Climate Model Tuning Without Hyperparameters

Nikki Lydeen[1], Timothy DelSole[1], Benjamin Cash[1]

[1]George Mason University, Fairfax, VA, USA

Key Points:

- Earth system models (ESMs) have many tunable parameters that are difficult to estimate and weakly constrained by theory.
- Kalman filter-based approaches are attractive options, but existing implementations require expensive offline hyperparameter selection.
- We propose a new Kalman filter algorithm that estimates model parameters and its hyperparameter simultaneously.

Corresponding author: Nikki Lydeen, nlydeen@gmu.edu

Abstract

This article introduces a new algorithm, *KalmRidge*, and demonstrates its ability to tune an Earth system model (ESM) using idealized experiments. Unlike similar algorithms, KalmRidge eliminates the need for offline hyperparameter selection, thereby substantially reducing computational expense. This is done by rewriting the update equations for the ensemble Kalman filter as an equivalent ridge regression problem, then applying standard cross-validation techniques to adaptively choose the regularization parameter. We propose that this algorithm, with time-mean spherical harmonic projections as tuning targets, provides a promising, tractable approach for parameter estimation.

Plain Language Summary

Earth system models (ESMs) depend on parameters that are difficult to estimate. Although these parameters are routinely estimated during model development, there is no standard approach for parameter estimation, and existing algorithms are expensive to apply. Contemporary algorithms have hyperparameters, i.e. parameters of the algorithms themselves rather than of the model, which also must be estimated. Estimating hyperparameters is very computationally expensive. This article introduces a new algorithm which eliminates the need for hyperparameter estimation, and demonstrates its application to a state-of-the-art ESM.

1 Introduction

Earth system models (ESMs) simulate atmospheric, oceanic, and terrestrial processes and their interactions on a global scale. Some of the governing equations, particularly the equations of large-scale motion, are derived from well-understood physical principles and may be discretized for numerical integration. In contrast, many important small-scale processes, such as those associated with clouds, cannot be explicitly resolved at feasible grid resolutions. Instead, the grid-scale aggregate effects of these processes are estimated using semi-empirical functions, called parameterizations, which depend on parameters that are weakly constrained by observations. Tuning is challenging because ESM integrations are computationally expensive, and it requires running many ESM integrations with different parameter values. Tuning is further complicated by interactions between processes, which may cause biases in parameter estimates due to compensating errors.

Hourdin et al. (2017) and Schmidt et al. (2017) documented contemporary tuning practices at several climate modeling centers. They found that different centers use different strategies and have different goals. In practice, these modeling centers use traditional, manual trial-and-error tuning, rather than automated tuning algorithms.

Estimating parameters in a dynamical model is a classic example of an inverse problem. An early geophysical application of this approach was introduced by Carrera and Neuman (1986) in the context of groundwater flow models. For problems where distributions are Gaussian, many inverse problems reduce to the Kalman filter. The Kalman filter is one of the most extensively studied recursive state estimation algorithms. It gives the best linear unbiased estimate (BLUE) under the assumptions that the forecast model is linear and the prior and observational errors are normally distributed (Kalman, 1960). Although the Kalman filter was developed originally as a method for estimating the state of system, it can be used to estimate both state and model parameters simply by augmenting the state vector with a parameter vector, treating the augmented parameters as unobserved state vectors. This methodology has been extensively applied in weather data assimilation (Annan, Lunt, Hargreaves, and Valdes (2005), Yang and DelSole (2009), Hu, Zhang, and Nielsen-Gammon (2010), and Koyama and Watanabe (2010)).

When estimating the parameters of an ESM, the usual focus is on the statistical moments on the model rather than the transient evolution. Various approaches have been proposed to address this, ranging from stochastic PDF estimation techniques, which include algorithms like Metropolis-Hastings and multiple iterations of very fast simulated annealing (Jackson et al., 2004), to methods that optimize explicit objective functions, such as the downhill simplex algorithm (Severijns & Hazeleger, 2005; Zhang et al., 2018). In this paper, we focus on Kalman filter-based methods. Within the context of climate model tuning, these approaches are often referred to as Kalman inversion methods (Iglesias et al., 2013; Schneider et al., 2017).

Many Kalman filter-based methods for tuning models incorporate at least one free hyperparameter that must be manually specified or estimated. For example, the ensemble Kalman filter often exhibits ensemble collapse, which is typically addressed through techniques like covariance inflation and localization– both requiring the tuning of hyperparameters. A hyperparameter is a parameter of the algorithm itself, rather than of the model being tuned. Free hyperparameters require user input for specification, unlike those that are hard-coded or adaptively estimated by the algorithm. Methods for online estimation of such hyperparameters, like the adaptive inflation algorithm (Anderson (2007)) and the adaptive localization algorithm (Bishop and Hodyss (2009)), have been developed. However, these too involve secondary hyperparameters. Unfortunately, there is often limited guidance on setting these values for applications far removed from the algorithm's initial use, requiring multiple algorithm runs to determine optimal settings, thereby increasing the practical cost of model tuning.

The purpose of this paper is to introduce an algorithm devoid of free hyperparameters. In the context of an ensemble Kalman filter, free hyperparameters typically arise either through covariance inflation or through observation error covariances. Our basic idea is to re-formulate the ensemble Kalman filter as an equivalent ridge regression problem, where the ridge parameter is identified with the hyperparameter. Then standard cross-validation techniques are used to select the hyperparameter adaptively. This approach estimates the hyperparameter separately and independently at each iteration of the filter, eliminating the need for presetting an initial value or allowing for a period of adaptation. Consequently, the algorithm provides immediate estimates right from the first iteration, continuing reliably in subsequent iterations.

In the next section, we review Kalman inversion and the associated ensemble version. Our proposed algorithm is then discussed in detail in Section 3. To place this algorithm into context, we compare it to various alternatives discussed in Section 4. One such alternative is an unregularized version that applies the Kalman filter without hyperparameters. In this scenario, ensemble methods are used to derive approximate covariances for the prior, and the observation error covariance matrix is calculated by using the sample covariance matrix from a long dataset. Another alternative involves using consistency diagnostics to estimate hyperparameters. For instance, Desroziers, Berre, Chapnik, and Poli (2005) show that the covariances of the innovations satisfy certain consistency constraints, a fact that has been used subsequently to develop online estimation algorithms of hyperparameters (Li et al., 2009). Another is to transform the problem of regularizing the observation error covariance into a problem of regularizing the prior covariance matrix, and then apply algorithms designed for covariance inflation, like those by Anderson (2007) and further refined by El Gharamti (2018). Additionally, we consider an algorithm proposed by Iglesias and Yang (2021) that adaptively minimizes discrepancies from observations and includes a criterion for early stopping in hyperparameter estimation. The performance of these algorithms is evaluated on both the Lorenz 96 model as a toy example and an Earth System Model in Sections 5 and 6, respectively. The paper concludes with a summary and discussion of our findings.

2 Kalman Inversion

The Kalman filter estimates the state $\boldsymbol{x}$ from two pieces of information: imperfect observations $\boldsymbol{o}$ of $\boldsymbol{x}$, and a prior distribution of $\boldsymbol{x}$. The state and observations are assumed to be related according to

$$\boldsymbol{o} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{\epsilon}$ is a random vector sampled from a multivariate normal distribution with zero mean and covariance matrix $\boldsymbol{R}$, which we denote as

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}). \tag{2}$$

The prior distribution is $\mathcal{N}(\overline{\boldsymbol{x}}_B, \boldsymbol{\Sigma}_B)$. Then, the Kalman filter updates $\boldsymbol{x}$ according to Bayes' theorem, yielding the normal distribution

$$\boldsymbol{x}_A \sim \mathcal{N}(\overline{\boldsymbol{x}}_A, \boldsymbol{\Sigma}_A), \tag{3}$$

where

$$\overline{\boldsymbol{x}}_A = \overline{\boldsymbol{x}}_B + \boldsymbol{\Sigma}_B\boldsymbol{H}^\top \left(\boldsymbol{H}\boldsymbol{\Sigma}_B\boldsymbol{H}^\top + \boldsymbol{R}\right)^{-1} \left(\boldsymbol{o} - \boldsymbol{H}\overline{\boldsymbol{x}}_B\right), \tag{4}$$

and

$$\boldsymbol{\Sigma}_A = \boldsymbol{\Sigma}_B - \boldsymbol{\Sigma}_B\boldsymbol{H}^\top \left(\boldsymbol{H}\boldsymbol{\Sigma}_B\boldsymbol{H}^\top + \boldsymbol{R}\right)^{-1} \boldsymbol{H}\boldsymbol{\Sigma}_B. \tag{5}$$

Equations (4) and (5) are known as the *Kalman filter update equations.*

Following Schneider et al. (2017), we estimate parameters $\boldsymbol{\theta} \in \mathcal{R}^{P\times 1}$ using statistical moments $\boldsymbol{m} \in \mathcal{R}^{S\times 1}$ computed from model integrations. Accordingly, the augmented state vector is

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{m} \\ \boldsymbol{\theta} \end{pmatrix} \in \mathcal{R}^{(S+P)\times 1}. \tag{6}$$

Using statistical moments or other quantities instead of instantaneous states to estimate parameters is often called *Kalman inversion* (Iglesias et al., 2013). Since only the moments contained in $\boldsymbol{x}$ are observable, $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{m}$, hence

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{I}_{S\times S} & \boldsymbol{0}_{S\times P} \end{pmatrix}. \tag{7}$$

In practice, we estimate the prior distribution from a finite ensemble. If we have an ensemble of size E with members $\boldsymbol{x_1}, \ldots, \boldsymbol{x_E}$, then

$$\hat{\boldsymbol{\Sigma}}_B = \boldsymbol{F}_B\boldsymbol{F}_B^\top \tag{8}$$

gives an unbiased estimate of $\boldsymbol{\Sigma}_B$, where

$$\boldsymbol{F}_B = \frac{1}{\sqrt{E-1}} \begin{pmatrix} \boldsymbol{x}_1 - \hat{\overline{\boldsymbol{x}}}_B & \cdots & \boldsymbol{x}_E - \hat{\overline{\boldsymbol{x}}}_B \end{pmatrix} \quad \text{and} \quad \hat{\overline{\boldsymbol{x}}}_B = \frac{1}{E}\sum_{i=1}^{E} \boldsymbol{x}_i. \tag{9}$$

Substituting $\hat{\boldsymbol{\Sigma}}_B$ for $\boldsymbol{\Sigma}_B$ in the Kalman filter update equations is the starting point for the ensemble Kalman filter (EnKF) update equations. Like the KF, the EnKF is an iterative algorithm; the updated ensemble becomes the initial ensemble at the next iteration, and new ESM integrations are computed using the new parameter estimates.

Although the above Kalman filter estimates both the statistical moments and the parameters, only the parameter estimates are retained. Thus, at the next iteration, the initial ensemble utilizes only the ensemble of parameter estimates from the previous iteration, whereas the moments are discarded since the dynamical model recomputes them at the next iteration. For chaotic models, the state used for the initial condition at the next iteration is arbitrary since, ideally, the model is run long enough to yield moments that are independent of the initial state.

3 Derivation of KalmRidge

Applying the update equations requires specifying the noise covariance matrix $\boldsymbol{R}$. In many applications, assumptions about the structure of $\boldsymbol{R}$ can reduce this problem to the selection of a small number of parameters (henceforth "hyperparameters", to prevent confusion with ESM parameters). Schneider et al. (2017) selected $\boldsymbol{R}$ by multiplying a diagonal matrix of climatological variances (henceforth $\boldsymbol{Q}$) by the square of a hyperparameter r:

$$\boldsymbol{R} = r^2\boldsymbol{Q} \tag{10}$$

They found that their parameter estimates depend on r, but they did not propose a strategy to select its value. Instead, they re-executed their algorithm several times, with different values of r for comparison. This process substantially increases the computational expense, and it is unclear how one would choose the best value of r without prior knowledge of the true parameter values.

