Supervised Calibration and Uncertainty Quantification of Subgrid Closure Parameters using Ensemble Kalman Inversion

by

Adeline Hillier

B.S. Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 2022

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author . . . .
Department of Electrical Engineering and Computer Science
May 6, 2022

Certified by . . . .
Raffaele Ferrari
Cecil & Ida Green Professor of Oceanography
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Supervised Calibration and Uncertainty Quantification of Subgrid Closure Parameters using Ensemble Kalman Inversion

by

Adeline Hillier

Submitted to the Department of Electrical Engineering and Computer Science
on May 6, 2022, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Data-driven approaches are increasingly being used to identify and remove structural biases in dynamical models for real-world systems. However, because model updates alter the dependency of a model on its free parameters, evidence about structural biases is often muddied by the variable influences of inadequately-tuned parameters on the model solution. We elaborate a framework for model development that combines calibration, sensitivity analysis, and uncertainty quantification of free parameters to shed light on where structural biases are likely to exist in a model, and where the model may be unnecessarily complex. The approach is useful for general applications because it is easy to implement, derivative-free, robust against model instabilities, and computationally inexpensive, requiring a modest number of model evaluations. A diffusive closure for turbulence penetrated by air-sea fluxes of the ocean surface, presently called the “Convective Turbulent Kinetic Parameterization," is developed as a testbed for and proof-of-concept for the approach. Modifications to the traditional Ensemble Kalman Inversion [1] algorithm are devised to improve convergence during the calibration phase of this process. Further, the Calibrate Emulate Sample [2] framework for uncertainty quantification is validated with modifications.

Thesis Supervisor: Raffaele Ferrari
Title: Cecil & Ida Green Professor of Oceanography

Contents

Part I

Introduction

From the spread of tumor cells to the swirls of a tornado, real-world systems whose laws are time-invariant can be represented mathematically as sets of differential equations that relate unresolved variables and resolved variables by unknown constants called free parameters. Dynamical models can be integrated forward in time to reveal future states of the system—a task that, when taken up by computers, lends the ability to preview and probe possible futures sooner than they may arrive. However, most dynamical models are approximate, due not only to the numerical error implicit in the scheme used to integrate them, but because they are structurally biased in that they reflect incomplete knowledge, often assuming false relationships between the relevant variables and the uncertain free parameters. Additionally, whether the parameters reflect universal constants or summarize the collective contributions of many local conditions, there will almost inevitably be error due to the parameters being improperly chosen. As such, accurate models can take years to develop by hand.

This project leverages computational tools to expedite model development, specifically for applications in which model gradient information is unavailable and uncertainty quantification is needed. It begins with the formulation of the development problem as one of supervised learning of free parameters in a Bayesian inverse problem framework, combining sensitivity analysis and parametric uncertainty quantification to identify where uncertainty is being introduced into the model, and where the model can be simplified. We focus on a particular model development application in climate modeling: developing a sub-grid parameterization for vertical turbulent fluxes in the Ocean Surface Boundary Layer, (OSBL) i.e. a mathematical model that captures the physics of turbulence that develops in the upper ocean in response to blowing winds, heat loss, and evaporation.

We propose several modifications to the traditional Ensemble Kalman Inversion algorithm [1] that were found to enable automatic tuning of the free parameters of the aforementioned sub-grid parameterization. We demonstrate that introducing these changes leads to improved in- and out-of-sample performance of the model when calibrated against high-resolution numerical simulations. Following parameter optimization with EKI, the samples generated by EKI are used to perform a variance-based sensitivity analysis on the parameters to gauge how uncertainty in the model output is partitioned among the input parameters.

Lastly, parametric uncertainty is quantified using Markov Chain Monte Carlo (MCMC) in accordance with the Emulate and Sample steps of a Calibrate Emulate Sample framework.

0.1 Background

Developing accurate numerical models is a problem at the intersection of many domains. On the one hand, it is a problem of understanding the laws of the system, extrapolating relationships between variables and conforming to known universal constraints, and on the other it is a problem of assimilating observational data to exploit observed patterns that are as yet unexplained by theory-based models. Historically, development of parameterizations of vertical mixing in the OSBL has been purely theory-driven. As yet, these have been limited in fidelity by the potentially problematic assumptions that they rely on—for example, that diffusion is purely down-gradient (i.e. that turbulence only moves against concentration gradients), or that the mixed layer is perfectly mixed (i.e. that the temperature profile is exactly vertical). It is well understood that existing models are biased and fail to reproduce real climate dynamics over large time windows. This may be attributed to a historical lack of available data for robust evaluation or calibration; observational data was usually limited in spatial or temporal resolution. As such, ocean parameterizations have typically been tuned by hand or designed to be simple, to allow physics-informed guessing of a few rather than many different parameter values. All said, the complexity of the problem has thus far eluded precise analytic descriptions from modelers. Meanwhile, as high-resolution simulations and experimental observations become easier to obtain and faster to process, the field has opened to data-driven approaches to model discovery as an alternative to theory-based modeling. Sub-grid parameterizations are being discovered by neural networks to replace physics-based parameterizations altogether [3], [4]. Using a physics-informed approach that directly addresses the boundary layer problem under investigation, Ramadhan et al. embed and train neural networks within fluid dynamical equations averaged over patches of ocean attempting to express unresolved turbulence at scales smaller than the patch without having to guess at pencil-to-paper equations as the existing models do [5]. Meanwhile, attempts are being made to “discover" governing equations of fluid dynamics using a sparsity-enforcing modification of ridge regression to identify and match coefficients to individual terms in a library of candidate linear and nonlinear terms that might appear in the governing equations [6].

Although machine learning models can be constrained to obey physical principles such as conservation laws, these models tend to have a large memory footprint and, because they can only memorize patterns in the data they are trained on, likely need to be re-trained any time they are introduced to new climatological settings (or whenever the spatial and/or temporal resolution is adjusted). Such models promise neither to extrapolate nor to offer insight into parametric uncertainty. This project takes the perspective that outsourcing model development entirely to machines is risky and therefore ought to be reserved, at least for now, for model evaluation or residual models. Theory-based parameterizations, on the other hand, because they are tethered to the known physics and involve only a few rather than many degrees of freedom (parameters), should theoretically be generalizable and robust against overfitting compared to data-dependent approaches. A parameterization that comprises a comfortable number of degrees of freedom in the form of free parameters can be theory-based–and therefore computationally inexpensive, faithful to known physics, and interpretable–while still taking advantage of the existing wealth of observational data by being optimized against it. In any case, the data assimilation problem is one of learning parameters.

To make the assimilation problem explicit, certain uncertainty-permitting assumptions need to be made about the relationship between the model and data. A common model for this relationship is provided by the Bayesian inverse problem, where the forward model $\mathcal{G}$ is modeled as reproducing the data $\boldsymbol{y}$ when applied to some optimal parameters $\boldsymbol{\theta}^*$, taking into account some uncertainty on the data, $\boldsymbol{\eta}$: [7]

$$\boldsymbol{y} = \mathcal{G}(\boldsymbol{\theta}^*) + \boldsymbol{\eta} \tag{1}$$

Typically the forward model $\mathcal{G}$ will depend on several variables other than the free parameters $\boldsymbol{\theta}^*$, such as initial and boundary conditions; however, for the purposes of a single experiment, these variables do not vary and indeed, the forward map output can include a combination of results from various settings of these other variables. When the parameters and observation noise in (0.3) are modeled as parameterized distributions, this relationship becomes an explicit framework within which we can apply Bayesian inference to infer model parameters from experimental data. For parameter estimation, it provides a natural least-squares loss function with which to assess model performance; for uncertainty quantification, it provides a natural closure for the conditional likelihood of a parameter given the data.

