quad-EAKF
100
200
300
400
500
600
700
800
Augmented Ensemble Size