We now show that the Kalman filter update equations are equivalent to the solution of a ridge regression problem, with r^2 serving as the ridge parameter λ. This equivalence naturally suggests that λ can be estimated using cross-validation techniques commonly used in standard ridge regression problems. To show this, let $\lambda = r^2$, so that we can rewrite Equation 4 as

$$\begin{aligned}\overline{\boldsymbol{x}}_A &= \overline{\boldsymbol{x}}_B + \boldsymbol{\Sigma}_B\boldsymbol{H}^\top\left(\boldsymbol{H}\boldsymbol{\Sigma}_B\boldsymbol{H}^\top + \lambda\boldsymbol{Q}\right)^{-1}(\boldsymbol{o} - \boldsymbol{H}\overline{\boldsymbol{x}}_B)\\ &= \overline{\boldsymbol{x}}_B + \boldsymbol{F}_B\boldsymbol{F}_B^\top\boldsymbol{H}^\top\left(\boldsymbol{H}\boldsymbol{F}_B\boldsymbol{F}_B^\top\boldsymbol{H}^\top + \lambda\boldsymbol{Q}\right)^{-1}(\boldsymbol{o} - \boldsymbol{H}\overline{\boldsymbol{x}}_B)\,.\end{aligned}$$

Then, applying the matrix identity

$$\boldsymbol{U}^\top\left(\boldsymbol{A} + \boldsymbol{U}\boldsymbol{U}^\top\right)^{-1} = \left(\boldsymbol{I} + \boldsymbol{U}^\top\boldsymbol{A}^{-1}\boldsymbol{U}\right)^{-1}\boldsymbol{U}^\top\boldsymbol{A}^{-1} \tag{11}$$

with the identifications $\boldsymbol{U} = \boldsymbol{H}\boldsymbol{F}_B$ and $\boldsymbol{A} = \lambda\boldsymbol{Q}$ gives

$$\overline{\boldsymbol{x}}_A = \overline{\boldsymbol{x}}_B + \boldsymbol{F}_B\left(\lambda\boldsymbol{I} + \boldsymbol{F}_B^\top\boldsymbol{H}^\top\boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{F}_B\right)\boldsymbol{F}_B^\top\boldsymbol{H}^\top\boldsymbol{Q}^{-1}(\boldsymbol{o} - \boldsymbol{H}\overline{\boldsymbol{x}}_B). \tag{12}$$

We may write this in a more recognizable form by defining

$$\boldsymbol{X} = \boldsymbol{Q}^{-1/2}\boldsymbol{H}\boldsymbol{F}_B \tag{13}$$

and

$$\boldsymbol{y} = \boldsymbol{Q}^{-1/2}(\boldsymbol{o} - \boldsymbol{H}\overline{\boldsymbol{x}}_B), \tag{14}$$

in which case Equation 4 becomes

$$\overline{\boldsymbol{x}}_A = \overline{\boldsymbol{x}}_B + \boldsymbol{F}_B\hat{\boldsymbol{\beta}} \tag{15}$$

where

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}^\top\boldsymbol{y}. \tag{16}$$

Equation 16 is identical to the equation of the ridge regression estimator (Hastie et al., 2009). Hence, we propose selecting λ using standard k-fold cross-validation methods in that context. This involves dividing the X and y into k equal parts, or "folds". For each fold, the model is trained on $k-1$ folds and tested on the remaining fold. This process repeats k times, with each fold used exactly once as the test set. Unlike simple train/test splitting, k-fold cross-validation allows all data to be used for out-of-sample testing.

With this approach, we can determine λ *adaptively* as the algorithm iterates, reducing the required number of integrations. Additionally, the approach automatically solves the problem of choosing a suitable error norm when given observations with different units of measure, since cross-validation is performed with the transformed variables X and y. Note that the cross-validation is performed on the $S+P$ rows of X; hence,

unlike typical cross-validation, the variables of the state vector are withheld, rather than the ensemble members. Recall that P is the number of parameters and S is the number of moments.

The corresponding covariance update equation can be derived using standard identities, as follows:

$$\begin{aligned}
\boldsymbol{\Sigma}_A &= \boldsymbol{\Sigma}_B - \boldsymbol{\Sigma}_B \boldsymbol{H}^\top \left(\boldsymbol{H}\boldsymbol{\Sigma}_B\boldsymbol{H}^\top + \lambda\boldsymbol{Q}\right)^{-1} \boldsymbol{H}\boldsymbol{\Sigma}_B && (17)\\
&= \boldsymbol{F}_B\boldsymbol{F}_B^\top - \boldsymbol{F}_B\boldsymbol{F}_B^\top\boldsymbol{H}^\top \left(\boldsymbol{H}\boldsymbol{F}_B\boldsymbol{F}_B^\top\boldsymbol{H}^\top + \lambda\boldsymbol{Q}\right)^{-1} \boldsymbol{H}\boldsymbol{F}_B\boldsymbol{F}_B^\top && (18)\\
&= \boldsymbol{F}_B\left[\boldsymbol{I} - \boldsymbol{F}_B^\top\boldsymbol{H}^\top \left(\boldsymbol{H}\boldsymbol{F}_B\boldsymbol{F}_B^\top\boldsymbol{H}^\top + \lambda\boldsymbol{Q}\right)^{-1} \boldsymbol{H}\boldsymbol{F}_B\right]\boldsymbol{F}_B^\top && (19)\\
&= \boldsymbol{F}_B\left(\boldsymbol{I} + \boldsymbol{F}_B^\top\boldsymbol{H}^\top \left(\lambda\boldsymbol{Q}\right)^{-1} \boldsymbol{H}\boldsymbol{F}_B\right)\boldsymbol{F}_B^\top && (20)\\
&= \boldsymbol{F}_B\boldsymbol{D}\boldsymbol{F}_B^\top && (21)
\end{aligned}$$

where

$$\boldsymbol{D} = \left(\boldsymbol{I} + \lambda^{-1}\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1} = \lambda\left(\lambda\boldsymbol{I} + \boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}. \quad (22)$$

Once we have a value for λ, we may compute $\boldsymbol{D}$ and update the perturbed-parameter ensemble with

$$\boldsymbol{F}_A = \boldsymbol{F}_B\boldsymbol{D}^{1/2}. \quad (23)$$

We used the R package `glmnet` to perform the ridge regression with cross-validation (Friedman et al., 2010). We found that `cv.glmnet` sometimes chooses poor λ values because the default search range is too narrow. To ensure a sufficiently broad range of candidate λ values, the limits of the search range are specified in the following way. Let the singular value decomposition (SVD) of $\boldsymbol{X}$ be

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{C}\boldsymbol{V}^\top, \quad (24)$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices and $\boldsymbol{C}$ is a diagonal matrix whose elements are the singular values $\{c_1, c_2, \cdots, c_S\}$, in descending order. Then, Equation 16 can be expressed as

$$\boldsymbol{\beta} = \boldsymbol{V}\boldsymbol{G}\boldsymbol{U}^\top, \quad (25)$$

where $\boldsymbol{G}$ is a diagonal matrix with elements

$$G_{ii} = \frac{c_i}{c_i^2 + \lambda}. \quad (26)$$

This expression implies that $\boldsymbol{G}$, and therefore $\boldsymbol{\beta}$, is insensitive to λ when $\lambda \ll c_S^2$ or when $\lambda \gg c_1^2$. Hence, the sensitive range is $c_1^2 < \lambda < c_S^2$. However, the column vectors of $\boldsymbol{F}_B$ sum to the zero vector, since the ensemble members are centered, which implies that $\boldsymbol{X}$ is not full column rank. Therefore, $c_S = 0$, so we use the *second* smallest singular value, c_{S-1} to define the lower limit of λ. To be precise, the lower limit is chosen to be a factor of ten less than the second smallest squared singular value, and the upper limit is chosen to be a factor of ten greater than the largest squared singular value. Accounting for the fact that `cv.glmnet` normalizes $\boldsymbol{y}$ by its standard deviation and normalizes the sum-square error in the objective function by the dimension of $\boldsymbol{y}$, the upper and lower limits of λ are chosen to be

$$\lambda_{\text{lower}} = \frac{1}{10}\frac{\sigma_y}{S}c_{S-1}^2 \quad \text{and} \quad \lambda_{\text{upper}} = 10\frac{\sigma_y}{S}c_1^2, \quad (27)$$

where S is the number of elements of $\boldsymbol{y}$ and σ_y is the standard deviation of the elements of $\boldsymbol{y}$.

Compared to the cost of integrating an ESM, computing the SVD is essentially "free", so the cost of this procedure is negligible, unlike offline hyperparameter tuning.

This completes the derivation of the KalmRidge algorithm. In the next section, we will discuss some related algorithms that may serve as a basis for comparison.

4 Alternatives to KalmRidge

We have already discussed the Schneider et al. (2017) approach, but there are other algorithms which address the same problem. In the following subsections, we discuss a few alternatives that will be used as a basis for comparison.

4.1 Unregularized Kalman Filter

In certain situations, one might consider implementing the Kalman filter without regularization. For example, with a sufficiently large ensemble size and extensive observational data, simple sample estimates may provide adequately precise estimates of the prior and observational error covariance matrices, potentially obviating the need for regularization. Here, we explore situations where a large observational dataset is available for estimating $\boldsymbol{R}$. It is important to note, however, that realistic ESM tuning does not typically fit this scenario, as the observational time series is generally too short, resulting in a singular error covariance matrix $\boldsymbol{R}$. It is also important to note that the covariances pertain to statistical moments calculated over a specific time window T_w, rather than instantaneous states commonly dealt with. If T_o denotes the total observational time period, then $\boldsymbol{R}$ can be derived from the moments estimated across T_o/T_w intervals within the observational dataset. The resulting estimate for $\boldsymbol{R}$ will be singular unless the length of the observational period T_o is greater than the product of T_w and the dimension of $\boldsymbol{R}$.

4.2 Consistency Diagnostics

Another approach to estimate the hyperparameter is to choose it to satisfy a consistency diagnostic. One such diagnostic was proposed by Desroziers et al. (2005), who showed that if $\boldsymbol{\Sigma}_B$ and $\boldsymbol{R}$ are correctly specified, then

$$\mathbb{E}[\boldsymbol{d}_{o-a}\boldsymbol{d}_{o-b}^{\top}] = \boldsymbol{R} \tag{28}$$

where

$$\boldsymbol{d}_{o-a} = \boldsymbol{o} - H\boldsymbol{x}_B \tag{29}$$

is the post-fit residual and

$$\boldsymbol{d}_{o-b} = \boldsymbol{o} - H\boldsymbol{x}_B \tag{30}$$

is the pre-fit residual (i.e. the innovation). Li et al. (2009) used these equations to estimate a covariance inflation factor. Here, we use it to estimate the hyperparameter, r. Specifically, if we assume that $\boldsymbol{R} = r^2\boldsymbol{Q}$, as before, then

$$r^2 = \frac{\operatorname{tr}\left(\mathbb{E}[\boldsymbol{d}_{o-a}\boldsymbol{d}_{o-b}^{\top}]\right)}{\operatorname{tr}(\boldsymbol{Q})} \tag{31}$$

gives an estimate of the hyperparameter. Since the right side of Equation 31 implicitly depends on r, the hyperparameter value is determined by fixed-point iteration. With this value of r, one can use ensemble Kalman inversion as previously described. It is necessary to re-evaluate Equation 31 with each iteration.