0.1.1 Parameter Estimation

Tuning is an often-overlooked step in the model development process [8]–[10]. It can be extremely challenging in high-dimensional settings given that robust parameter estimates can usually only be achieved if parameters are tuned jointly rather than in isolation. However, this step cannot be ignored, as apparent model biases are easily misattributed to structural errors if model parameters have not been optimally tuned, sometimes driving unnecessary and wasteful structural changes to the parameterization scheme or increases in model resolution [11]. If qualitative evidence of model-data misfit is to be used to glean evidence about a model's current structural deficiencies and streamline improvements to the model's physical formulation, such evidence ought to be minimally corrupted by the influence of improperly tuned parameters.

Traditional gradient descent methods for parameter tuning iteratively update a parameter estimate $\boldsymbol{\theta}$ from an initialization $\boldsymbol{\theta}_0$ so that it moves down-gradient toward a local loss minimum according to a time-discretized version of $\dot{\boldsymbol{\theta}} = -\nabla\Phi(\boldsymbol{\theta})$, where $\Phi(\boldsymbol{\theta})$ is the model-data misfit function to be minimized. Variants of gradient descent use momentum-augmented methods, or adaptive learning rate algorithms, to accelerate convergence. Alternative gradient-based methods such as Quasi-Newton and Gauss-Newton methods improve convergence in convex regions of parameter space by taking into account estimated second derivatives of the objective. The two main problems with such methods are that (1) they rely on the gradient information, or adjoints, of the forward map G, which is often difficult or costly to compute, and (2) it can be hard to find a good starting point $\boldsymbol{\theta}_0$ for the dynamic. It is therefore desirable to find computationally cheap approaches that don't require adjoints. Several such derivative-free (“zeroth-order") optimization approaches are already being applied to infer unknown parameters from noisy observations in a climate modeling context. Williamson et al. propose history matching (“iterative refocusing") as an approach to whittle down the parameter space to regions where the model solution meets a certain plausibility threshold. However, this method relies on repeated reuse of an emulator for the forward map, which is often unreliable or costly to obtain. In a very similar approach arising from a very similar philosophy of model development to that adopted in this thesis, Golaz et al. used ensemble-based Nelder–Mead optimization to calibrate the parameters of a sub-grid closure for the cloud boundary layer, with the observations provided by a suite of six LES [12]. Other derivative-free approaches include population-based meta-heuristic algorithms such as particle swarm optimization, genetic algorithms, and differential evolution, as well as single-solution meta-heuristic algorithms like simulated annealing. This thesis concerns itself with a class of interacting agent-based Bayesian parameter estimation algorithms adapted from the Ensemble Kalman (EK) filter originally used for state estimation [1]. Ensemble Kalman Inversion (EKI) is a derivative-free optimization algorithm whereby an ensemble of particles is distributed throughout the parameter space through random sampling from Gaussian priors and then iteratively updated until the particles converge toward a loss minimum. The individual particles perform an approximate form of gradient descent in which the derivatives are approximated by ensemble differences [1]. In addition to being gradient-free, EKI is advantageous compared to alternative approaches for several reasons. First, it's an ensemble method, which means it's inherently parallelizable and has convergence advantages for when the loss landscape is noisy. It can also be easily modified to accommodate the occurrence of parameters that cause the model to go numerically unstable, where other algorithms might be stopped in their tracks. A second advantage is that EKI's numerical cost is known ahead of computation, given the number of algorithm iterations is not contingent on some inherent stop condition. A final advantage, subject to approximations, is that the regularized variant of EKI can be leveraged for uncertainty quantification.

Although the EKI algorithm is superficially quite simple and has been characterized theoretically at length [1], [13]–[15], there remain a large number degrees of freedom available to users to improve EKI's performance. We elaborate a practical approach to using EKI for general applications, where CATKE (short for convective adjustment turbulent eddy kinetic energy) is used a testbed for the approach. CATKE is a model representing the rates of mixing induced by turbulence arising from winds, cooling or precipitation at the ocean surface. The model depends on 17 free parameters whose values are not known a priori. Developing a general approach for EKI amounts to detailing and validating suggested solutions for such questions as how to choose and constrain EKI priors, how to characterize the uncertainty in synthetic observations, in order to estimate a covariance matrix for observational noise, how to handle numerical instabilities in the model solution, and how to optimize the EKI time step size for performance in nonlinear settings.

0.1.2 Uncertainty Quantification via CES

In applications such as embedded control systems, epidemic forecasting and climate modeling that seek to outpace the relevant time scales of evolution of the dynamics of interest, data assimilation tasks must often be carried out in real time or “online." In such applications it is not feasible or robust to repeatedly re-calibrate the model on all evidence accumulated thus far; we instead seek to estimate probability distributions over possible parameter values providing a natural way to incorporate new observational evidence into an existing model. While EKI is not constructed to learn uncertainty but rather to reach consensus around a single parameter value that makes the model-data misfit small, EK methods are currently being applied for uncertainty quantification within the recently proposed Calibrate Emulate Sample (CES) framework [2]. A noisy variant of EKI known as Ensemble Kalman Sampling constitutes the first step (“Calibrate") in the CES procedure–to zero in on the neighborhood of optimal parameter values and produce samples of the parameter-data map $\mathcal{G}(\boldsymbol{\theta})$ in that neighborhood. In the second stage (“Emulate"), a Gaussian Process is trained to interpolate the parameter-data pairs generated from the EKS sampling so that further calls to the expensive forward map are not needed. In the final stage (“Sample"), MCMC is used to approximate the parameter posteriors by sampling in the general region of the optimal parameters, where the parameter-to-data map is replaced by the Gaussian Process emulator evaluated on the given parameter input. Replacing the parameter-to-data map in the sampling stage makes the otherwise expensive uncertainty quantification computationally tractable for large problems. The interpolation function provided by the Gaussian Process is also potentially smoother than the actual data that it interpolates, facilitating the MCMC traversal of the loss landscape, hopefully leading to more accurate posterior estimates. To validate the approach, Dunbar et al. [16] apply CES to calibrating and estimating posteriors on the parameters of an idealized general circulation model of the atmosphere in a perfect-model setting, where the model used to generate truth data is identical to the model to be tuned, but is blind to the true parameter values during the tuning procedure.

Part II

Methods

In this thesis we focus on a climate application for model development, elaborated in section 0.2. In section 0.2.2, we describe the set of high-resolution numerical simulations used as reference “ground truth" data against which model accuracy is assessed. The simulations capture a diversity of physical settings plausibly encountered in the real world. In section 0.3, the model and observations are construed in the Bayesian inverse problem context. In section 0.4, the tuning algorithm is described at length. Implementation details and modifications to the algorithm are motivated in the context of the climate modeling problem at hand. Finally, the procedure for estimating parametric uncertainty using Gaussian processes and MCMC is described in section 0.5.

