0.1 Revised and generalized IEnKS, SDA pseudo-code

The below is the pseudo-code for the general form of the single data assimilation IEnKS(-N) that I utilize. Particularly, I make a few adjustments to either make steps abstract within the routine, to compact the algorithm, or to fix missing items in the pseudo-code from Bocquet and Sakov (2013, 2014):

- Note, we iterate the mean state at the end of the loop in line 29 of the below instead of at line 10 in the below. In Bocquet and Sakov (2013) Algorithm 1, this corresponds to removing line 5 and moving this step to line 16; in Bocquet and Sakov (2014) Algorithm 2, this corresponds to removing line 5 and moving this step to line 14. This is more consistent in the sense that we will use the final iterate for the mean state in line 44 below to compute the updated ensemble. Without this step, in the pseudo-code of Bocquet and Sakov (2013, 2014) Algorithms 1 and 2, we use the iterated $\mathbf{w}$ but not the iterated mean state. This inconsistency is corrected in the Python code you supplied, where there is a final iteration of the mean state outside of the while loop, maintaining consistency between the iterated mean state and weights. However, the proposed change below makes this more compact and also neglects an unnecessary step where we iterate the mean state with $\mathbf{w} = \mathbf{0}$ in the first iteration.

- We abstract the use of the ensemble conditioning transformation $\mathbf{T}$ in which it can equal either the transform version or simply $\epsilon \mathbf{I}_N$. In this way, $\mathbf{T}$ plays the same role throughout the algorithm, only differing in that we redefine $\mathbf{T}$ in every loop of the transform version and only after the loop terminates in the bundle version. For IEnKS-N versions, note that there is always the final revised Hessian approximation based on the adaptive inflation. This abstraction of the conditioning matrix $\mathbf{T}$ is actually used in DAPPER as well, it seems like Patrick and I both came to this conclusion independently.

- The "**else if** bundle" statement in line 41 arises to differentiate that, if we are using the transform, tuned inflation version, we have already computed $\mathbf{T} = \widetilde{\mathcal{H}}^{-\frac{1}{2}}$ within the loop on line 31 and this need not be re-computed. In the bundle form, we have to compute this as it was not computed in the last loop while loop.

- N_{eff} is not defined in the IEnKS-N implementations in either of Bocquet and Sakov (2013, 2014), yet I found this definition in the Python code you supplied. Without this, both IEnKS-N versions tend to be unstable and have sub-optimal performance in most configurations of ensemble size and lag length. With this included, the stability and performance has generally been enhanced and results have been reproduced from Bocquet and Sakov (2013, 2014). I tend to think this is a critical step, as both the tuned inflation and IEnKS-N schemes use most of the same code, with the minor exceptions indicated below, and I have little reason to think that I computed the gradient and Hessian approximations incorrectly otherwise.

- I also made note of the need to generate the filtered / forecast statistics, and that by the use of the conditioning matrix $\mathbf{T} = \mathbf{I}_N$ in the first iteration, this step can be combined in the first iterate of the transform version.

need plus sign in line 21 or 24, review make this converge to the EnKF in the L=0 case GMD hussaire 2016 atmospheric chemistry possibly use Gauss-Newton or LBFGS with primal version check with the model error correction

Algorithm 1 Gauss-Newton IEnKS(-N), lag L shift S, SDA general implementation

Require: t_L is present time. Transition model $\mathcal{M}_{k+1\leftarrow k}$, observation operators H_k at t_k. Algorithm parameters: ϵ, tol, j_{max} . $\mathbf{E}_0$, the ensemble at t_0, y_k the observation at t_k. λ is the inflation factor. $\mathbf{U}$ is an orthogonal matrix of size $N \times N$ satisfying $\mathbf{U}\mathbf{1} = \mathbf{1}$. Parameters for IEnKS-N include $\epsilon_N = 1 + \frac{1}{N}$, $N_{\text{eff}} = N + 1$.