4.3 Adaptive Inflation

It is well known that the ensemble Kalman filter suffers from ensemble collapse, and strategies like covariance inflation and covariance localization have been employed to prevent this (Anderson & Anderson, 1999). A natural question is whether these strategies could be used in parameter estimation. In fact, the hyperparameter r can be interpreted as a kind of covariance inflation factor. This can be shown by the substitution $\boldsymbol{R} = r^2\boldsymbol{Q}$ into Equations 4 and 5, followed by algebraic rearrangement:

$$\overline{\boldsymbol{x}}_A = \overline{\boldsymbol{x}}_B + (r^{-2}\boldsymbol{\Sigma}_B)\boldsymbol{H}^{\top}\left(\boldsymbol{H}(r^{-2}\boldsymbol{\Sigma}_B)\boldsymbol{H}^{\top} + \boldsymbol{Q}\right)^{-1}(\boldsymbol{o} - \boldsymbol{H}\overline{\boldsymbol{x}}_B) \tag{32}$$

and

$$r^{-2}\mathbf{\Sigma}_A = (r^{-2}\mathbf{\Sigma}_B) - (r^{-2}\mathbf{\Sigma}_B)\boldsymbol{H}^\top \left(\boldsymbol{H}(r^{-2}\mathbf{\Sigma}_B)\boldsymbol{H}^\top + \boldsymbol{Q}\right)^{-1} \boldsymbol{H}(r^{-2}\mathbf{\Sigma}_B). \tag{33}$$

Equations 32 and 33 show that introducing the free hyperparameter r^2 is equivalent to inflating the background covariance matrix $\mathbf{\Sigma}_B$ by r^{-2}, then deflating the analysis covariance matrix $\mathbf{\Sigma}_A$. This suggests that an *adaptive covariance inflation* algorithms can help solve the free hyperparameter problem, as those algorithms provide a way to estimate that hyperparameter.

One of the first adaptive inflation algorithms was Anderson (2007), which was improved by Anderson (2009). Later, El Gharamti (2018) modified Anderson (2009) to use inverse gamma distributions instead of Gaussian distributions. Specifically, we consider the El Gharamti (2018) algorithm. The e subscript indicates the ensemble member and the s subscript indicates the state variable. This algorithm uses a sequential filter and the following inflation rule:

$$x_{s,e}^{\text{inf.}} = \sqrt{\lambda_s}(x_{s,e} - \overline{x}_s) + \overline{x}_s \tag{34}$$

Note that each state variable has its own inflation factor, λ_s. The algorithm uses Bayes' rule with inverse gamma priors to update these inflation factors. The inverse gamma distribution is defined by

$$p(\lambda) = \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{-\alpha-1}\exp(-\beta/\lambda), \tag{35}$$

where $\alpha > 0$ is called the shape parameter and $\beta > 0$ is called the scale parameter. Since these parameters are estimated automatically within the algorithm, they are not free hyperparameters.

To maintain compatibility with code written for the Anderson (2009) algorithm, the El Gharamti (2018) algorithm uses Gaussian parameters and translates them to and from inverse gamma parameters when updating the distribution. The inverse gamma distribution is advantageous because, unlike the Gaussian distribution, it is supported only on the positive real numbers. This is important because inflation factors must be positive.

The user must specify an initial guess (prior) for the inflation factor variances. This implies that the algorithm *does* have a hyperparameter, but the algorithm can update its value adaptively. However, in practice the variance update is often disabled, such that only the mean update is retained. The inflation factor means are initialized at unity (i.e. no inflation) in all applications we have seen, so we do not consider them to constitute free hyperparameters.

4.4 Annealed Regularization

Iglesias and Yang (2021) proposed another algorithm for estimating the hyperparameter. Using our notation, the hyperparameter is estimated at each iteration n by

$$\lambda_n^{-1} = \min\left(\max\left(\frac{E}{2\langle\Phi\rangle_n}, \sqrt{\frac{E}{2\langle\Phi,\Phi\rangle_n}}\right), 1 - t_n\right) \tag{36}$$

where

$$t_n = \begin{cases} \sum_{j=0}^{n-1}\lambda_j^{-1} & n \geq 1, \\ 0 & n = 0 \end{cases} \tag{37}$$

$$\Phi_n = \left\{\frac{1}{2}\|\boldsymbol{Q}^{-1/2}\left(\boldsymbol{o} - \boldsymbol{H}\boldsymbol{x}_{n,e}\right)\|^2\right\}_{e=1}^{E}. \tag{38}$$

The constraint involving t_n causes the value of λ_n^{-1} to tend toward zero.

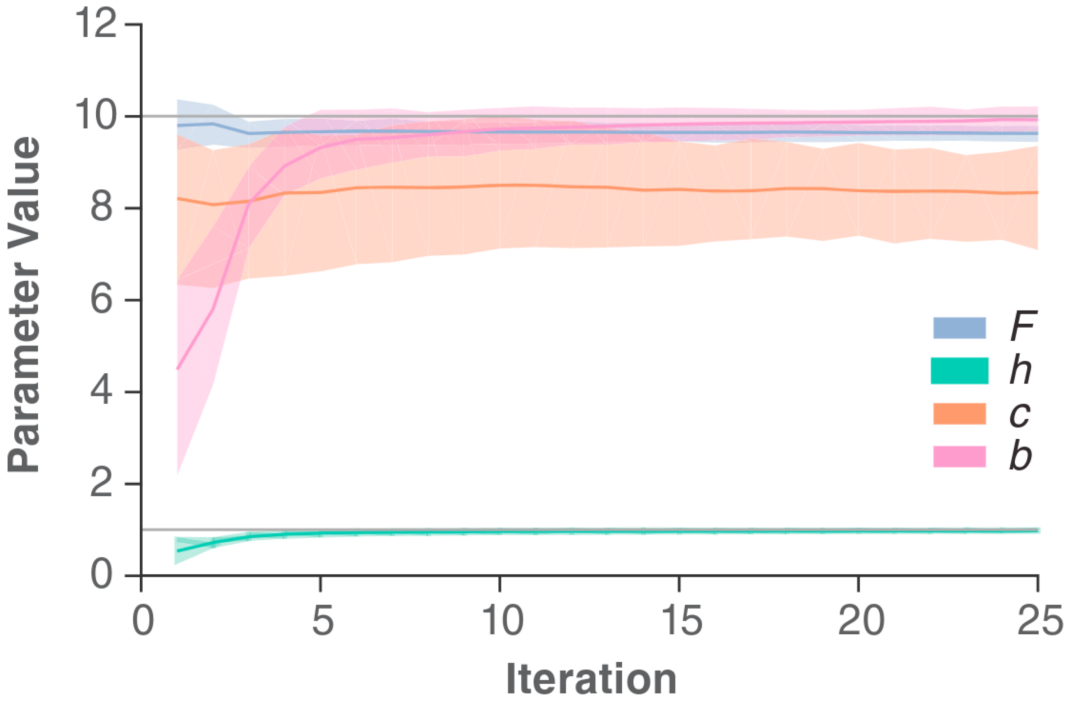

Figure 1. Parameter estimation in Lorenz 96 with EnKI (Schneider et al., 2017).

5 Experiments with Lorenz 96

We now illustrate the above methods using a simple dynamical system. The Lorenz 96 model is an idealized dynamical system with governing equations

$$\frac{dX_k}{dt} = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F - hc\overline{Y}_k, \tag{39}$$

$$\frac{1}{c}\frac{dY_{jk}}{dt} = -bY_{j+1,k}(Y_{j+2,k} - Y_{j-1,k}) - Y_{jk} + \frac{h}{J}X_k, \tag{40}$$

and

$$\overline{Y}_k = \frac{1}{J}\sum_{j=1}^{J} Y_{j,k}. \tag{41}$$

It features $K = 36$ slow variables X_k, and JK fast variables Y_{jk} ($J = 10$ fast variables per slow variable), which form cyclic chains:

$$X_{k+K} = X_k \quad Y_{j,k+K} = Y_{jk} \quad Y_{j+J,k} = Y_{j,k+1}. \tag{42}$$

This dynamical system represents some important aspects of real ESMs, such as advection, external forcing, dissipation, coupling, and chaos (Lorenz, 2006). The fast variables are broadly analogous to higher-resolution atmospheric states and the slow variables are analogous to lower-resolution oceanic states.

The model depends on four parameters: F, an external forcing; h, which controls the coupling strength; c, a relative damping time scale; and b, which modulates advection in the fast dynamics.

Letting $\langle \cdots \rangle$ denote simultaneous averaging over both time (long-term) and j, it follows from Equations 39 and 40 that

$$\left\langle X^2 \right\rangle = F \left\langle X \right\rangle - hc \left\langle X\overline{Y} \right\rangle \tag{43}$$

and

$$\left\langle \overline{Y^2} \right\rangle = \frac{h}{J} \left\langle X\overline{Y} \right\rangle, \tag{44}$$

noting that individual fast (or slow) variables are statistically exchangeable (Lorenz & Emanuel, 1998). These relationships suggest that we can tune Lorenz 96 using the following $5K = 180$ moments as targets, as Schneider et al. (2017) did:

$$\boldsymbol{m} = \left(\langle X \rangle \quad \langle \overline{Y} \rangle \quad \langle X^2 \rangle \quad \langle X\overline{Y} \rangle \quad \left\langle \overline{Y^2} \right\rangle \right)^\top. \tag{45}$$

In the subsections that follow, we compare the performance of different approaches using these statistical moments as targets.

5.1 Ensemble Kalman Inversion

Schneider et al. (2017) applied (non-adaptive) EnKI to the Lorenz 96 model. Specifically, they created a very long control integration ($T = 46416$) with $F = c = b = 10$ and $h = 1$, then used EnKI to estimate the parameters of the control integration using the moments in Equation 45 as (proxy) observations. Their initial ensemble, of size 100, used normal priors for (F, h, b) with $\mu = (10, 0, 5)$, $\sigma^2 = (10, 1, 10)$; and a log-normal prior for c with $\mu = 2$, $\sigma^2 = 0.1$. They found that EnKI produces reasonable parameter estimates (Figure 1).

Figure 2 shows results from experiments with KalmRidge. The mean estimates are similar to those of Schneider et al. (2017), but our ensembles collapsed whereas theirs did not. However, their EnKI implementation added noise to the "observations", whereas our KalmRidge algorithm did not. Despite the similarity in the mean estimates, KalmRidge did not require offline hyperparameter selection, unlike their EnKI.

Both implementations fail to adequately estimate c. To investigate this deficiency, we examined the sensitivity of the moments to parameter perturbations. Figure 3 shows the variations in X^2 and $\overline{Y^2}$ due to changes in one parameter while the other parameters are fixed to their control values. The figure shows that variations in F, h, and b lead to relatively tight (albeit nonlinear) relationships, whereas variations in $\log(c)$ do not. In fact, none of the moments exhibit sensitivity to $\log(c)$ near the control value (not shown), suggesting an identifiability problem. It is likely that the insensitivity to perturbations of $\log(c)$ explains the poor estimates of c.

5.2 Unregularized Kalman Filter

The results from the "basic statistics" approach are shown in Figure 4. Despite a great abundance of proxy observations (the same $T = 46416$ control integration as in the previous section), this method suffers from very rapid ensemble collapse and correspondingly gives poor estimates for some of the parameters.

5.3 Consistency Diagnostics

The results from the consistency diagnostic method are shown in Figure 5. This method is slightly less successful at estimating the F parameter than KalmRidge, but more importantly, ensemble collapse occurs much more quickly in the consistency diagnostic method.