0.2 Model and Observations

0.2.1 Ocean Surface Boundary Layer Column Models

Climate models are inherently approximate in that they rely on subgrid-scale models to approximate dynamics that occur at scales finer than the model grid. For grid cells in the topmost 100 to 500 meters of the ocean, also known as the ocean surface boundary layer (OSBL), the collective effects of subgrid-scale turbulent motions, from the smallest micro scales of dissipation to the length of the cell, are typically approximated within one-dimensional “column" models that attempt to capture the time evolutions of horizontally-averaged quantities within the cell without explicitly resolving the flow in three dimensions, given only initial and boundary conditions (Figure 0-1). The resolved variables are typically momentum and tracers such as temperature, salinity, carbon, and dissolved chemicals. The turbulent processes responsible for the vertical redistribution of tracers within the OSBL also regulate the rates at which these tracers are exchanged with the ocean interior below and drawn down from the atmosphere above. In order to exhibit strong extrapolation ability, these models should be able to reproduce dynamics in patches of ocean with varied combinations of boundary conditions, including arbitrary amounts of convective mixing (due to heating/cooling and evaporation/precipitation), shear-driven mixing (due to wind), and rotation. We consider column model configurations that solve the horizontally averaged

Boussinesq equations,

$$\partial_t \overline{\boldsymbol{u}} - f\overline{\boldsymbol{v}} = -\partial_z \, \overline{\boldsymbol{w'u'}} \tag{2}$$

$$\partial_t \overline{\boldsymbol{v}} + f\overline{\boldsymbol{u}} = -\partial_z \, \overline{\boldsymbol{w'v'}} \tag{3}$$

$$\partial_t \overline{\boldsymbol{c_\theta}} = -\partial_z \, \overline{\boldsymbol{w'c'_\theta}} \tag{4}$$

where $\boldsymbol{u}$ is east-west velocity, $\boldsymbol{v}$ is north-south velocity, and $\boldsymbol{c_\theta}$ is conservative temperature. Overlined variables represent averages over the horizontal domain while primed variables represent turbulent fluctuations about the average. Note that in equation (4) the diffusive term, $\kappa\partial_z^2\overline{\boldsymbol{c_\theta}}$ where κ is the molecular thermometric diffusivity, is omitted, leaving just the unresolved advective term, $\partial z\overline{\boldsymbol{w'c'_\theta}}$, because the advective term dominates. Equations 2 - 4 are unclosed because the stress terms $\overline{\boldsymbol{w'u'}}$, $\overline{\boldsymbol{w'v'}}$ and $\overline{\boldsymbol{w'c'_\theta}}$, also referred to as vertical turbulent fluxes, are unresolved. The modeling problem therefore amounts to providing a closure for these unresolved fluxes. The parameterization developed in this thesis models these fluxes as down-gradients of the mean profiles of the respective variables, with the constant of proportionality being an eddy diffusivity which is itself parameterized in terms of resolved variables and free parameters. This closure is presently called the Convective Adjustment Turbulent Kinetic Energy Equation, or “CATKE," and is currently being developed by Gregory Wagner at MIT.

0.2.2 Observations

Optimal values for CATKE's free parameters cannot be derived from first principles and are inferred from data. Depending on the goals of the developers and stage of model development, the data could come in the form of interpolated field observations or high-resolution, idealized numerical simulations, sometimes referred to in this thesis as synthetic observations. In this thesis reference data is provided by a set of three dimensional Large Eddy Simulations (LES) that provide high-fidelity time evolutions of horizontally-averaged velocity and temperature profiles for simulated cells of ocean that are subjected to constant heat ($\overline{\boldsymbol{w'c'_\theta}}_{z=0}$) and wind surface forcings ($\overline{\boldsymbol{w'u'}}_{z=0}$) over the course of several days. Such high-resolution LES are believed to be sufficiently accurate and permit long enough integration times to be useful

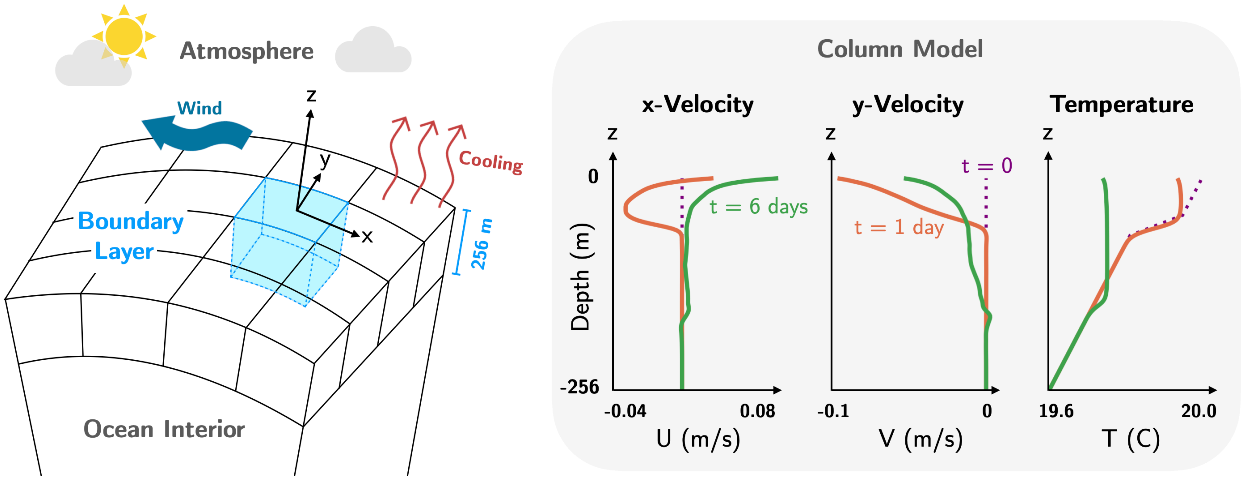

Figure 0-1: Ocean Surface Boundary Layer parameterizations aim to capture unresolved dynamics within 3-D grid cells (e.g. the cell shaded in blue) from the ocean's top-most layer. The dynamics within each given cell are summarized by one-dimensional column models (right), which track the "horizontally-averaged" (averaged over x- and y- across the cell) profiles of velocity and temperature within the cell as they evolve from time $t = 0$. In the example, constant wind stress and cooling at the ocean surface over a course of 6 days manifest in the column model representation as non-zero velocities and gradually-decreasing temperature.

for parameter inference.

In previous work, synthetic observational data was generated in the form of three suites of five high-resolution large eddy simulations–a 2-day suite, a 4-day suite, and a 6-day suite. The five simulations of each suite capture various combinations of surface momentum flux, surface buoyancy flux (positive values indicate cooling), and Coriolis parameter (rotation) values, with a uniform initial condition as summarized in Table 1. The LES parameter values were chosen so that the mixed layer depth reaches 100-150 meters by the end of the simulation (that is, approximately 50% of the vertical domain) so that interactions with the artificial bottom of the cell are inconsequential. In all simulations, the model domain is 512x512x256 meters with 256 grid points in the horizontal and 128 grid points in the vertical, giving 2-meter isotropic grid spacing. The initial condition has three layers, where in each layer the temperature increases linearly with proximity to the surface, and the gradient is smallest in the middle layer. A constant Anisotropic Minimum Dissipation (AMD) closure is used to model the molecular dissipation of tracer fluctuations and dissipation of kinetic energy. To achieve high temporal resolution we set the simulation time step to 10 minutes, giving 144 snapshots of the model solution per physical day.