if bundle **then**
    $\mathbf{T} = \epsilon \mathbf{I}_N$
else if transform **then**
    $\mathbf{T} = \mathbf{I}_N$
end if
$j = 0, \mathbf{w} = \mathbf{0}$
$\mathbf{x}_0^{(0)} = \mathbf{E}_0 \frac{\mathbf{1}}{N}$
$\mathbf{A}_0 = \mathbf{E}_0 - \mathbf{x}_0^{(0)} \mathbf{1}^\top$
while $j \leq j_{max}$ **do**
    $\mathbf{E}_0 = \mathbf{x}_0^{(j)} \mathbf{1}^\top + \mathbf{A}_0 \mathbf{T}$
    $\mathbf{E}_{L-S-1} = \mathcal{M}_{L-S-1\leftarrow 0}(\mathbf{E}_0)$
    for $s = 1, \cdots, S$ **do**
        $k = L - S + s$
        $\mathbf{E}_k = \mathcal{M}_{k\leftarrow k-1}(\mathbf{E}_{k-1})$
        $\overline{\mathbf{y}}_k = H_k(\mathbf{E}_k) \frac{\mathbf{1}}{N}$
        $\mathbf{Y}_k = \left(H(\mathbf{E}_k) - \overline{\mathbf{y}}_k \mathbf{1}^\top \right) \mathbf{T}^{-1}$
    end for
    if IEnKS-N **then**
        $\zeta = \frac{1}{\epsilon_N + \mathbf{w}^\top \mathbf{w}}$
        $\nabla \widetilde{\mathcal{J}} = \zeta \left(N_{\text{eff}} \right) \mathbf{w} - \sum_{k=L-S}^{L} \mathbf{Y}_k^\top \mathbf{R}_k^{-1} \left(\mathbf{y}_k - \overline{\mathbf{y}}_k \right)$
        $\widetilde{\mathcal{H}} = (N_{\text{eff}} - 1) \mathbf{I}_N \sum_{k=L-S}^{L} \mathbf{Y}_k^\top \mathbf{R}_k^{-1} \mathbf{Y}_k$
    else
        $\nabla \widetilde{\mathcal{J}} = (N - 1) \mathbf{w} - \sum_{k=L-S}^{L} \mathbf{Y}_k^\top \mathbf{R}_k^{-1} \left(\mathbf{y}_k - \overline{\mathbf{y}}_k \right)$
        $\widetilde{\mathcal{H}} = (N - 1) \mathbf{I}_N \sum_{k=L-S}^{L} \mathbf{Y}_k^\top \mathbf{R}_k^{-1} \mathbf{Y}_k$
    end if
    Solve: $\widetilde{\mathcal{H}} \Delta \mathbf{w} = \nabla \widetilde{\mathcal{J}}$
    $\mathbf{w} := \mathbf{w} - \Delta \mathbf{w}$
    $j := j + 1$
    $\mathbf{x}_0^{(j)} = \mathbf{x}_0^{(0)} + \mathbf{A}_0 \mathbf{w}$
    if transform **then**
        $\mathbf{T} = \widetilde{\mathcal{H}}^{-\frac{1}{2}}$
    end if
    if $\parallel \Delta \mathbf{w} \parallel < tol$ **then**
        break while
    end if
end while
if IEnKS-N **then**
    $\zeta = \frac{1}{\epsilon_N + \mathbf{w}^\top \mathbf{w}}$
    $\widetilde{\mathcal{H}} = N_{\text{eff}} \left(\zeta \mathbf{I}_N - 2\zeta^2 \mathbf{w}\mathbf{w}^\top \right) + \sum_{k=L-S}^{L} \mathbf{Y}_k^\top \mathbf{R}_k^{-1} \mathbf{Y}_k$
    $\mathbf{T} = \widetilde{\mathcal{H}}^{-\frac{1}{2}}$
else if bundle **then**
    $\mathbf{T} = \widetilde{\mathcal{H}}^{-\frac{1}{2}}$
end if
$\mathbf{E}_0 = \mathbf{x}_0^{(j)} \mathbf{1}^\top + \sqrt{N-1} \mathbf{A}_0 \mathbf{T} \mathbf{U}$
$\mathbf{E}_S = \mathcal{M}_{S\leftarrow 0}(\mathbf{E}_0)$
Generate filtered / forecast ensemble (can be combined with first iteration of transform version): $\mathbf{E}_{S+L} = \mathcal{M}_{L+S\leftarrow S}(\mathbf{E}_S)$
$\mathbf{x}_S = \mathbf{E}_S \frac{\mathbf{1}}{N}$
$\mathbf{E}_S = \mathbf{x}_S \mathbf{1}^\top + \lambda \left(\mathbf{E}_S - \mathbf{x}_S \mathbf{1}^\top \right)$

0.2 Figures of recent results and projected experiments

I'm re-running some experiments so that the schemes are more fully consistent with updated versions of the algorithms. Once all schemes have been given a final test (most likely in the next 1-2 weeks), I'll prepare a final publishable version of the code to run all experiments with statistical significance. I think the most direct comparison between all methods is using roughly the same version of the ensemble transform and update steps, but to vary: (i) the place in the DAW where the update is performed; and (ii) if there is a single iteration, (possibly) multiple iterations or no iterations. I think it is logical in this sense to compare the IEnKS transform version, the ETKS classic version and the the single-iteration ensemble transform Kalman smoother (SIETKS), over configurations in which inflation is tuned or adaptive, and / or the method uses single data assimilation or multiple data assimilation. The Lin-IEnKS may be relevant as well. As I understand, this is basically using IEnKS with a single iteration, and with the transform version, would basically run at the cost of the SIETKS. However, one big difference is that the EnKF-N optimization can be utilized in the SIETKS with no additional ensemble forecasts, contrary to the Lin-IEnKS.

The current state of work is as follows. I have just implemented the primal version of the EnKF-N analysis, using Newton with a strong Wolfe condition linesearch, using Julia optimization libraries. The primal EnKF-N update is most directly comparable with the IEnKS-N in its implementation, so that the primal version will be used to generate adaptive inflation in the ETKS classic and SIETKS in the final runs. Currently pictured results for the EnKS-N and SIETKS-N use the dual version, and the primal version is pending a formal validation (though test cases are extremely good, with notable improvements using the Wolfe condition linesearch over other methods). I have also just finished developing the IEnKS-MDA scheme for comparison. This also appears to be working well, but this involved an overall refactoring of all the Gauss-Newton IEnKS method, and the new structure should be formally benchmarked for SDA as well. I have also tried to improve the stability and accuracy of the IEnKS transform version by using the stable, SVD-based square-root-inverse rather than the built-in Julia version. This is currently being benchmarked for an overall improvement on the earlier transform version.

For the final comparison between methods, I am considering the following array of experiments. Experimental configurations that have been validated are denoted with a V. Configurations that are being re-validated with improvements / optimization are denoted with R. Configurations that are pre validation are denoted with a P. Experiments that I'm decided whether to run or not are denoted with a D. My biggest concern is with the MDA adaptive inflation using all weights equal to one. Estimating a power of the posterior is indeed something fishy, and I am unsure that I want to use these particular experiments in a final comparison.

Configuration / Method	ETKS-Classic	SIETKS SDA	SIETKS MDA	IEnKS SDA	IEnKS MDA
tuned inflation, $\Delta_t = 0.05$	V	V	V	R	P
tuned inflation, $\Delta_t = 0.10$	V	V	V	R	P
adaptive inflation, $\Delta_t = 0.05$	R	R	D	R	D
adaptive inflation, $\Delta_t = 0.10$	R	R	D	R	D

Below is the data that was available up to yesterday. Unfortunately, the power went out on my server today and I couldn't process the up-to-date results as I had planned until last minute, and various runs are incomplete. Some of the following figures are missing pieces of data for the reasons discussed above. I am only presenting the $\Delta_t = 0.05$ data as this is generally more complete at the moment. Despite the incomplete results, I feel the results presented are very good and worth demonstrating in their incomplete form.