5.4 Adaptive Inflation

El Gharamti (2018) algorithm has previously been implemented in NCAR's Data Assimilation Research Testbed (DART) (Anderson et al., 2004). This is a Matlab implementation, but our work has been in Python and R, so we translated the Matlab code

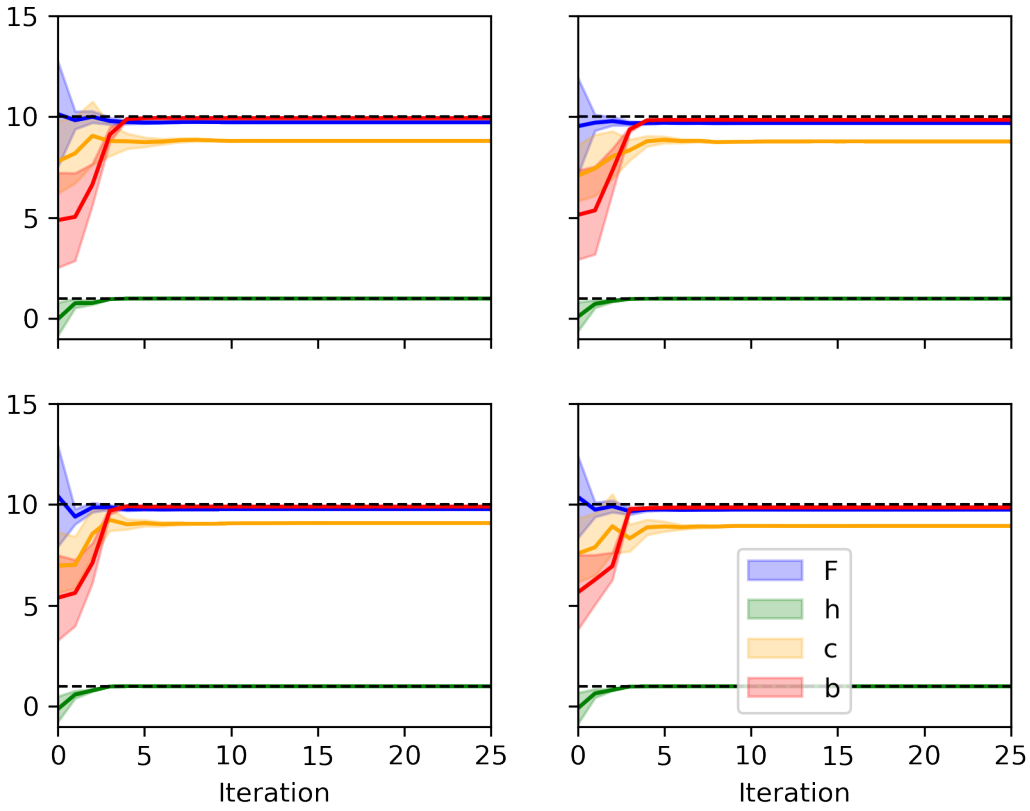

Figure 2. Lorenz 96 parameter estimates (means and interquartile ranges) from KalmRidge. The true parameter values are depicted with dashed lines. Note that some of the dashed lines overlap. Each subfigure is an independent run of the KalmRidge algorithm, each starting with a slightly different initial ensemble.

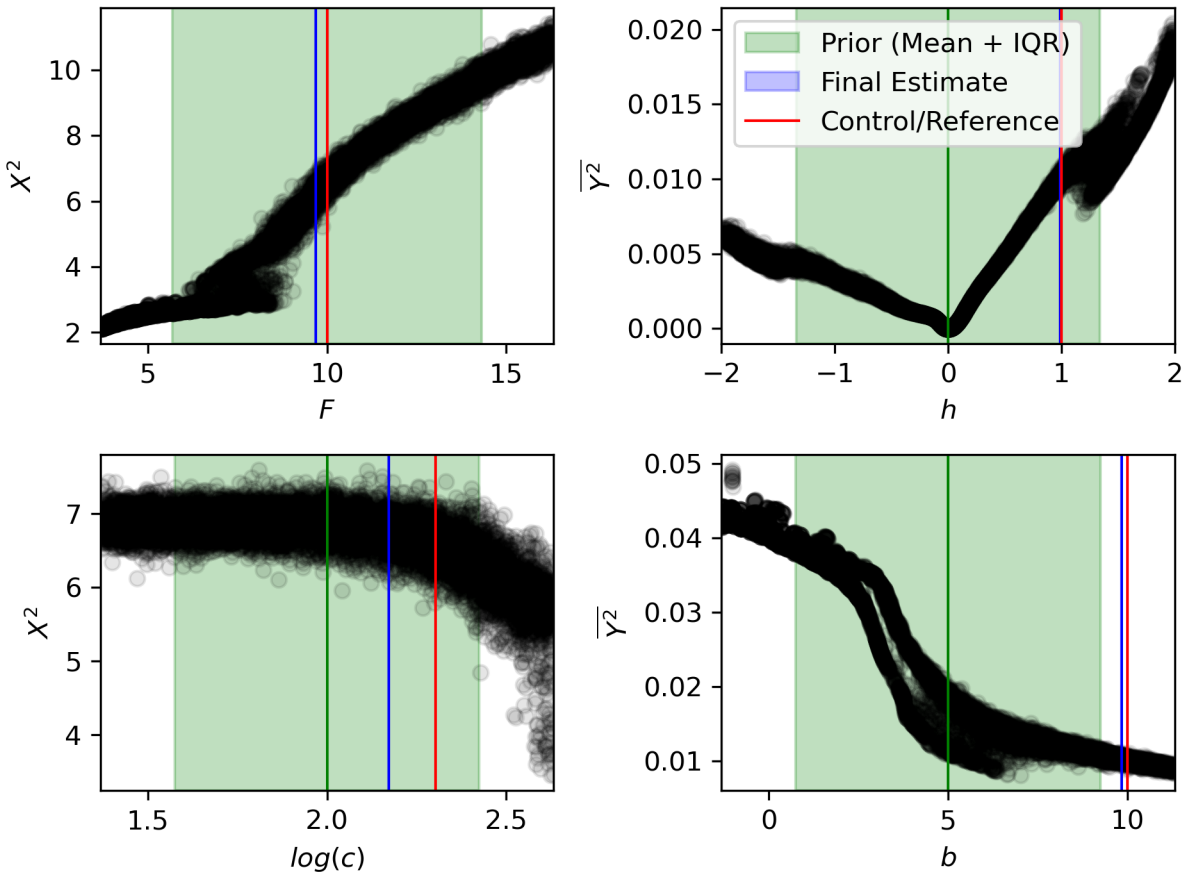

Figure 3. Single-parameter Lorenz 96 sensitivity experiments. In each plot, three parameters were fixed to their control values while the remaining parameter was varied. The vertical axis gives the spatiotemporal mean of a particular moment. Each black dot corresponds to a separate integration of Lorenz 96. The total number of Lorenz 96 integrations was 10,000.

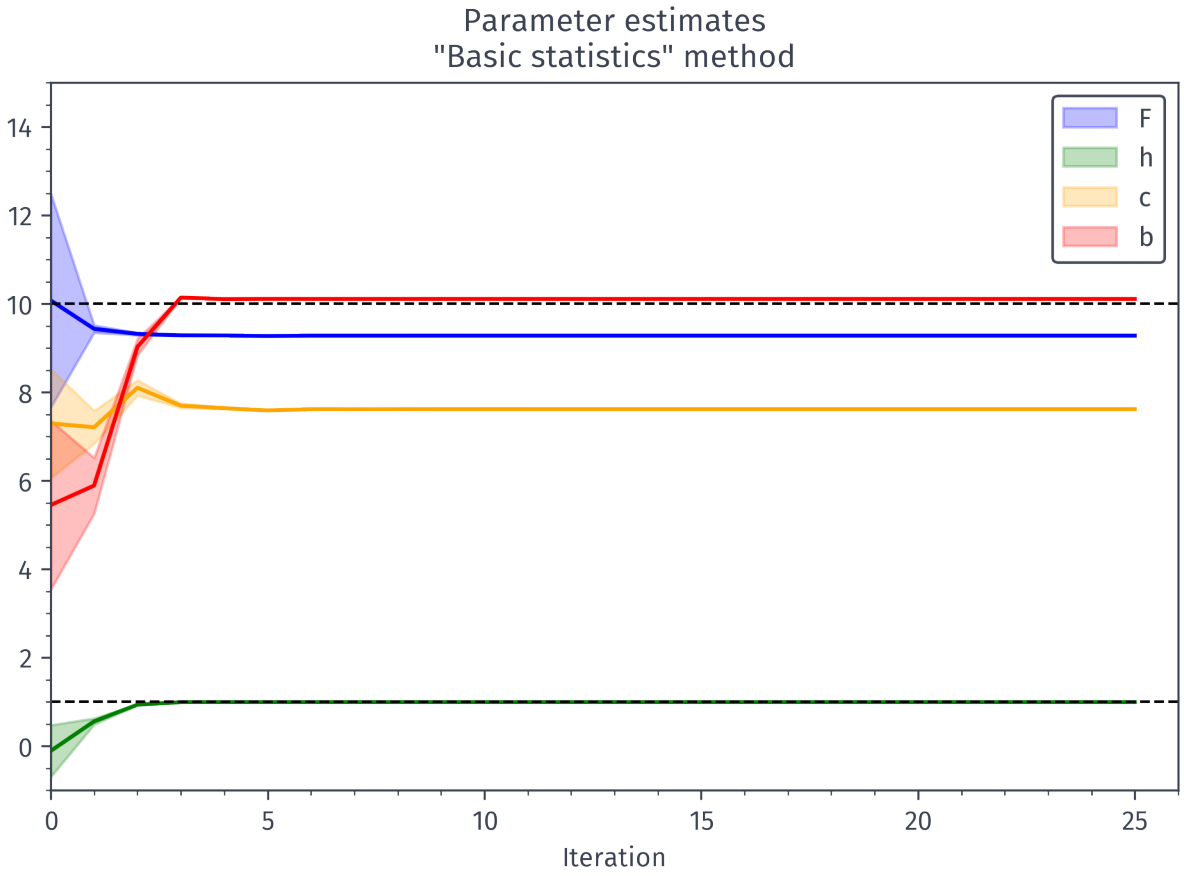

Figure 4. Lorenz 96 parameter estimates (means and interquartile ranges) from the "basic statistics" approach.

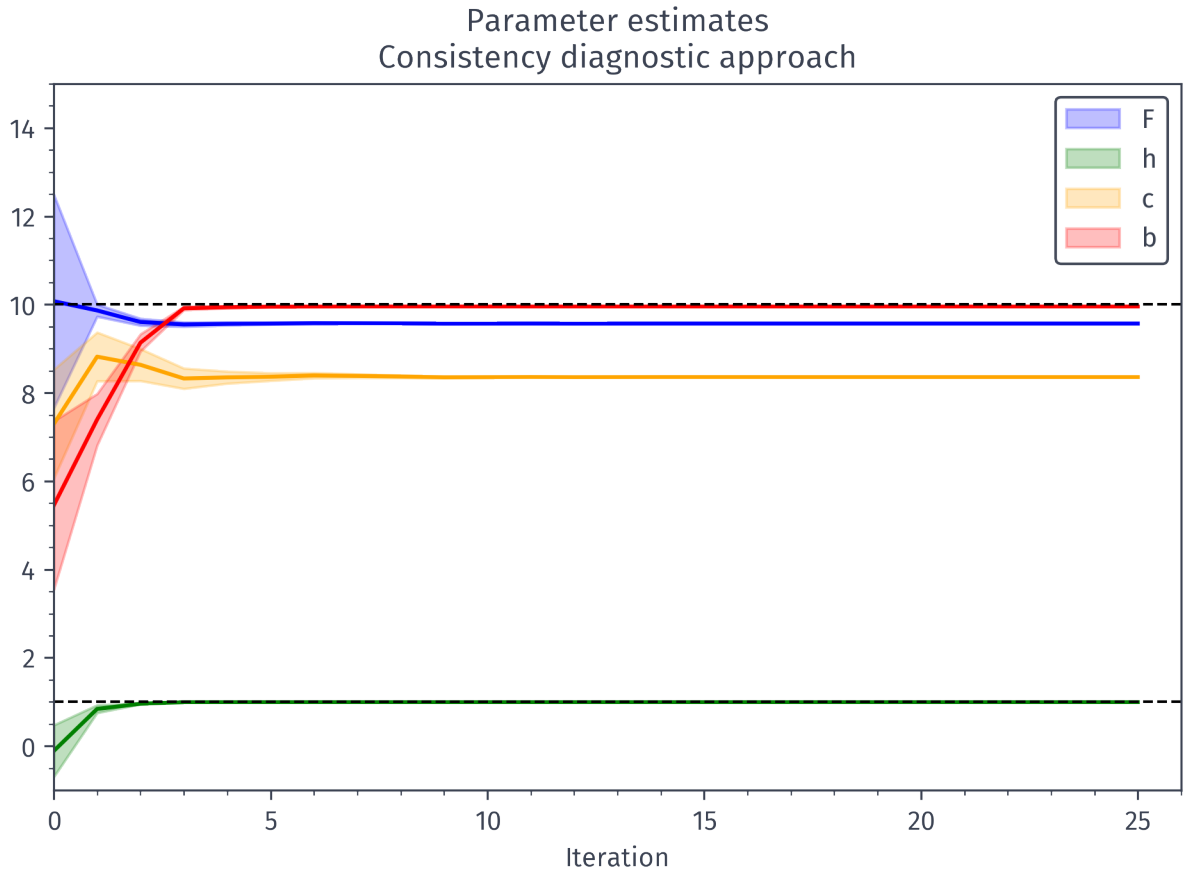

Figure 5. Lorenz 96 parameter estimates (means and interquartile ranges) from the consistency diagnostic approach.