0.2.3 Implementation

Fluid flow simulations are carried out in the open-source Julia software package Oceananigans.jl [17], which solves fluid flow simulations in 1-, 2- and 3 dimensions using a finite volume discretization on an Arakawa-C staggered grid, where values are located at cell centers and fluxes are located at cell faces [18]. For the single column model with CATKE, Oceananigans implements a semi-implicit backward Euler time discretization scheme to evolve the relevant variables according to (2), (3), and (4). We fix the CATKE model time step at $\Delta t = 5$ minutes while 10-second time steps were used to compute the LES solution. Further, we limit the vertical resolution of the CATKE solution to 8 meters and accordingly coarse-grain the LES solution wherever comparison is needed. CATKE is one of several subgrid closure options implemented in Oceananigans. Fluxes are imposed at the grid surface ($z = 0$m, i.e. the interface between the ocean and atmosphere) and set to zero at the lower boundary ($z = 256$m, i.e. the ocean interior). For the LES, horizontally periodic boundary conditions are enforced. A fifth-order Weighted Essentially Non-Oscillatory (WENO) advection scheme for tracers (temperature) is used along with a 3rd order Runge-Kutta scheme for time stepping. Further details on simulation numerics can be found in the documentation for Oceananigans.jl [17].

0.3 Bayesian Inverse Problem

To recapitulate the previous section, CATKE's parameters are to be inferred via quantitative comparison between the solutions of a low-resolution (32 vertical grid points) single column model simulation that use CATKE and the horizontally-averaged solutions of high-resolution (256x256x128) LES. The LES solutions, taken together, are treated as the “synthetic observation" that the coarse simulation “model" is calibrated to, where all of the parameters of the coarse “model" are contained within the constituent sub-grid closure, CATKE.

The inverse problem is to find the unknown parameters $\boldsymbol{\theta}^*$ that bring the model into agreement with the truth in the sense of (1). The specific statistics that the forward model $\mathcal{G}$

computes have to be selected for each use case to provide a concise and unbiased summary of the complex model solution, containing a balanced set of features most relevant for assessing the model-data discrepancy. In the OSBL context, this summary could be, for example, a vector of concatenated velocity and temperatures profiles at all or some time steps of the CATKE solution. In this work, the observation is obtained by normalizing all of the 1-D spatial time series data from days 1, 2, 3 and 4 of the 4-day observation suite and concatenating it into a single column vector. The corresponding forward map output for a given parameter set $\boldsymbol{\theta}$ is obtained by initializing variables $\overline{\boldsymbol{u}}$, $\overline{\boldsymbol{v}}$, $\overline{\boldsymbol{w}}$, $\overline{\boldsymbol{c_\theta}}$ and $\overline{\boldsymbol{e}}$–the turbulent kinetic energy $\overline{\boldsymbol{e}} = 1/2(\overline{\boldsymbol{u}'^2 + \boldsymbol{v}'^2 + \boldsymbol{w}'^2})$–in the column model to the LES solution at 0.5 days and evolving the model to day 4, then applying an identical procedure to obtain a single column vector from the model solution. Identical boundary conditions, such as surface forcings $\overline{\boldsymbol{w'c'_\theta}}_{z=0}$ and $\overline{\boldsymbol{w'u'}}_{z=0}$, are imposed on the model domain as were imposed to generate the reference LES solutions. In the strong cooling scenario, the trivial velocity information is omitted from the forward map and observation; in the rotation-free scenario, the north-south velocity information is ignored. This leaves 17 variables to be recorded across the 5 simulations; thus the forward map contains $2176 = 4 \times 17 \times 32$ entries representing 4 time snapshots of 17 length-32 profiles across 5 simulations. Thus $\boldsymbol{y}$, $\mathcal{G}(\boldsymbol{\theta})$, and $\boldsymbol{\eta}$ all inhabit the space $\mathbb{R}^{2176}$. Importantly, because CATKE does not explicitly set out to resolve $\overline{\boldsymbol{e}}$ to a high degree of physical accuracy but rather as a proxy for the useful information that $\overline{\boldsymbol{e}}$ contains, the normalized profiles $\overline{\boldsymbol{e}}$ are re-scaled by a factor of 0.01 when incorporated into the forward map.

Statistical descriptions of $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$ are the two design parameters not yet specified in our articulation of the Bayesian inverse problem. To be motivated in section 0.4.1, several simplifying assumptions are made: that the observation noise $\boldsymbol{\eta}$ is drawn from a Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}_{\boldsymbol{y}})$, that the prior distribution of $\boldsymbol{\theta}$ is also Gaussian $\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Gamma}_{\boldsymbol{\theta}})$, and that $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$ are a priori independent. The choice of $\boldsymbol{\Gamma}_{\boldsymbol{\theta}}$ will be explained in 0.4.2. In order to estimate $\boldsymbol{\Gamma}_{\boldsymbol{y}}$, we trilinearly interpolate LES solutions with 2 meter, 4 meter and 8 meter vertical resolutions to the desired 16 meters (32 grid points) and compute the Bessel-corrected covariance of the observation vectors obtained from the respective solutions. Alternative formulations could represent $\boldsymbol{\Gamma}_{\boldsymbol{y}}$ as a scaled Identity matrix or use the scale of turbulent fluctuations computed in the LES solutions as a proxy for the uncertainty of the mean states for the relevant variables. In the upcoming section Ensemble Kalman Inversion is described with application to minimizing the model-data misfit for CATKE.

0.4 Ensemble Kalman Inversion

The purpose of calibration is to bring a model into agreement with observational data of some kind. In its present formulation, CATKE is formulated with 17 scalar parameters. In this section we describe how we employ Ensemble Kalman Inversion (EKI) to tune these parameters. As we will see in section 0.4.1, the EKI process aims to minimize the following regularized least-squares objective [15]

$$\Phi_R(\boldsymbol{\theta}) = \frac{1}{2}\left\|{\boldsymbol{\Gamma}_{\boldsymbol{y}}}^{-\frac{1}{2}}(\boldsymbol{y} - \mathcal{G}(\boldsymbol{\theta}))\right\|^2 + \frac{1}{2}\left\|{\boldsymbol{\Gamma}_{\boldsymbol{\theta}}}^{-\frac{1}{2}}(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})\right\|^2 \tag{5}$$

The first term measures the distance from the CATKE solution to the truth, down-weighted by the uncertainty on the truth. The second term implements Tikhonov regularization [1]. From a Bayesian optimization perspective, Tikhonov regularization is a natural consequence of incorporating prior information with likelihood data; from a classical optimization perspective, it is warranted by the ill-posed nature of the problem: multiple equally-good solutions might exist, and penalizing parameters that stray far from the estimated priors allows for more efficient parameter search, at the cost of possible overfitting to the priors.