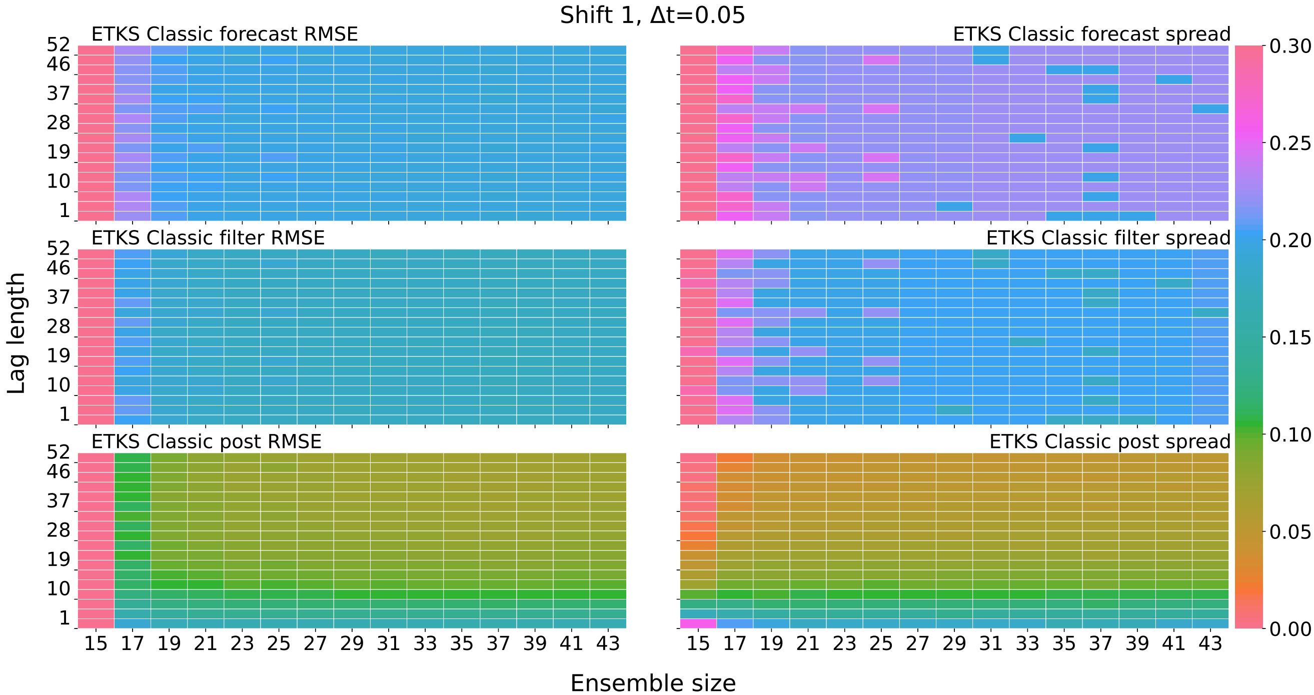

Figure 1. ETKS RMSE and spread averaged over 2×10^4 analyses, with an additional, initial 5×10^3 analyses discarded as spin-up. Inflation choice is optimized for lowest smoother RMSE over $\lambda\in\{1.00+i\times0.01\}_{i=0}^{10}$. Single data assimilation, $\Delta t=0.05$.

References

Bocquet, M. and Sakov, P.: Joint state and parameter estimation with an iterative ensemble Kalman smoother, Nonlinear Processes in Geophysics, 20, 803–818, 2013.

Bocquet, M. and Sakov, P.: An iterative ensemble Kalman smoother, Quarterly Journal of the Royal Meteorological Society, 140, 1521–1535, 2014.