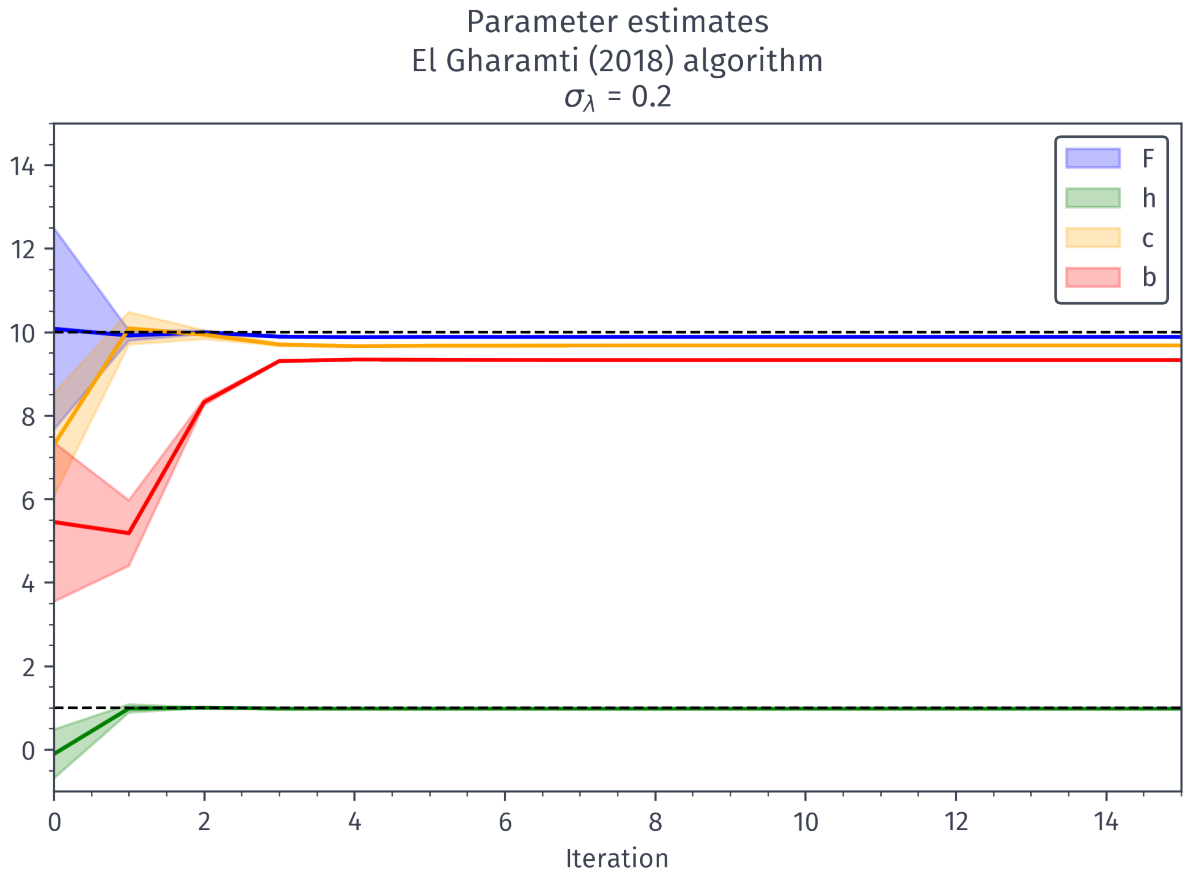

Figure 6. Successful parameter estimation with the El Gharamti (2018) algorithm.

to Python. Consistent with the Matlab implementation, the Python implementation disables the variance update, retaining only the mean update.

The results of applying our implementation of the El Gharamti (2018) algorithm to the Lorenz 96 tuning problem are shown in Figure 6. The figure shows that the algorithm works reasonably well for $\sigma_{\lambda_b} = 0.2$, which is much smaller than the values considered by the author and used in DART (typically around 0.6). However, increasing it much beyond $\sigma_{\lambda_b} = 0.2$ causes immediate failure – the parameter variance becomes very large, rapidly. The estimation problem is seen in the c parameter consistently, and this parameter is difficult to estimate, as was shown previously in this section. Since this algorithm offers no method for choosing σ_{λ_b}, and the estimates from the algorithm are sensitive to the value of σ_{λ_b}, we consider σ_{λ_b} to be a free hyperparameter.

5.5 Annealed Regularization

Like our algorithm, the Iglesias and Yang (2021) algorithm does not have a free hyperparameter. However, as seen in Figure 7, their algorithm terminates before the parameter estimates converge. One might suggest continuing the iteration, but this is not possible since the algorithm must stop when λ^{-1} reaches zero. The variations in the hyperparameter estimates with iteration for KalmRidge and adaptive regularization are shown in Figure 8 and Figure 9, respectively. Note that the hyperparameter values chosen by the adaptive regularization algorithm trend downward, whereas those chosen by

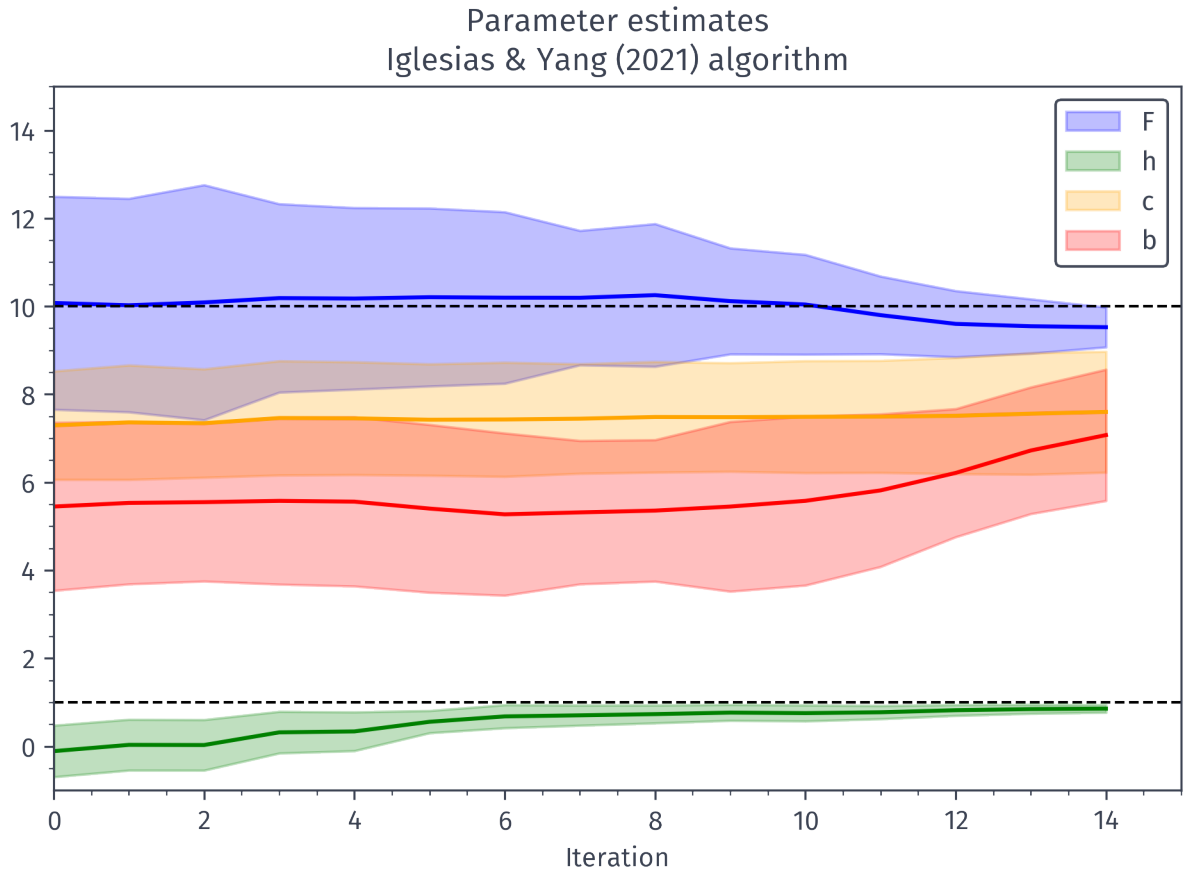

Figure 7. Lorenz 96 parameter estimates (means and interquartile ranges) using the Iglesias and Yang (2021) algorithm. The true parameter values are depicted with dashed lines.

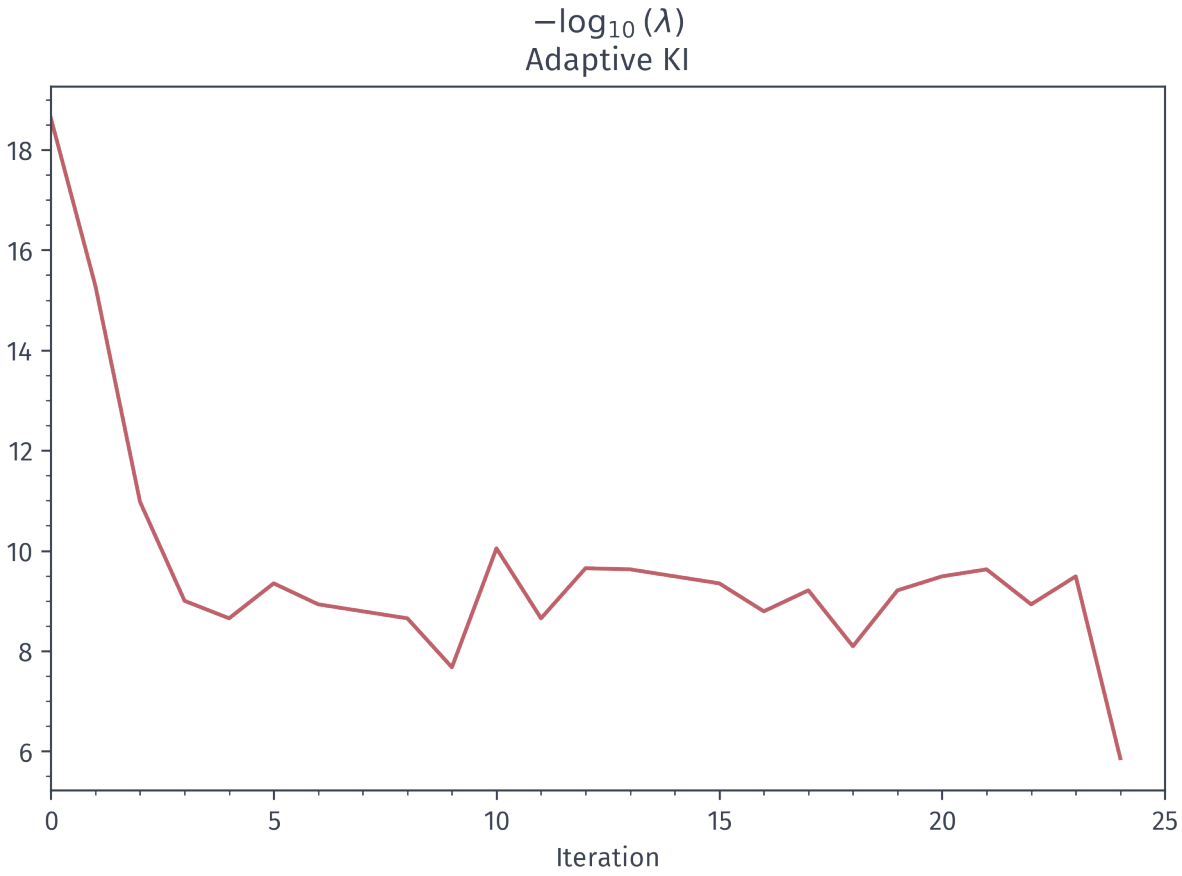

Figure 8. Evolution of $-\log_{10}(\lambda)$ in KalmRidge when applied to Lorenz 96.