EKI's grand limitation as an optimization approach is that the ensemble mean is only guaranteed to converge by theory if the forward map is linear and the Gaussian assumptions hold [13], which is almost never the case. Improving EKI's convergence in nonlinear settings is an active area of research [19]. However, there is evidence that EKI already succeeds in nonlinear settings in practice. For example, EKI has demonstrated some success in optimizing the weights of a neural network [15]. Chada et al. draw comparisons between EKI parameter updates and Gauss-Newton optimization to show that EKI can converge in finite iterations given appropriate adjustments to the noise covariance matrix at each step, even when the objective is non-convex [19].

0.4.1 Detailed Formulation

Parameter Estimation as a Bayesian Inverse Problem

The inverse problem as outlined by (1) is to find the unknown $\boldsymbol{\theta}^*$ that permit this equality between the model output and the truth. Given the aforementioned normality assumptions, the density of $\boldsymbol{\eta}$ and the prior density are multivariate normal distributions.

$$\pi_{\boldsymbol{\eta}}(\boldsymbol{\eta}) \propto e^{-\frac{1}{2}(\boldsymbol{\eta}-\boldsymbol{\mu_\eta})^T \boldsymbol{\Gamma}_{\boldsymbol{y}}^{-1}(\boldsymbol{\eta}-\boldsymbol{\mu_\eta})} = e^{-\frac{1}{2}||\boldsymbol{\eta}-\boldsymbol{\mu_\eta}||^2_{\boldsymbol{\Gamma_y}}} \tag{6}$$

$$\pi_0(\boldsymbol{\theta}) \propto e^{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\mu_\theta})^T \boldsymbol{\Gamma}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu_\theta})} = e^{-\frac{1}{2}||\boldsymbol{\theta}-\boldsymbol{\mu_\theta}||^2_{\boldsymbol{\Gamma_\theta}}} \tag{7}$$

where we have used the notation $\|\boldsymbol{x}\|^2_{\boldsymbol{A}} := \left\|\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{x}\right\|^2 = (\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{x})^T(\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{A}^{-1}\boldsymbol{x}$ for the Mahalanobis norm given symmetric, positive definite $\boldsymbol{A}$. The posterior density, $\pi(\boldsymbol{\theta}|\boldsymbol{y})$, is proportional to the product of the likelihood $\pi(\boldsymbol{y}|\boldsymbol{\theta})$ and prior $\pi_0(\boldsymbol{\theta})$. Note that $\boldsymbol{y}$ conditioned on $\boldsymbol{\theta}$ is distributed as $\boldsymbol{\eta} = \boldsymbol{y} - \mathcal{G}(\boldsymbol{\theta})$. Therefore

$$\pi(\boldsymbol{y}|\boldsymbol{\theta}) = \pi_{\boldsymbol{\eta}}(\boldsymbol{y} - \mathcal{G}(\boldsymbol{\theta})) \tag{8}$$

Combining equations 7 and 8, we have, by Bayes' theorem,

$$\begin{aligned} \pi(\boldsymbol{\theta}|\boldsymbol{y}) &\propto \pi_{\boldsymbol{\eta}}(\boldsymbol{y}|\boldsymbol{\theta})\pi_0(\boldsymbol{\theta}) \\ &\propto e^{-\frac{1}{2}||\boldsymbol{y}-\mathcal{G}(\boldsymbol{\theta})||^2_{\boldsymbol{\Gamma_y}}} e^{-\frac{1}{2}||\boldsymbol{\theta}-\boldsymbol{\mu_\theta}||^2_{\boldsymbol{\Gamma_\theta}}} \\ &= e^{-\Phi_R(\boldsymbol{\theta})} \end{aligned} \tag{9}$$

where $\Phi_R(\boldsymbol{\theta})$ is the negative log density

$$\Phi_R(\boldsymbol{\theta}) = \frac{1}{2}||\boldsymbol{y} - \mathcal{G}(\boldsymbol{\theta})||^2_{\boldsymbol{\Gamma_y}} + \frac{1}{2}||\boldsymbol{\theta} - \boldsymbol{\mu_\theta}||^2_{\boldsymbol{\Gamma_\theta}} \tag{10}$$

also known as the regularized least-squares objective, where the subscript R denotes regularization. Note that maximizing the posterior density (9) is the same as minimizing (10); the minimizer of the objective is the MAP estimator of (9).

Ensemble Kalman Filtering (EnKF)

EKI has its roots in an algorithm called Ensemble Kalman Filtering (EnKF). Innovations meant to overcome the shortcomings of EKI are often first derived in the context of the EnKF, which continues to see widespread use in fields such as numerical weather prediction that seek to forecast the trajectories of large state variables. The EnKF is designed for sequential data assimilation problems, whereby a state $\boldsymbol{x}$ evolves in time according to some ODE

$$\partial_t \boldsymbol{x} = f(\boldsymbol{x}, t) \quad \text{(Dynamic)} \tag{11}$$

starting from an initial condition $\boldsymbol{x}(t_0)$ that is uncertain and is therefore modeled as a random variable with PDF $\pi_0(\boldsymbol{x})$. Because the initial condition is uncertain, all subsequent realizations of the state are also uncertain and have some PDF $\pi(\boldsymbol{x}, t)$ associated with them. Absent any further information about $\boldsymbol{x}$, there is an analytical relation for how the PDF evolves in time and it is given by the continuity equation $\partial_t \pi = -\mathrm{div}(\pi f)$. However, there is new information that arrives at various discrete time steps in this evolution so that π has to be modified at each iteration. This information comes in the form of "observations" $\boldsymbol{y}(t)$ that are assumed to be linearly transformed measurements of $\boldsymbol{x}$, taking into account some Gaussian measurement error $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}_{\boldsymbol{y}})$.

$$\boldsymbol{y}(t_q) = \boldsymbol{H}\boldsymbol{x}(t_q) + \boldsymbol{\eta} \quad \text{for } q = 1, 2, ..., M \quad \text{(Observation)}$$

where $\boldsymbol{H} \in \mathbb{R}^{d_y \times d_x}$ is a linear operator and M is the number of observations. This new data has to be assimilated into the projection of $\pi(\boldsymbol{x}, t)$ each time a new observation is made, hence the description of the problem as one of 'sequential data assimilation'. The solution amounts to iteratively evolving $\pi(\boldsymbol{x}, t_q)$ to $\pi(\boldsymbol{x}, t_{q+1})$ according to the continuity relation but with adjustments to $\pi(\boldsymbol{x}, t_{q+1})$ that incorporate $\boldsymbol{y}(t_{q+1})$ at each step. The former updates, where the underlying dynamics are evolved, constitute the "evolution" or "forecast" stage of the algorithm, while the iterative, measurement-informed adjustments constitute the "update" or "analysis" stage. To make the problem tractable $\pi(\boldsymbol{x}, t)$ usually needs to be approximated with some parameterized statistical model $\rho(\boldsymbol{x}|\boldsymbol{z}(t))$. Then the problem is reduced one of tracking the parameters $\boldsymbol{z}(t) \in \mathbb{R}^{d_z}$; that is, $\boldsymbol{z}(t)$ replaces $\pi(\boldsymbol{x}, t)$ as the variable which is explicitly evolved.