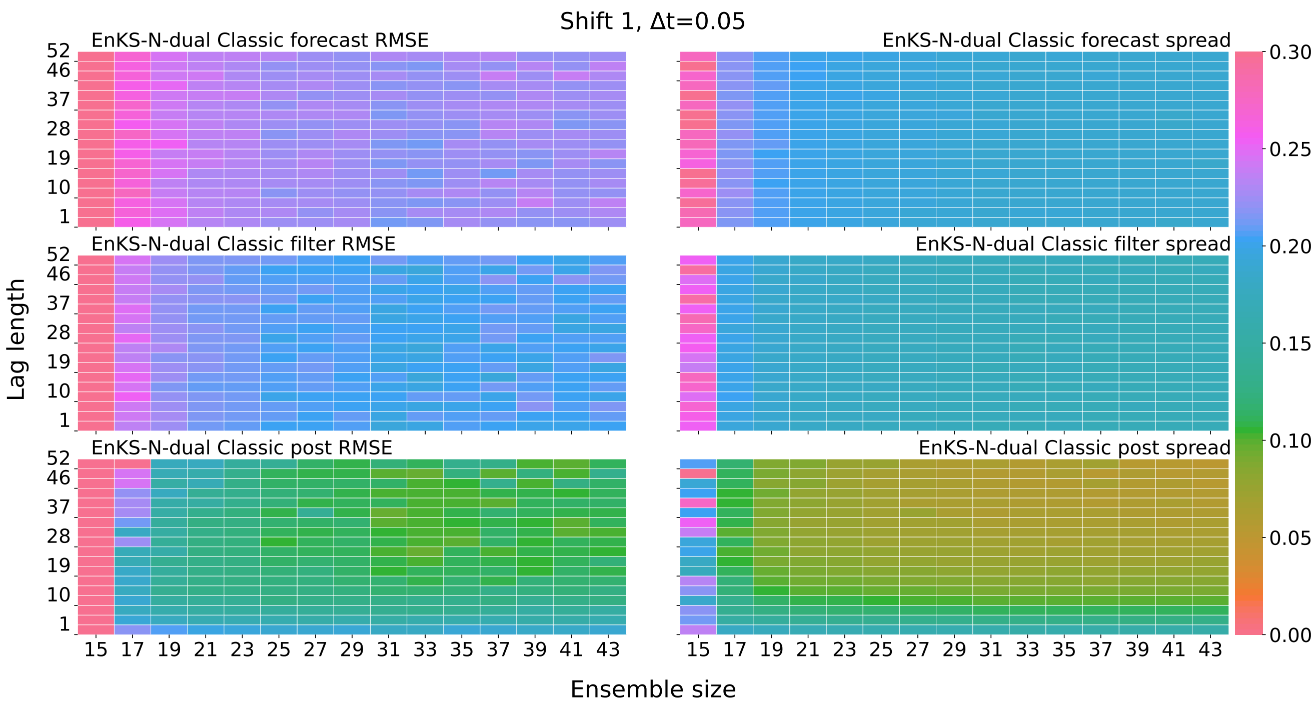

Figure 2. EnKS-N RMSE and spread averaged over 2×10^4 analyses, with an additional, initial 5×10^3 analyses discarded as spin-up. Adaptive inflation with finite-size formalism. Single data assimilation, $\Delta t = 0.05$.

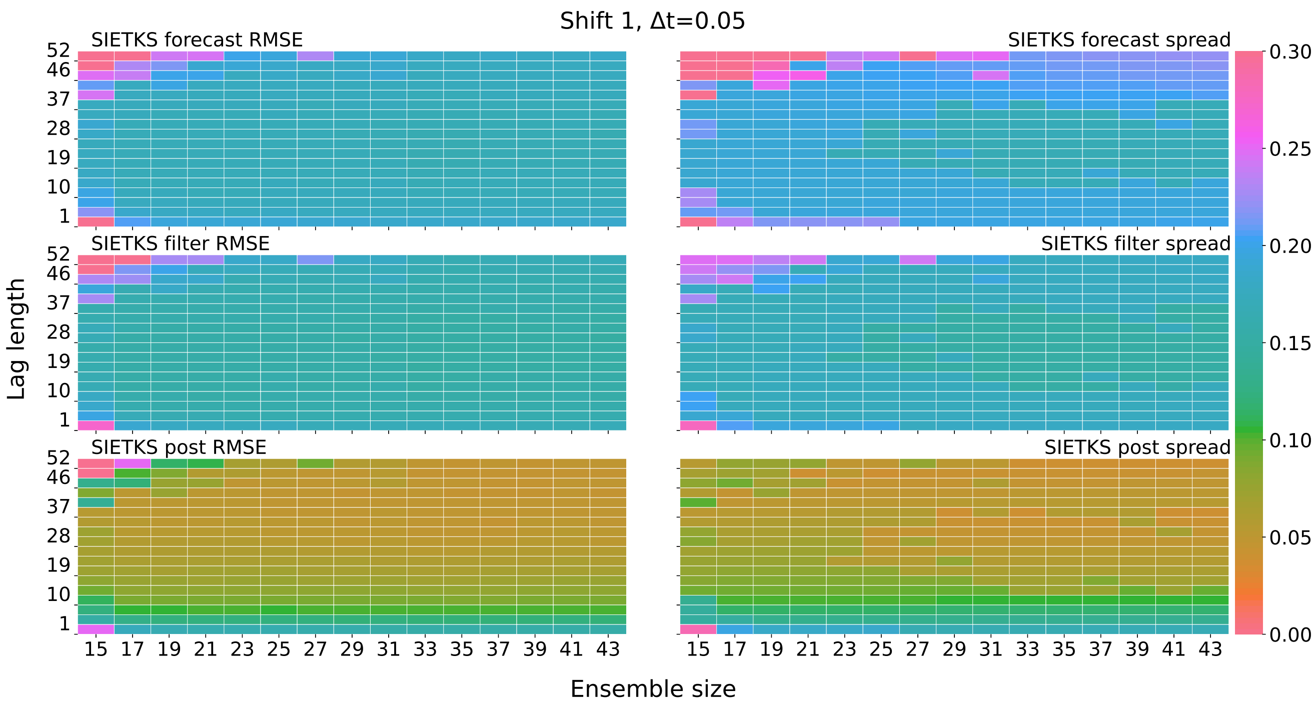

Figure 3. SIETKS RMSE and spread averaged over 2×10^4 analyses, with an additional, initial 5×10^3 analyses discarded as spin-up. Inflation choice is optimized for lowest smoother RMSE over $\lambda \in \{1.00 + i \times 0.01\}_{i=0}^{10}$. Single data assimilation, $\Delta t = 0.05$.