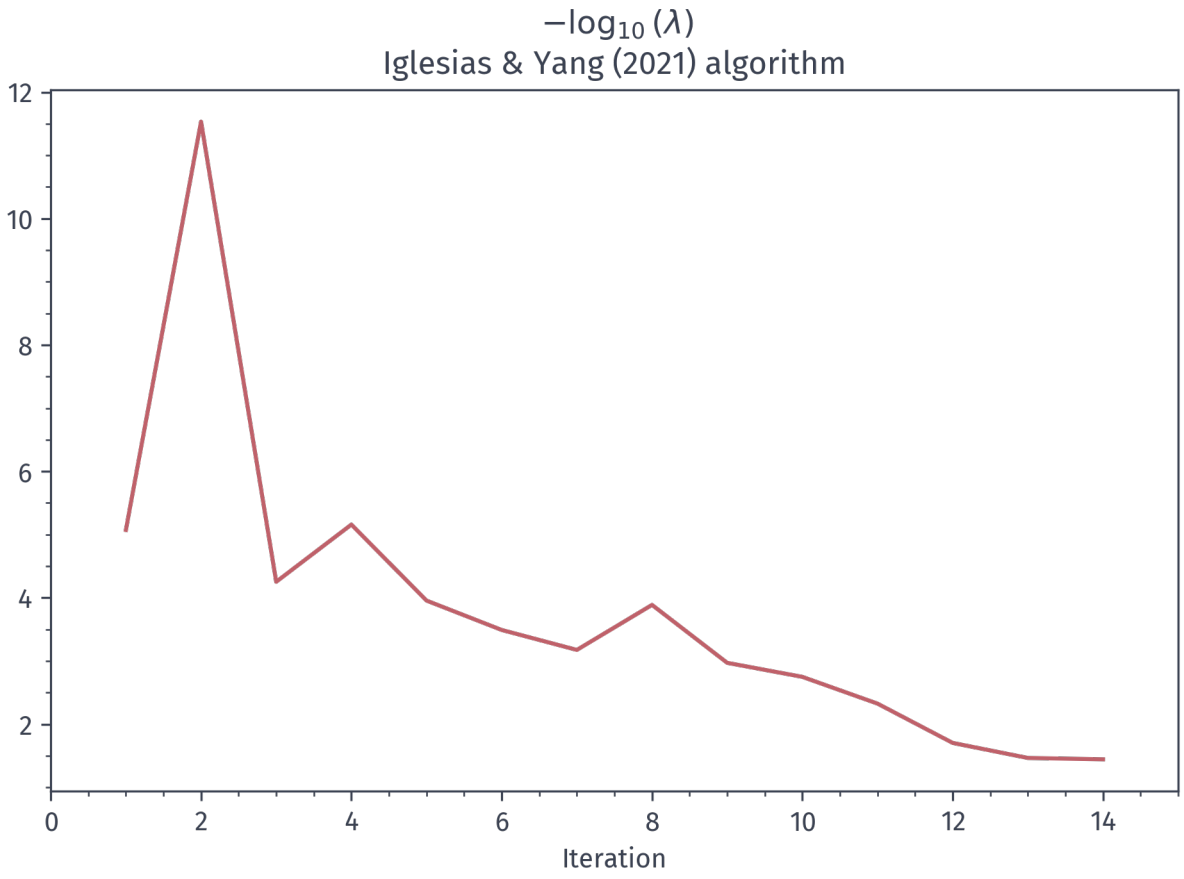

Figure 9. Evolution of $-\log_{10}(\lambda)$ in the Iglesias and Yang (2021) algorithm when applied to Lorenz 96.

KalmRidge do not. Unlike adaptive regularization, KalmRidge can be iterated until convergence, since λ is not forced to evolve in any particular way across iterations.

6 Demonstrations With CESM2

Unlike the Lorenz 96 model, which is simple "toy model", the Community Earth System Model version 2 (CESM2) represents all major Earth system components and features hundreds of tunable parameters (Danabasoglu et al., 2020). In this section, we demonstrate applications of KalmRidge to CESM2.

Parameter selection is a major unsolved problem in ESM tuning. Here, our goal is to demonstrate the usefulness of a tuning algorithm for a small number of selected parameters. Many ESM tuning studies focus on cloud parameters due to their important role in quantifying uncertainty in climate projections. For the purpose of illustration, we selected two parameters that appear frequently in previous studies on tuning and sensitivity of CESM, following recommendations by NCAR scientists (see Acknowledgements).

The first selected parameter is called `clubb_gamma_coef` (henceforth γ). This parameter is part of the Cloud Layers Unified By Binormals (CLUBB) parameterization and controls the skewness of vertical velocity (Golaz et al., 2002). CESM2 is highly sensitive to this parameter and it is commonly tuned to achieve a TOA energy balance target (Danabasoglu et al., 2020; Guo et al., 2014; Woelfle et al., 2019). The second parameter is called `micro_mg_dcs` (henceforth D_{cs}). This parameter is part of the Morrison-Gettelman cloud microphysics parameterization and determines the autoconversion size from ice to snow (Gettelman & Morrison, 2015). Like γ, D_{cs} is known to be highly useful for tuning radiation budgets (Zhao et al., 2013). Both γ and D_{cs} were investigated in connection with the double-ITCZ bias (Woelfle et al., 2019). We performed sensitivity experiments and found that γ and D_{cs} are highly correlated with the spatiotemporal means of sea-surface temperature (SST) and longwave cloud forcing, respectively.

As in the preceding section, we used KalmRidge to estimate the parameter values of a given control integration. In particular, we used a 500-year two-degree fully coupled preindustrial control, which is part of the Coupled Model Intercomparison Project version 6 (CMIP6). This integration used the parameter values $\gamma = 0.28$ and $D_{cs} = 200\times 10^{-6}$.

Our goal is to tune parameters with reasonable computational expense. For concreteness, assume that, for each iteration of KalmRidge, we can afford to integrate CESM2 for up to 40 years. Each simulated year costs approximately 1500 core-hours on the Cheyenne supercomputer. If each ensemble member is run for Y years, then the ensemble size E is chosen to satisfy $EY = 40$. We chose $Y = 2$, which implies $E = 20$. Hence, we used 20-member two-year ensembles.

Our initial perturbed-parameter ensemble, given in Table 1, was a uniform random sample with $\gamma \in [0.25, 0.35]$ and $D_{cs} \in [200 \times 10^{-6}, 800 \times 10^{-6}]$. The literature does not clearly define suitable parameter ranges, so we used these broad ranges following recommendations by NCAR scientists. Each ensemble member was initialized at 0031-01-01 using a restart from the control (for more information, see the "Open Research" section).

Before we can apply KalmRidge, we must choose which statistical moments to target. Our choice was guided by a list of properties which model developers consider "decisive" for tuning, as indicated by a survey of modeling centers (Hourdin et al., 2017). We selected a subset of model output fields for consideration, as listed in Table 2. We selected these particular fields because they are simple and rather uncontroversial – the SST, for example, is an common "first choice".

clubb_gamma_coef	micro_mg_dcs ($\times 10^6$)
0.32917	517
0.27894	310
0.29663	347
0.27104	277
0.29237	588
0.2818	449
0.30702	463
0.34786	679
0.28595	462
0.30232	256
0.28154	418
0.27828	272
0.26183	584
0.25641	615
0.31531	352
0.30680	755
0.29615	668
0.32782	722
0.27089	297
0.26318	630

Table 1. Initial perturbed-parameter ensemble used for tuning experiments.

For each field chosen, we computed anomalies by subtracting the annual cycle estimated from the control integration. Then, we computed moments by taking the time means of the leading 100 spherical harmonic projections. Unlike individual grid-cell values, spherical harmonics capture large-scale orthogonal patterns: the first is the global mean, the next few correspond to dipoles, and the patterns decrease in spatial scale thereafter. Figure 10 depicts some of the leading spherical harmonics.

We conducted three experiments. In the "kitchen sink" experiment, we considered all fields listed in Table 2. Then, to investigate the necessity of these fields, we performed two further experiments: one using only SST, and one using both SST and longwave cloud forcing. Figure 11 depicts the results of these experiments. We see that the parameters are well-estimated in all three experiments, even after a single iteration. The estimates using SST only are slightly biased after two iterations. Introducing the longwave cloud forcing reduces this bias, whereas the other fields seem to be uninformative as their inclusion produces only minor changes.

In the above experiments, all ensemble members were initialized from the same state. We also conducted preliminary experiments in which each ensemble member was initialized from states drawn independently from the control integration using a Halton sequence (Halton, 1960). The initial ensemble for these experiments is given in Table 3 and comparisons of the results with the preceding experiments are shown in Figure 12. As seen in Figure 12, using a common initialization usually gave better estimates.

7 Discussion

In this article, we introduced a new algorithm, *KalmRidge*, and demonstrated applications to Lorenz 96 and CESM2. We found that KalmRidge successfully reproduces results from Schneider et al. (2017) when applied to Lorenz 96.

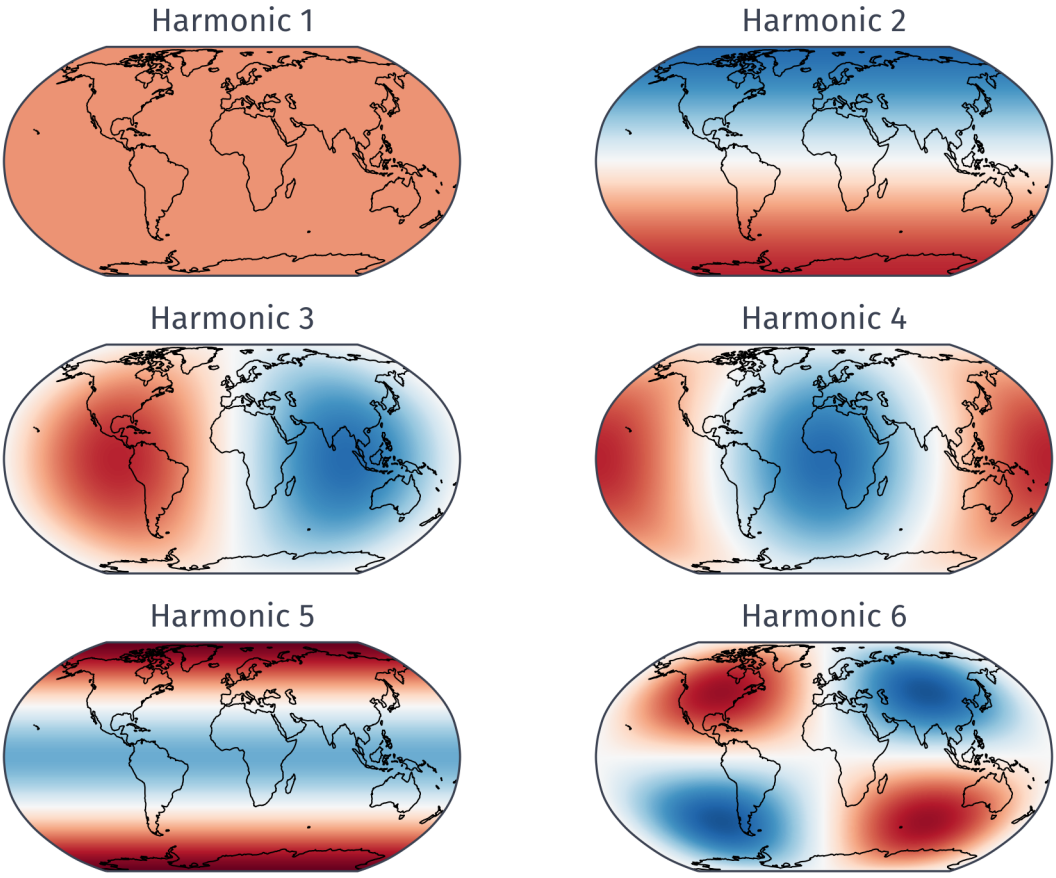

Figure 10. Leading spherical harmonic patterns.