Consider the special case that the ODE (11) is linear and $\rho(\boldsymbol{x}|\boldsymbol{z}(t))$ is Gaussian with the parameters $\boldsymbol{z}(t)$ representing the mean $\boldsymbol{\mu_x}(t)$ and covariance $\boldsymbol{P}(t)$. Then the mean and covariance updates at each time t_q during the prediction stage would be readily given in closed form by the Kalman filter, representing the result of an application of Bayes' rule to the forecasted priors on $\boldsymbol{x}$ based on the new evidence. The update for time t_q to t_{q+1} is:

$$\boldsymbol{\mu_x}(t_{q+1}) = \boldsymbol{\mu_x}(t_q) + \boldsymbol{K}(t_q)(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\mu_x}(t_q)) \quad \text{(KF analysis)} \tag{12}$$

$$\boldsymbol{P}(t_{q+1}) = (\boldsymbol{I} - \boldsymbol{K}(t_q)\boldsymbol{H})\boldsymbol{P}(t_q) \quad \text{(KF analysis)} \tag{13}$$

where $\boldsymbol{K}(t_q)$ is the Kalman gain defined as $\boldsymbol{K}(t_q) = \boldsymbol{P}(t_q)\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}(t_q)\boldsymbol{H}^T + \boldsymbol{\Gamma_y})^{-1}$. This can be rewritten to impart the intuition that the update gives lower or higher priority to the measurement compared to the forecast depending on the relative uncertainty $\boldsymbol{\Gamma_y}$ of the observation and that of the dynamic (11). Though the Kalman filter is an unbiased estimator in the present context (meaning it is accurate in expectation), its applicability is limited by the linearity assumption on (11). Additionally, if the state is many-dimensional, storing the covariance matrices and carrying out the necessary matrix inversions in the Kalman updates can become intractable [20]. The EnKF was developed to partially address both of these issues.

For each snapshot q in time, EnKF uses an ensemble of J state particles $\{\boldsymbol{x}_q^{(j)}\}_{j=1}^J$ that constitute independent solutions to (11) and together stand in for a true analytic Gaussian description of the distribution over possible states. The resulting statistical model is then formulated no longer as $\mathcal{N}(\boldsymbol{\mu_x}, \boldsymbol{P})$ but rather as a sum of Dirac delta functions

$$\pi_q^J(\boldsymbol{x}) = \sum_{j=1}^{J} \delta(\boldsymbol{x} - \boldsymbol{x}_q^{(j)}(t)) \tag{14}$$

parameterized by particle locations $\{\boldsymbol{x}_q^{(j)}\}$. It is in this sense that the EnKF is a Monte Carlo implementation of the Bayesian update problem. The empirical mean and covariance of the ensemble distribution are given by

$$\overline{\boldsymbol{x}}_q = \frac{1}{J}\sum_{j=1}^{N}\boldsymbol{x}_q^{(j)} \quad \boldsymbol{C}^{\boldsymbol{xx}} = \frac{1}{J-1}\sum_{j=1}^{J}(\boldsymbol{x}_q^{(j)} - \overline{\boldsymbol{x}}_q)(\boldsymbol{x}_q^{(j)} - \overline{\boldsymbol{x}}_q)^T \tag{15}$$

During the forecast stage, the EnKF particles are evolved according to (11). The magic of the algorithm is in how the analysis step is carried out; how does one update the particles such that their resulting distribution resembles that which the Kalman filter would have projected? There are stochastic and a deterministic solutions to this problem [20]–[22]. In the stochastic formulation, the particles are updated as

$$\boldsymbol{x}_{\boldsymbol{q+1}}^{\boldsymbol{(j)}} = \boldsymbol{x}_{\boldsymbol{q}}^{\boldsymbol{(j)}} + \boldsymbol{K}_{\mathrm{emp}}(\boldsymbol{y}_q + \boldsymbol{\eta}_q^{(j)} - \boldsymbol{H}\boldsymbol{x}_q^{(j)}) \ \text{ (EnKF analysis)} \tag{16}$$

where $\boldsymbol{K}_{\mathrm{emp}}$ is the estimated (empirical) Kalman gain,

$$\boldsymbol{K}_{\mathrm{emp}} = \boldsymbol{C}^{\boldsymbol{xx}}\boldsymbol{H}^*(\boldsymbol{H}\boldsymbol{C}^{\boldsymbol{xx}}\boldsymbol{H}^* + \boldsymbol{\Gamma}_{\boldsymbol{y}})^{-1} \tag{17}$$

with $\boldsymbol{H}^*$ denoting the adjoint of $\boldsymbol{H}$. The “stochasticity" in (16) comes from the additive observation perturbation $\boldsymbol{\eta}$. Compared to the deterministic alternative, this approach has been suggested to be more stable against Gaussianity violations in the underlying state [21]. The artificial perturbation can also help break the subspace property and potentially counteract a known artefact [23] of EnKF that the empirical covariance tends to underestimate the true covariance. However, the approach has the potential to exacerbate sampling error for small ensembles, and is thus sometimes abandoned in favor of deterministic approaches, including more complicated Sequential Monte Carlo approaches that incorporate adjustable weights into the ensemble distribution formulation $\pi_q^J(\boldsymbol{x})$, turning it into a weighted sum [24], [25]. One such approach is directly contrasted with the EnKF for inverse problems (EKI) in [26].

EnKF to EKI: from State Estimation to Parameter Estimation

The idea to reinterpret the EnKF method as an approach to solve inverse problems came from the oil industry. The underlying state $\boldsymbol{x}$ which we seek to recover is formulated as one which evolves according to a discrete dynamic [1]

$$\boldsymbol{x}_{n+1} = \begin{pmatrix} \boldsymbol{\theta}_n \\ \mathcal{G}(\boldsymbol{\theta}_n) \end{pmatrix} \quad \text{(Dynamic)} \tag{18}$$

where $n \in \mathbb{Z}^+$. Each realization of the state is assumed to have a Gaussian PDF associated with it, and as before the distribution at time n is approximated by an ensemble $\{\boldsymbol{x}_n^{(j)}\}_{j=1}^J$. The empirical mean and covariance of the ensemble can be decomposed into their constituent parts,

$$\overline{\boldsymbol{x}}_n = \frac{1}{J-1}\sum_{j=1}^{J} \boldsymbol{x}_n^{(j)} = \begin{pmatrix} \overline{\boldsymbol{\theta}}_n \\ \overline{\boldsymbol{g}}_n \end{pmatrix}, \quad \boldsymbol{C}_n^{\boldsymbol{xx}} = \begin{pmatrix} \boldsymbol{C}_n^{\boldsymbol{\theta\theta}} & \boldsymbol{C}_n^{\boldsymbol{\theta g}} \\ \boldsymbol{C}_n^{\boldsymbol{\theta g}T} & \boldsymbol{C}_n^{\boldsymbol{gg}} \end{pmatrix} \tag{19}$$

where $\boldsymbol{C}_n^{\boldsymbol{ab}} = \frac{1}{J-1}\sum_{j=1}^J (\boldsymbol{a}_n^{(j)} - \overline{\boldsymbol{a}}_n)(\boldsymbol{b}_n^{(j)} - \overline{\boldsymbol{b}}_n)^T$ for general $\boldsymbol{a}$, $\boldsymbol{b}$, and superscript $\boldsymbol{g}$ represents the collection of forward map outputs $\{\mathcal{G}(\boldsymbol{\theta}_n^{(j)})\}_{j=1}^J$. As before, the observations are modeled as a noisy linear transformation of the state [1],