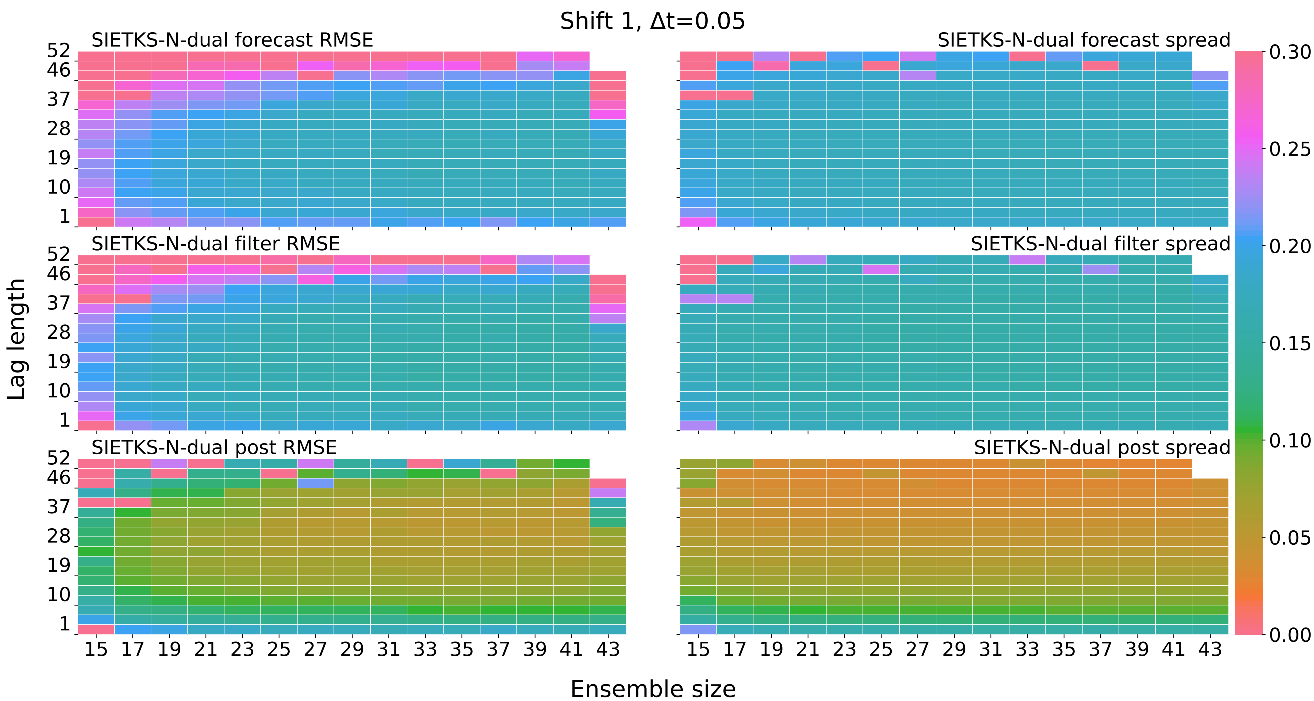

Figure 4. SIETKS RMSE and spread averaged over 2×10^4 analyses, with an additional, initial 5×10^3 analyses discarded as spin-up. Adaptive inflation with finite-size formalism. Single data assimilation, $\Delta t = 0.05$.

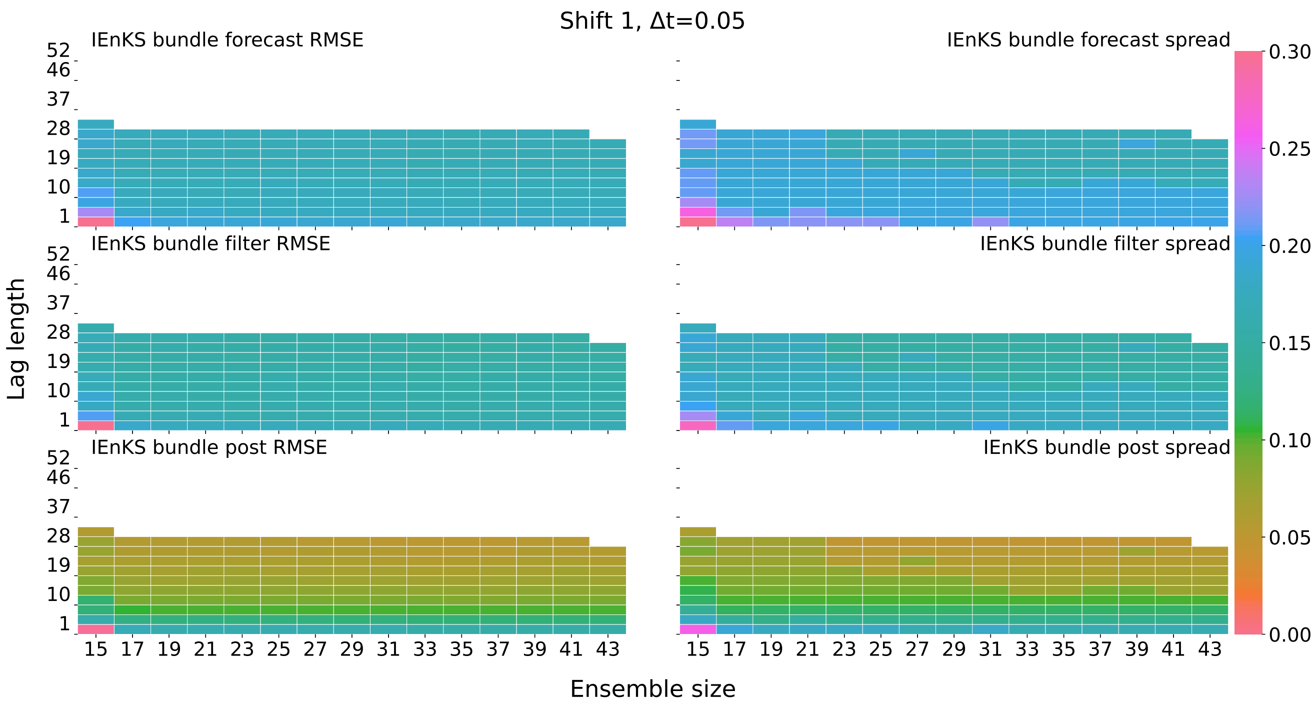

Figure 5. IEnKS bundle RMSE and spread averaged over 2×10^4 analyses, with an additional, initial 5×10^3 analyses discarded as spin-up. Inflation choice is optimized for lowest smoother RMSE over $\lambda \in \{1.00 + i \times 0.01\}_{i=0}^{10}$. Single data assimilation, $\Delta t = 0.05$.