Short name	Long name
FLNS	Net longwave flux at the surface
FLNT	Net longwave flux at the top of model
FSNS	Net solar flux at the surface
FSNT	Net solar flux at the top of model
ICEFRAC	Fraction of surface area covered by sea ice
LWCF	Longwave cloud forcing
PRECT	Total precipitation rate
SNOWHICE	Water equivalent snow depth (ice)
SNOWHLND	Water equivalent snow depth (land)
SST	Sea surface temperature
SWCF	Shortwave cloud forcing
TS	Surface temperature (radiative)

Table 2. CESM2 fields used for tuning experiments.

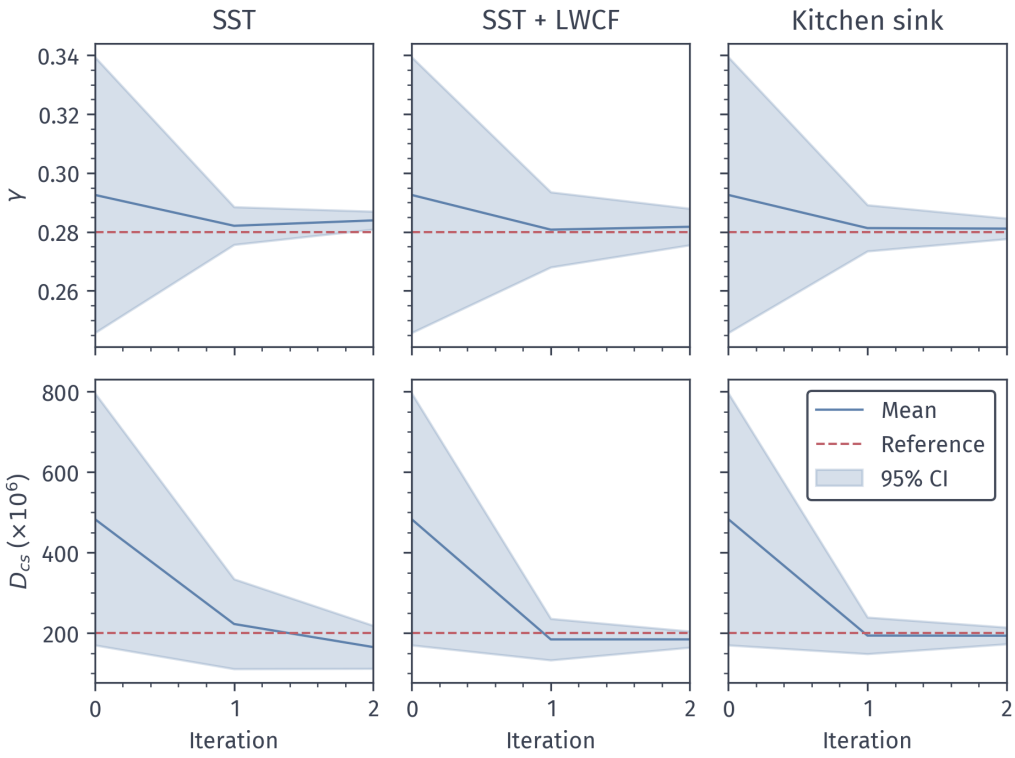

Figure 11. Results of CESM experiments, depicting parameter estimates with 95% confidence intervals before iterating (iteration zero) and in subsequent iterations.

`clubb_gamma_coef`	`micro_mg_dcs` ($\times 10^6$)	Initialization time
0.258838	626	0281-01-01
0.308838	359	0471-01-01
0.283838	559	0371-01-01
0.333838	759	0101-01-01
0.271338	315	0201-01-01
0.321338	515	0301-01-01
0.296338	715	0491-01-01
0.346338	248	0391-01-01
0.255713	448	0041-01-01
0.305713	648	0131-01-01
0.280713	381	0231-01-01
0.330713	581	0431-01-01
0.268213	781	0331-01-01
0.318213	278	0061-01-01
0.293213	478	0151-01-01
0.343213	678	0251-01-01
0.261963	211	0451-01-01
0.311963	411	0351-01-01
0.286963	611	0016-01-01
0.336963	344	0121-01-01

Table 3. Initial perturbed-parameter ensemble used for mixed-initialization experiments.

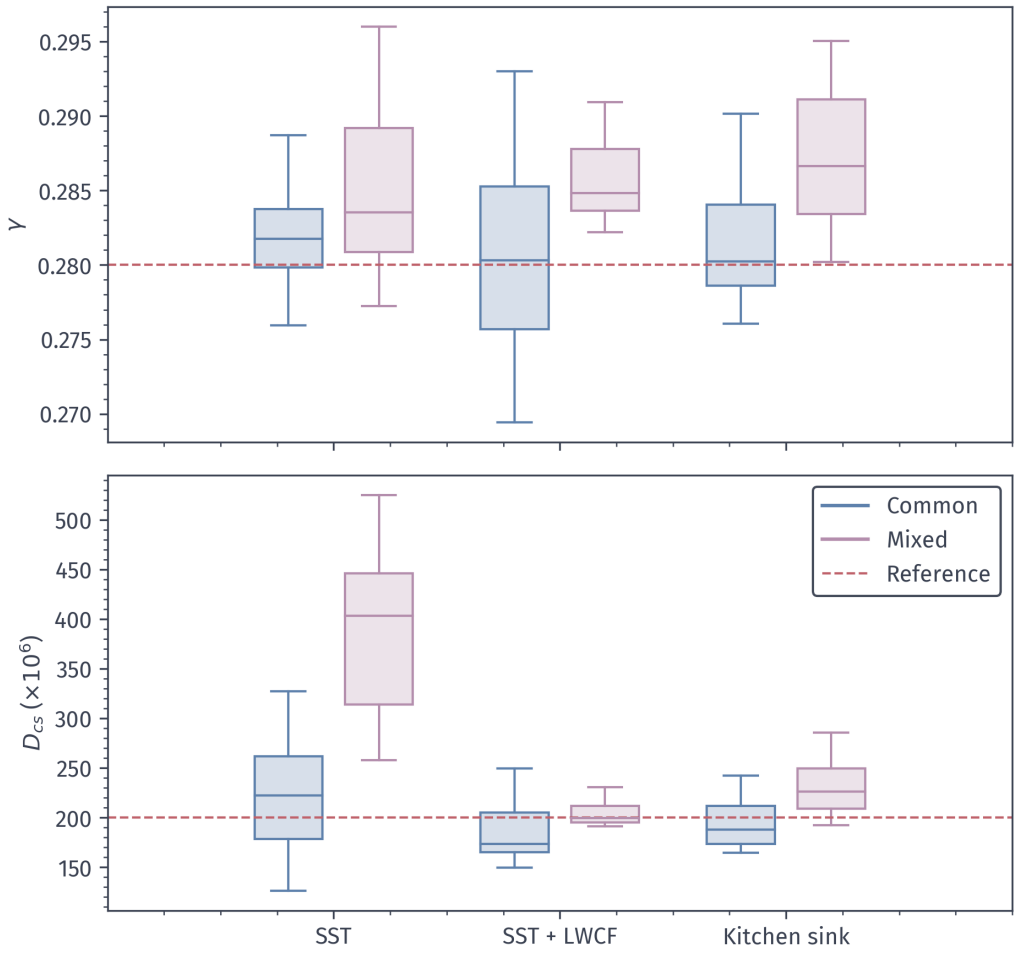

Figure 12. Comparison between common- and mixed-initialization experiments after one iteration.

However, unlike the form of EnKI presented in that article, KalmRidge does not have a free hyperparameter. This eliminates the need for computationally expensive offline hyperparameter selection. We also proposed using spherical harmonic projections, which systematically capture large-scale spatial features, as tuning targets. We have not explored the sensitivity of the algorithm to different basis vectors, but it is plausible that Laplacian eigenvectors are particularly advantageous to tuning ESMs, given that they systematically decompose variability based on spatial length scale. We note that the Laplacian eigenvectors used here may be computed for arbitrary domains, including disconnected land masses or ocean area(DelSole & Tippett, 2015). Laplacian eigenvectors computed for regional domains could facilitate more localized tuning efforts.

KalmRidge, using spherical harmonic projections, demonstrated impressive performance in estimating the parameters of a CESM2 preindustrial control integration. Despite its simplicity, this algorithm also demonstrated its robustness in the presence of irrelevant information – many of the moments, particularly in the "kitchen sink" experiment (e.g. the 57th spherical harmonic of the sea ice fraction), are irrelevant but do not detract from the performance of KalmRidge.

Our results (Figure 12) suggest that using a common initial condition for all ensemble members is better than using mixed initial conditions, but more experiments would be needed to assess the generality of this result.

One caveat of KalmRidge is that its uncertainty estimates are too small. This was also observed, to a greater extent, in the other tuning methods that we evaluated, except for those by Schneider et al. (2017) and Iglesias and Yang (2021). However, the Schneider et al. (2017) algorithm requires the user to specify the value of a free hyperparameter, but provides no procedure to estimate it. Without such a procedure, it is unclear how one could determine the hyperparameter value without already knowing the optimal parameter values. The Iglesias and Yang (2021) algorithm does not have a free hyperparameter, but gives poor point estimates for the parameters. Hence, our algorithm is unique in that it gives good point estimates for the parameters without the requisite tuning of a free hyperparameter.

The model experiments presented in this article are admittedly idealized, particularly since we assume that all model error is due to misspecification of two parameters. For realistic applications, the algorithm must be generalized to account for the annual cycle and climate change. Nevertheless, KalmRidge performs well in this idealized setting, and any algorithm which cannot is unlikely to be useful when applied to more realistic problems.

8 Open Research

The CMIP6 preindustrial control integration used as the tuning control was obtained from NCAR Campaign Storage at `/glade/campaign/collections/cmip/CMIP6/timeseries-cmip6/` (Eyring et al., 2016). These data are also available at `https://www.earthsystemgrid.org/dataset/ucar.cgd.cesm2.b.e21.B1850.f09_g17.CMIP6-piControl.001.atm.proc.monthly_ave.html` and are licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.

The latest version of the KalmRidge software is available at `https://github.com/nlydeen/adaptive-ki/` (Lydeen, 2022). The Community Earth System Model (CESM) version 2.1.1 is available at (Danabasoglu et al., 2019) and is developed at `https://github.com/ESCOMP/cesm/`. We used Conda version 4.14.0 for package management, available under a BSD 3-Clause License at `https://docs.conda.io/en/latest/miniconda.html` and developed at `https://github.com/conda/conda/`. Python and R codes are interfaced with RPy2. Data analysis was conducted with Numpy 1.7.1 (Harris et al., 2020), available under a BSD 3-Clause License at `https://numpy.org/install/` and devel-

oped at `https://github.com/numpy/numpy/`; XArray 0.20.1 (Hoyer & Hamman, 2017), available under an Apache 2.0 License at (Hoyer et al., 2021) and developed at `https://github.com/pydata/xarray`; and Pandas 1.3.5, availabe under a BSD 3-Clause License at (Reback et al., 2021) and developed at `https://github.com/pandas-dev/pandas/`. Plots were created with Matplotlib 3.5.2 (Hunter, 2007), available under the Matplotlib License at (Caswell et al., 2022) and developed at `https://github.com/matplotlib/matplotlib/`.