$$\boldsymbol{y}_n = \boldsymbol{H}\boldsymbol{x}_n + \boldsymbol{\eta}_n \quad \text{(Observation)} \tag{20}$$

where $\boldsymbol{H} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{I} \end{pmatrix}$ is a linear projection operator that picks out the second component of $\boldsymbol{x}_n$ thus mapping $\mathbb{R}^{d_y+d_\theta}$ to $\mathbb{R}^{d_y}$, and $\{\boldsymbol{\eta}_n\}_{n\in\mathbb{Z}^+}$ are i.i.d. realizations of the observation uncertainty variable $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma}_{\boldsymbol{y}})$. Applying (19) the empirical Kalman gain (17) becomes

$$\boldsymbol{K}_{\text{emp},\boldsymbol{n}} = \begin{bmatrix} \boldsymbol{C}_n^{\boldsymbol{\theta g}} \\ \boldsymbol{C}_n^{\boldsymbol{gg}} \end{bmatrix} (\boldsymbol{C}_n^{\boldsymbol{gg}} + \boldsymbol{\Gamma}_{\boldsymbol{y}})^{-1} \tag{21}$$

leading to the state update analogous to (16), [1]

$$\boldsymbol{\theta}_{n+1}^{(j)} = \boldsymbol{\theta}_n^{(j)} + \boldsymbol{C}_n^{\boldsymbol{\theta g}}(\boldsymbol{C}_n^{\boldsymbol{gg}} + \boldsymbol{\Gamma}_{\boldsymbol{y}})^{-1}(\boldsymbol{y} + \boldsymbol{\eta}_n^{(j)} - \mathcal{G}(\boldsymbol{\theta}_n^{(j)})) \quad \text{(EKI Analysis)} \tag{22}$$

which using the definition of $\boldsymbol{C}^{\boldsymbol{\theta g}}$ can be rewritten to convey that the updated ensemble remains confined to the linear span of the initial ensemble despite that the update coefficients may be nonlinear, [1]

$$\boldsymbol{\theta}_{n+1}^{(j)} = \boldsymbol{\theta}_{n}^{(j)} + \frac{1}{J}\sum_{k=1}^{J}\langle \mathcal{G}(\boldsymbol{\theta}_n^{(k)}) - \overline{\mathbf{g}},\ (\boldsymbol{C}_{n+1}^{\boldsymbol{gg}} + \boldsymbol{\Gamma}_{\boldsymbol{y}})^{-1}(\boldsymbol{y} + \boldsymbol{\eta}_n^{(j)} - \mathcal{G}(\boldsymbol{\theta}_n^{(j)})\rangle\ \boldsymbol{\theta}_n^{(k)} \tag{23}$$

Connection to Sequential Monte Carlo

Note that time in the sense of t_q in (0.4.1) has no meaning for EKI because observations for EKI are not chronologically evolving. EKI adopts a new interpretation of pseudo-time corresponding not to when new measurements are made but rather when particle updates are made given the fixed observation $\boldsymbol{y}$. Equation (9) carries the prior to the posterior in a single application of Bayes' rule, based exclusively on information from a single forward run for each ensemble member. A more incremental approach, known in the Sequential Monte Carlo (SMC) literature [7], [22], [25], [26] as likelihood tempering, can be taken so that $N > 1$ updates are taken to arrive at the posterior:

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) \propto \pi_0(\boldsymbol{\theta}) \prod_{n=1}^{N} [\pi_{\boldsymbol{\eta}}(\boldsymbol{y}|\boldsymbol{\theta})]^{\frac{1}{N}} \tag{24}$$

$$\propto \pi_0(\boldsymbol{\theta}) \prod_{n=1}^{N} e^{-\frac{1}{2N}||\boldsymbol{\Gamma}_{\boldsymbol{y}}^{-\frac{1}{2}}(\boldsymbol{y}-\mathcal{G}(\boldsymbol{\theta}))||^2} \tag{25}$$

$$= \pi_0(\boldsymbol{\theta}) \prod_{n=1}^{N} e^{-\frac{1}{2}||(\frac{1}{N}\boldsymbol{\Gamma}_{\boldsymbol{y}})^{-\frac{1}{2}}(\boldsymbol{y}-\mathcal{G}(\boldsymbol{\theta}))||^2} \tag{26}$$

Intuitively, this interpretation is more useful for EKI because it means that when several Kalman updates are applied, information from several forward model ensembles can be incorporated in the process of reaching the approximated posterior. Additionally, it leads to a continuous-time interpretation of EKI that yields theoretical results and permits adaptive time-stepping schemes that allow one to tune the rate of convergence of the ensemble. There is an intuitive justification of (26): applying Bayes' rule N times with the same data entails repeated re-use of data, and the accompanying variance adjustment $\boldsymbol{\Gamma}_{\boldsymbol{y}} \mapsto \frac{1}{N}\boldsymbol{\Gamma}_{\boldsymbol{y}}$ compensates for this re-use. Observe that if the product in (25) does not stop at N and is taken to infinity, the posterior will concentrate on the parameters that minimize Φ_R. This is essentially the strategy that EKI takes to pursue the loss-minimizing parameters. We use an EKI variant where a flexible “time-step" parameter h replaces $\frac{1}{N}$ as the variance inflation parameter so that an adaptive algorithm can be used where the ensemble distribution approximates the posterior following N_{iter} iterations satisfying $\sum_{n=1}^{N_{\text{iter}}} h_n = 1$. The associated variance adjustment is thus $\boldsymbol{\Gamma_y} \mapsto h^{-1}\boldsymbol{\Gamma_y}$. Thus the EKI update formula (22) becomes

$$\boldsymbol{\theta}_{n+1}^{(j)} = \boldsymbol{\theta}_{n}^{(j)} + \boldsymbol{C}_n^{\boldsymbol{\theta g}}(\boldsymbol{C}_n^{\boldsymbol{gg}} + h_n^{-1}\boldsymbol{\Gamma_y})^{-1}(\boldsymbol{y} + \boldsymbol{\xi}_n^{(i)} - \mathcal{G}(\boldsymbol{\theta}_n^{(j)})) \tag{27}$$

where each $\boldsymbol{\xi}_n^{(j)}$ is an independent realization of $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma_y})$ for time step n and particle j. This can be re-arranged to be interpreted as Euler Mayurama discretized coupled It$\hat{o}$ SDEs [13], [14],

$$\dot{\boldsymbol{\theta}}^{(j)} = \boldsymbol{C}^{\boldsymbol{\theta g}}\boldsymbol{\Gamma_y}^{-1}(\boldsymbol{y} - \mathcal{G}(\boldsymbol{\theta}^{(j)})) + \boldsymbol{C}^{\boldsymbol{\theta g}}\boldsymbol{\Gamma_y}^{-\frac{1}{2}}\dot{W}^{(j)} \tag{28}$$

obtained by taking the limit $h \to 0$. This is equivalently obtained by taking the $h \to 0$ limit of (23),

$$\dot{\boldsymbol{\theta}}^{(j)} = -\frac{1}{J}\sum_{k=1}^{J}\langle \mathcal{G}(\boldsymbol{\theta}_n^{(k)}) - \bar{\mathbf{g}}_n,\, \boldsymbol{\Gamma_y}^{-1}(\boldsymbol{y} - \mathcal{G}(\boldsymbol{\theta}_n^{(j)}) + \sqrt{\boldsymbol{\Gamma_y}}\dot{W}^{(j)})\rangle\, \boldsymbol{\theta}_n^{(k)} \tag{29}$$