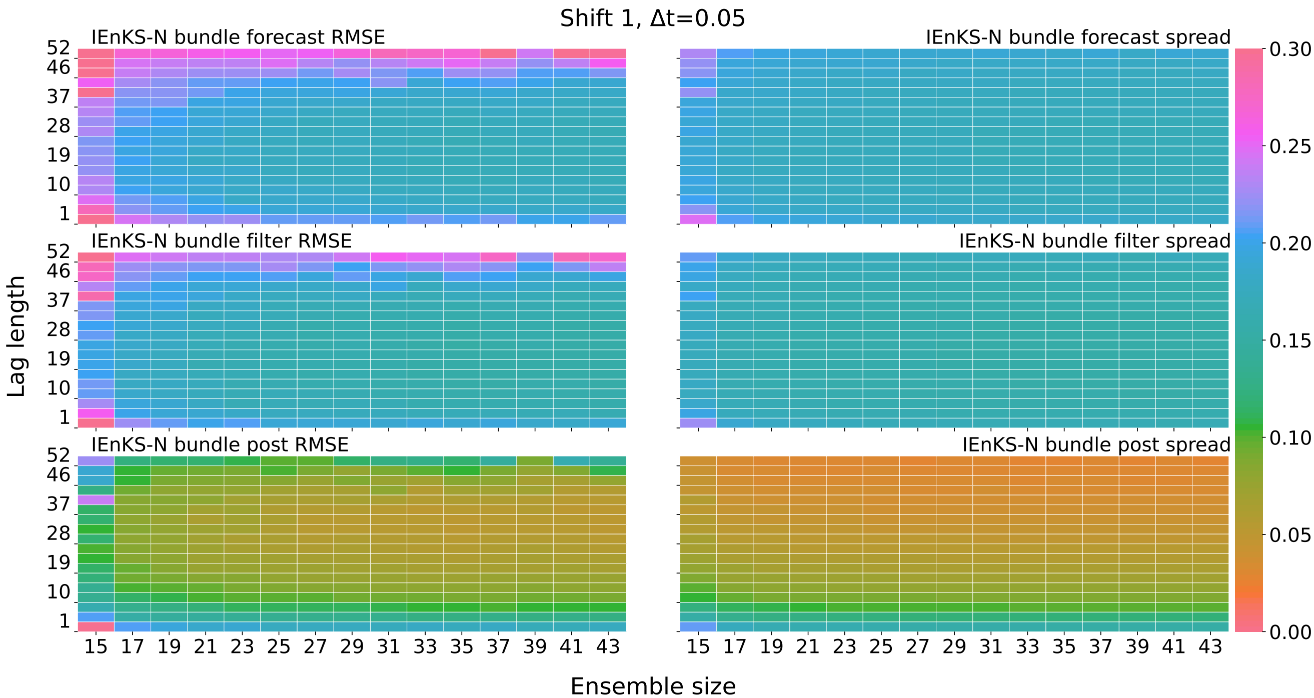

Figure 6. IEnKS-N bundle RMSE and spread averaged over 2×10^4 analyses, with an additional, initial 5×10^3 analyses discarded as spin-up. Adaptive inflation with finite-size formalism. Single data assimilation, $\Delta t = 0.05$.

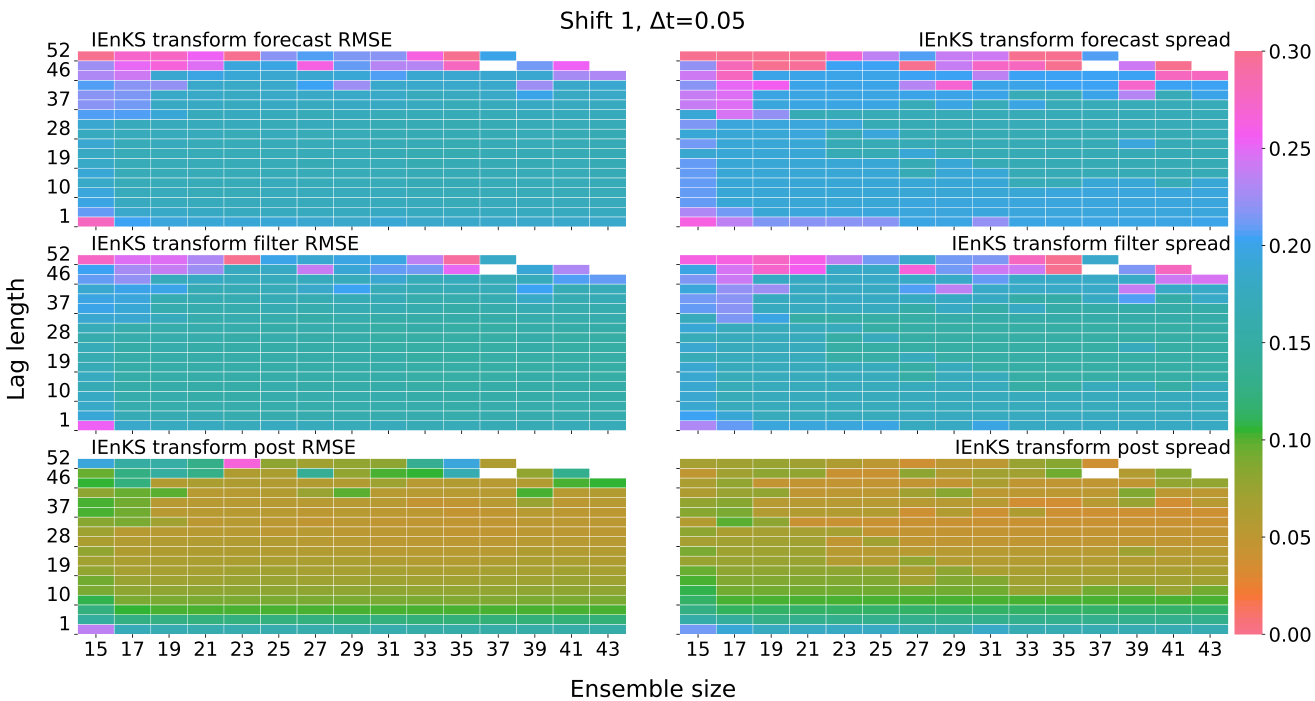

Figure 7. IEnKS transform RMSE and spread averaged over 2×10^4 analyses, with an additional, initial 5×10^3 analyses discarded as spin-up. Inflation choice is optimized for lowest smoother RMSE over $\lambda \in \{1.00 + i \times 0.01\}_{i=0}^{10}$. Single data assimilation, $\Delta t = 0.05$.