Acknowledgements

This research was primarily supported by the National Science Foundation (OAC-1934529). We also acknowledge high-performance computing support from Cheyenne (Computational and Information Systems Laboratory (CISL), 2017). Finally, we acknowledge Andrew Gettelman and Cécile Hannay for their guidance in selecting CESM2 parameters and plausible ranges.

References

Anderson, J. L. (2007, January). An adaptive covariance inflation error correction algorithm for ensemble filters. *Tellus A: Dynamic Meteorology and Oceanography*, *59*(2), 210. doi: 10.1111/j.1600-0870.2006.00216.x

Anderson, J. L. (2009, January). Spatially and temporally varying adaptive covariance inflation for ensemble filters. *Tellus A*, *61*(1), 72–83. doi: 10.1111/j.1600-0870.2008.00361.x

Anderson, J. L., & Anderson, S. L. (1999, December). A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, *127*(12), 2741–2758. doi: 10.1175/1520-0493(1999)127⟨2741:amciot⟩2.0.co;2

Anderson, J. L., Hoar, T., Raeder, K., & Collins, N. (2004). *Data assimilation research testbed (DART).* UCAR/NCAR - Computational and Information Systems Laboratory (CISL). Retrieved from `https://dart.ucar.edu/` doi: 10.5065/D6WQ0202

Annan, J. D., Lunt, D. J., Hargreaves, J. C., & Valdes, P. J. (2005, February). Parameter estimation in an atmospheric GCM using the ensemble Kalman filter. *Nonlinear Processes in Geophysics*, *12*(3), 363–371. doi: 10.5194/npg-12-363-2005

Bishop, C. H., & Hodyss, D. (2009, January). Ensemble covariances adaptively localized with ECO-RAP. part 1: tests on simple error models. *Tellus A*, *61*(1), 84–96. doi: 10.1111/j.1600-0870.2008.00371.x

Carrera, J., & Neuman, S. P. (1986, February). Estimation of aquifer parameters under transient and steady state conditions: 1. maximum likelihood method incorporating prior information. *Water Resources Research*, *22*(2), 199–210. doi: 10.1029/wr022i002p00199

Caswell, T. A., Lee, A., Droettboom, M., De Andrade, E. S., Hoffmann, T., Klymak, J., ... Kniazev, N. (2022). *matplotlib/matplotlib: Rel: v3.5.3* [software]. Zenodo. Retrieved from `https://zenodo.org/record/6982547` doi: 10.5281/ZENODO.6982547

Computational and Information Systems Laboratory (CISL). (2017). *Cheyenne: SGI ICE XA cluster.* UCAR/NCAR. Retrieved from `https://www2.cisl.ucar.edu/resources/computational-systems/cheyenne` doi: 10.5065/d6rx99hx

Danabasoglu, G., Lamarque, J.-F., Bacmeister, J., Bailey, D. A., DuVivier, A. K., Edwards, J., ... Strand, W. G. (2019, July 8). *CESM-release-cesm2.1.1* [software]. Zenodo. Retrieved from `https://zenodo.org/record/3895315` doi: https://doi.org/10.5281/zenodo.3895315

Danabasoglu, G., Lamarque, J.-F., Bacmeister, J., Bailey, D. A., DuVivier, A. K., Edwards, J., ... Strand, W. G. (2020, February). The Community Earth System Model version 2 (CESM2). *Journal of Advances in Modeling Earth Systems*, *12*(2). doi: 10.1029/2019ms001916

DelSole, T., & Tippett, M. K. (2015, September). Laplacian eigenfunctions for climate analysis. *Journal of Climate*, *28*(18), 7420–7436. doi: 10.1175/jcli-d-15-0049.1

Desroziers, G., Berre, L., Chapnik, B., & Poli, P. (2005, October). Diagnosis of observation, background and analysis-error statistics in observation space. *Quarterly Journal of the Royal Meteorological Society*, *131*(613), 3385–3396. doi: 10.1256/qj.05.108

El Gharamti, M. (2018, February). Enhanced adaptive inflation algorithm for ensemble filters. *Monthly Weather Review*, *146*(2), 623–640. doi: 10.1175/mwr-d-17-0187.1

Eyring, V., Bony, S., Meehl, G. A., Senior, C. A., Stevens, B., Stouffer, R. J., & Taylor, K. E. (2016, May). Overview of the Coupled Model Intercomparison Project phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development*, *9*(5), 1937–1958. doi: 10.5194/gmd-9-1937-2016

Friedman, J., Hastie, T., & Tibshirani, R. (2010, February). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*(1). doi: 10.18637/jss.v033.i01

Gettelman, A., & Morrison, H. (2015, February). Advanced two-moment bulk microphysics for global models. part I: Off-line tests and comparison with other schemes. *Journal of Climate*, *28*(3), 1268–1287. doi: 10.1175/jcli-d-14-00102.1

Golaz, J.-C., Larson, V. E., & Cotton, W. R. (2002, December). A PDF-based model for boundary layer clouds. part I: Method and model description. *Journal of the Atmospheric Sciences*, *59*(24), 3540–3551. doi: 10.1175/1520-0469(2002)059⟨3540:apbmfb⟩2.0.co;2

Guo, Z., Wang, M., Qian, Y., Larson, V. E., Ghan, S., Ovchinnikov, M., ... Zhou, T. (2014, August). A sensitivity analysis of cloud properties to CLUBB parameters in the single-column Community Atmosphere Model (SCAM5). *Journal of Advances in Modeling Earth Systems*, *6*(3), 829–858. doi: 10.1002/2014ms000315

Halton, J. H. (1960, December). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, *2*(1), 84–90. doi: 10.1007/bf01386213

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020, Sep 01). Array programming with NumPy. *Nature*, *585*(7825), 357-362. Retrieved from `https://doi.org/10.1038/s41586-020-2649-2` doi: 10.1038/s41586-020-2649-2

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. Springer New York.

Hourdin, F., Mauritsen, T., Gettelman, A., Golaz, J.-C., Balaji, V., Duan, Q., ... Williamson, D. (2017, March). The art and science of climate model tuning. *Bulletin of the American Meteorological Society*, *98*(3), 589–602. doi: 10.1175/bams-d-15-00135.1

Hoyer, S., & Hamman, J. (2017, April). xarray: N-d labeled arrays and datasets in python. *Journal of Open Research Software*, *5*(1), 10. Retrieved from `https://doi.org/10.5334/jors.148` doi: 10.5334/jors.148

Hoyer, S., Roos, M., Hamman, J., Keewis, Cherian, D., Fitzgerald, C., ... Wolfram, P. J. (2021). *pydata/xarray: v0.20.1* [software]. Zenodo. Retrieved from `https://zenodo.org/record/5648431` doi: 10.5281/ZENODO.5648431

Hu, X., Zhang, F., & Nielsen-Gammon, J. W. (2010, April). Ensemble-based simultaneous state and parameter estimation for treatment of mesoscale

model error: A real-data study. *Geophysical Research Letters*, *37*(8). doi: 10.1029/2010gl043017

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. Retrieved from `https://doi.org/10.1109/mcse.2007.55` doi: 10.1109/mcse.2007.55

Iglesias, M. A., Law, K. J. H., & Stuart, A. M. (2013, March). Ensemble Kalman methods for inverse problems. *Inverse Problems*, *29*(4), 045001. doi: 10.1088/0266-5611/29/4/045001

Iglesias, M. A., & Yang, Y. (2021, January). Adaptive regularisation for ensemble Kalman inversion. *Inverse Problems*, *37*(2), 025008. doi: 10.1088/1361-6420/abd29b

Jackson, C., Sen, M. K., & Stoffa, P. L. (2004, July). An efficient stochastic Bayesian approach to optimal parameter and uncertainty estimation for climate model predictions. *Journal of Climate*, *17*(14), 2828–2841. doi: 10.1175/1520-0442(2004)017⟨2828:aesbat⟩2.0.co;2

Kalman, R. E. (1960, March). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, *82*(1), 35–45. doi: 10.1115/1.3662552

Koyama, H., & Watanabe, M. (2010, August). Reducing forecast errors due to model imperfections using ensemble Kalman filtering. *Monthly Weather Review*, *138*(8), 3316–3332. doi: 10.1175/2010mwr3067.1

Li, H., Kalnay, E., & Miyoshi, T. (2009, January). Simultaneous estimation of covariance inflation and observation errors within an ensemble Kalman filter. *Quarterly Journal of the Royal Meteorological Society*, *135*(639), 523–533. doi: 10.1002/qj.371

Lorenz, E. N. (2006, July). Predictability – a problem partly solved. In *Predictability of weather and climate* (pp. 40–58). Cambridge University Press. doi: 10.1017/cbo9780511617652.004

Lorenz, E. N., & Emanuel, K. A. (1998, February). Optimal sites for supplementary weather observations: Simulation with a small model. *Journal of the Atmospheric Sciences*, *55*(3), 399–414. doi: 10.1175/1520-0469(1998)055⟨0399:osfswo⟩2.0.co;2

Lydeen, N. (2022). *nlydeen/adaptive-ki: Initial public release* [software]. Zenodo. Retrieved from `https://zenodo.org/record/7337913` doi: 10.5281/ZENODO.7337913

Reback, J., Jbrockmendel, McKinney, W., Van Den Bossche, J., Augspurger, T., Cloud, P., ... Skipper Seabold (2021). *pandas-dev/pandas: Pandas 1.3.5* [software]. Zenodo. Retrieved from `https://zenodo.org/record/5774815` doi: 10.5281/ZENODO.5774815

Schmidt, G. A., Bader, D., Donner, L. J., Elsaesser, G. S., Golaz, J.-C., Hannay, C., ... Saha, S. (2017, September). Practice and philosophy of climate model tuning across six US modeling centers. *Geoscientific Model Development*, *10*(9), 3207–3223. doi: 10.5194/gmd-10-3207-2017

Schneider, T., Lan, S., Stuart, A., & Teixeira, J. (2017, December). Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, *44*(24). doi: 10.1002/2017gl076101

Severijns, C. A., & Hazeleger, W. (2005, September). Optimizing parameters in an atmospheric general circulation model. *Journal of Climate*, *18*(17), 3527–3535. doi: 10.1175/jcli3430.1

Woelfle, M. D., Bretherton, C. S., Hannay, C., & Neale, R. (2019, July). Evolution of the double-ITCZ bias through CESM2 development. *Journal of Advances in Modeling Earth Systems*, *11*(7), 1873–1893. doi: 10.1029/2019ms001647

Yang, X., & DelSole, T. (2009, January). Using the ensemble Kalman filter to estimate multiplicative model parameters. *Tellus A: Dynamic Meteorology and*

Oceanography, *61*(5), 601. doi: 10.1111/j.1600-0870.2009.00407.x

Zhang, T., Zhang, M., Lin, W., Lin, Y., Xue, W., Yu, H., ... Zheng, W. (2018, December). Automatic tuning of the community atmospheric model (CAM5) by using short-term hindcasts with an improved downhill simplex optimization method. *Geoscientific Model Development*, *11*(12), 5189–5201. doi: 10.5194/gmd-11-5189-2018

Zhao, C., Liu, X., Qian, Y., Yoon, J., Hou, Z., Lin, G., ... Bao, J. (2013, November). A sensitivity study of radiative fluxes at the top of atmosphere to cloud-microphysics and aerosol parameters in the community atmosphere model CAM5. *Atmospheric Chemistry and Physics*, *13*(21), 10969–10987. doi: 10.5194/acp-13-10969-2013