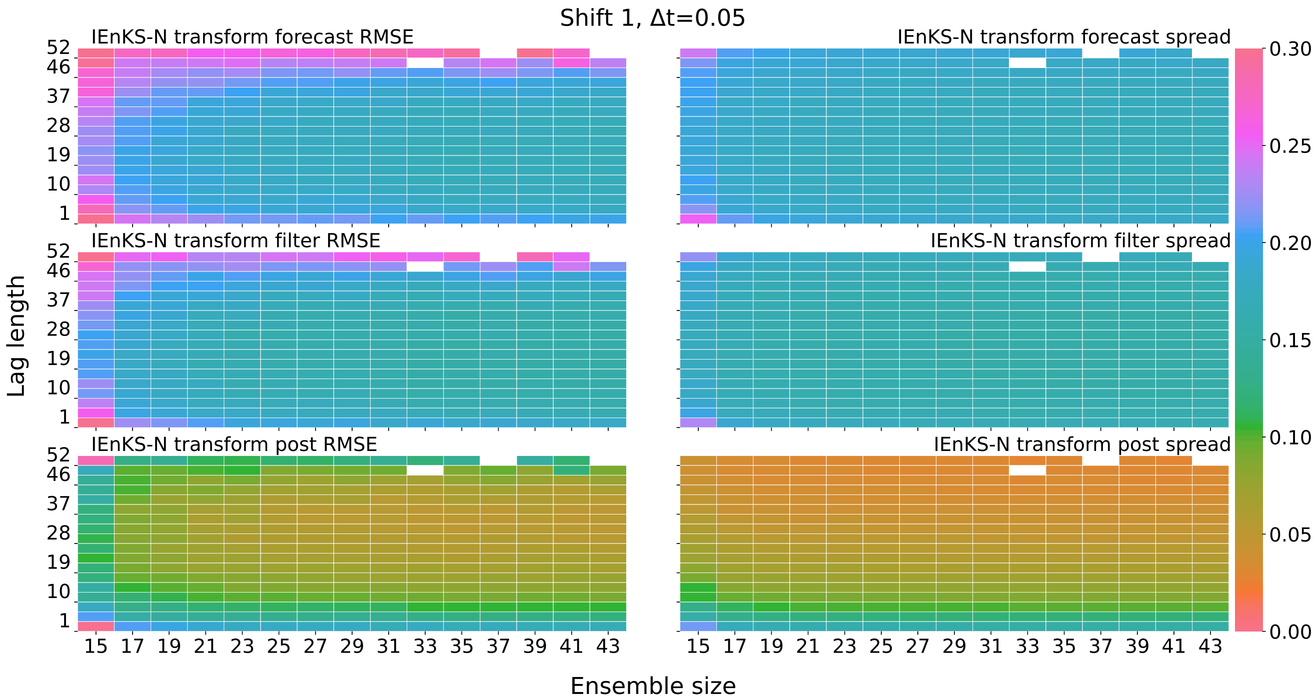

Figure 8. IEnKS-N transform RMSE and spread averaged over 2×10^4 analyses, with an additional, initial 5×10^3 analyses discarded as spin-up. Adaptive inflation with finite-size formalism. Single data assimilation, $\Delta t = 0.05$.

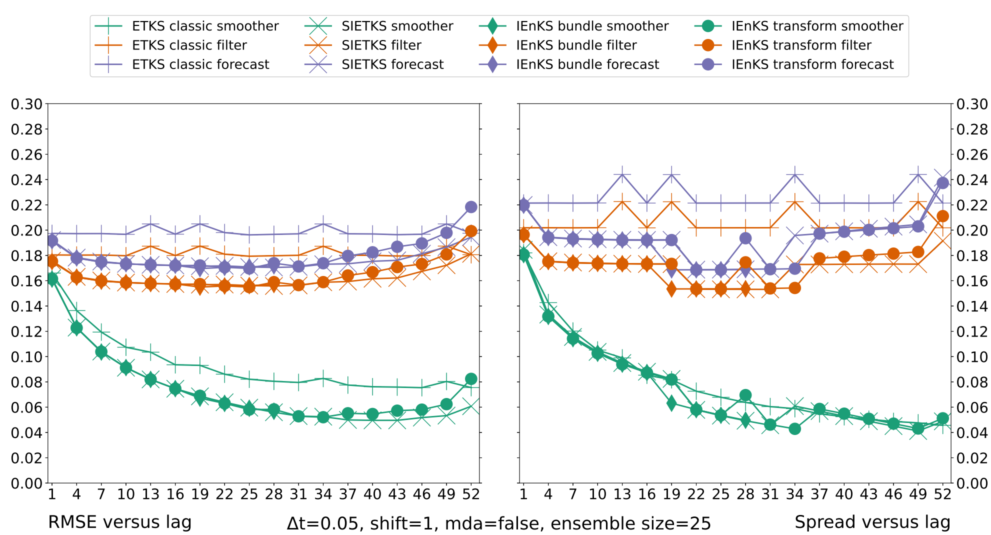

Figure 9. RMSE and spread averaged over 2×10^4 analyses, with an additional, initial 5×10^3 analyses discarded as spin-up. Ensemble size fixed at $N = 19$, lag length varied. Inflation choice is optimized for lowest smoother RMSE over $\lambda \in \{1.00 + i \times 0.01\}_{i=0}^{10}$. Single data assimilation, $\Delta t = 0.05$.

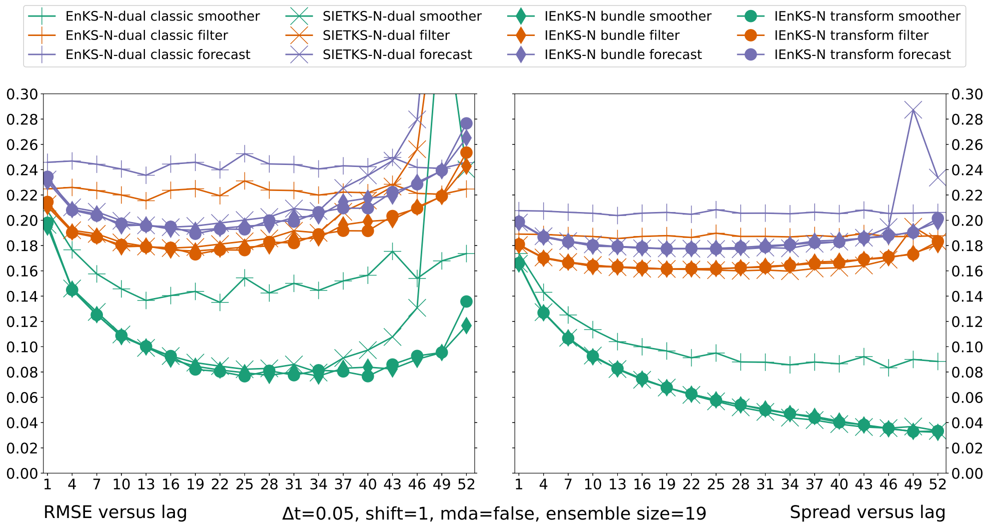

Figure 10. RMSE and spread averaged over 2×10^4 analyses, with an additional, initial 5×10^3 analyses discarded as spin-up. Ensemble size fixed at $N = 19$, lag length varied. Adaptive inflation with finite-size formalism. Single data assimilation, $\Delta t = 0.05$.

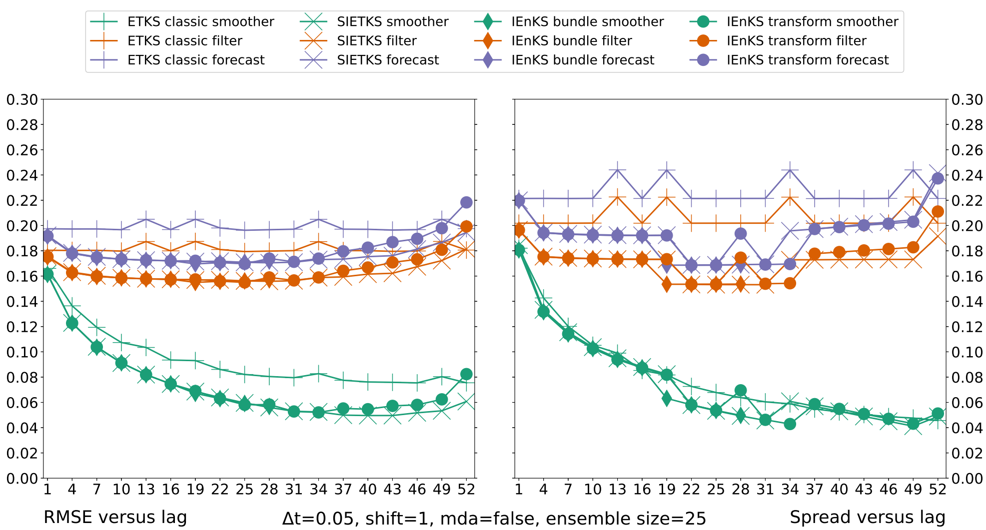

Figure 11. RMSE and spread averaged over 2×10^4 analyses, with an additional, initial 5×10^3 analyses discarded as spin-up. Ensemble size fixed at $N = 25$, lag length varied. Inflation choice is optimized for lowest smoother RMSE over $\lambda \in \{1.00 + i \times 0.01\}_{i=0}^{10}$. Single data assimilation, $\Delta t = 0.05$.

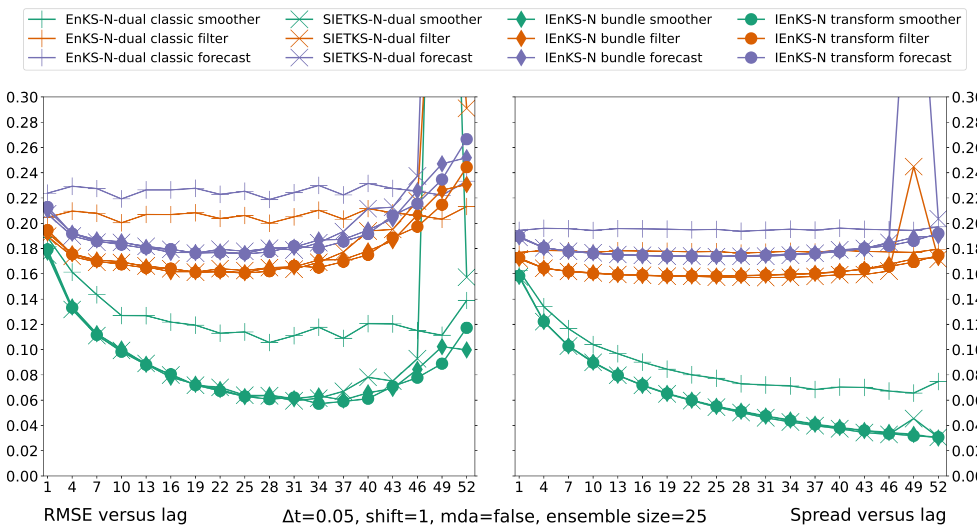

Figure 12. RMSE and spread averaged over 2×10^4 analyses, with an additional, initial 5×10^3 analyses discarded as spin-up. Ensemble size fixed at $N = 25$, lag length varied. Adaptive inflation with finite-size formalism. Single data assimilation, $\Delta t = 0.05$.