For inversion applications omission of the stochastic term admits the simple Euler discretization

$$\boldsymbol{\theta}_{n+1}^{(j)} = \boldsymbol{\theta}_{n}^{(j)} - \frac{\Delta t_n}{J}\sum_{k=1}^{J}\langle \mathcal{G}(\boldsymbol{\theta}_n^{(k)}) - \bar{\mathbf{g}}_n,\, \boldsymbol{\Gamma_y}^{-1}(\mathcal{G}(\boldsymbol{\theta}_n^{(j)}) - \boldsymbol{y})\rangle\, \boldsymbol{\theta}_n^{(k)} \tag{30}$$

with time step Δt_n. Note that (30) is not empirically equivalent to (27) as the information of $\boldsymbol{C}_n^{\boldsymbol{gg}}$ was lost when taking h to 0 as $(\boldsymbol{C}_n^{\boldsymbol{gg}} + h_n^{-1}\boldsymbol{\Gamma_y})^{-1} \mapsto \boldsymbol{\Gamma_y}^{-1}$. The first step in the EKI algorithm is to construct an initial ensemble by randomly sampling from the parameter priors J times. For sake of understanding we can imagine that for each sample that is taken, a “particle" is placed in the parameter space at the coordinates corresponding to that sample, forming a system $\{\boldsymbol{\theta}_0^{(j)}\}_{j=1}^J$ of J interacting particles. Then, N iterations of updates are performed. At each iteration the j^{th} ensemble member is updated according to equation (27) or equation (30). Equation (30) in particular makes sense intuitively if we interpret the update on particle j as a sum of contributions from the remaining particles to an approximation of the gradient of the loss with respect to $\boldsymbol{\theta}^{(j)}$, which is then negated so that $\boldsymbol{\theta}^{(j)}$ moves in the (approximate) direction of decreasing loss. The k^{th} particle's relative contribution to this approximate gradient is scaled by an inner product capturing, firstly, how far the k^{th} parameter's corresponding prediction $\mathcal{G}(\boldsymbol{\theta}^{(k)})$ falls from the average prediction $\overline{\boldsymbol{g}}$. If this first quantity is large, $\boldsymbol{\theta}^{(j)}$ will be persuaded to leap away from $\boldsymbol{\theta}^{(k)}$. The second term in the inner product is just the error on the j^{th} parameter's corresponding prediction, down-scaled by the uncertainty on the observation. If the error is very small or the observation is very uncertain, $\boldsymbol{\theta}^{(j)}$ will be relatively uninfluenced by the other particles and will shift only slightly in parameter space. The overall update dynamic can be thought of as inducing parameters to perform an approximate form of gradient descent where gradients are approximated by ensemble differences.

0.4.2 Optimization: Overcoming Convergence Obstacles

In this section, I will first describe several optimizations to the previously described “vanilla" EKI algorithm that were originally derived for the EnKF to address the poor statistics of small ensembles. I will then describe several more recent optimizations to EKI that have been inspired by gradient descent approaches for neural networks. Lastly I will discuss optimizations employed to address challenges arising from the occasional numerical instability of our prototype model.

MAP Estimators and Tikhonov Regularization

Assuming a linear forward model $\mathcal{G}$, EKI provably optimizes the least-squares loss function

$$L(\boldsymbol{\theta}) = \frac{1}{2}\left\|{\boldsymbol{\Gamma}_{\boldsymbol{y}}}^{-\frac{1}{2}}(\boldsymbol{y} - \mathcal{G}(\boldsymbol{\theta}))\right\|^2 \tag{31}$$

over the subspace defined by the initial ensemble, in the continuous time limit [13]. However, we typically wish for EKI to optimize the regularized loss function (10) whose minimizer, as mentioned previously, corresponds to the MAP estimator $\text{argmax}_{\boldsymbol{\theta}}\pi(\boldsymbol{\theta}|\boldsymbol{y})$ of the conditional posterior on $\boldsymbol{\theta}$. This can achieved using a clever trick laid out in [27] based on state augmentation. In a modified formulation of the inverse problem (1), we define augmented variables $\boldsymbol{z}$, $\mathcal{F}$ and $\boldsymbol{\eta}$ as

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{y} \\ \mathbf{0} \end{bmatrix}, \quad \mathcal{F}(\boldsymbol{\theta}) = \begin{bmatrix} \mathcal{G}(\boldsymbol{\theta}) \\ \boldsymbol{\theta} \end{bmatrix}, \quad \hat{\boldsymbol{\eta}} = \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\eta_\theta} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\mu_\theta} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Gamma_y} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Gamma_\theta} \end{bmatrix} \right) \tag{32}$$

which tracks two systems $\boldsymbol{y} = \mathcal{G}(\boldsymbol{\theta}) + \boldsymbol{\eta}$ and $\mathbf{0} = \boldsymbol{\theta} + \boldsymbol{\eta_\theta}$. Replacing the terms in (31) with the corresponding augmented variables and taking into account the non-zero mean of $\hat{\boldsymbol{\eta}}$, the reformulated inverse problem $\boldsymbol{z} = \mathcal{F}(\boldsymbol{\theta}) + \hat{\boldsymbol{\eta}}$ optimizes the desired objective. $\Phi_R(\boldsymbol{\theta})$,

$$\begin{aligned} \Phi_R(\boldsymbol{\theta}) &= \frac{1}{2} \left\| \begin{bmatrix} \boldsymbol{\Gamma_y} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Gamma_\theta} \end{bmatrix}^{-\frac{1}{2}} (\boldsymbol{z} - \mathcal{F}(\boldsymbol{\theta}) - \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\mu_\theta} \end{bmatrix}) \right\|^2 \qquad (33) \\ &= \frac{1}{2} \begin{bmatrix} (\boldsymbol{y} - \mathcal{G}(\boldsymbol{\theta}))^T \boldsymbol{\Gamma}_{\boldsymbol{y}}^{-1} & (\boldsymbol{\mu_\theta} - \boldsymbol{\theta})^T \boldsymbol{\Gamma}_{\boldsymbol{\theta}}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{y} - \mathcal{G}(\boldsymbol{\theta}) \\ \boldsymbol{\mu_\theta} - \boldsymbol{\theta} \end{bmatrix} \qquad (34) \\ &= \frac{1}{2} ||\boldsymbol{y} - \mathcal{G}(\boldsymbol{\theta})||^2_{\boldsymbol{\Gamma_y}} + \frac{1}{2} ||\boldsymbol{\theta} - \boldsymbol{\mu_\theta}||^2_{\boldsymbol{\Gamma_\theta}} \qquad (35) \end{aligned}$$

Overcoming Sampling Error and the Subspace Property

The move from analytic distributions to ensemble-approximated distributions (14) for the EnKF can be thought of as a dimensionality reduction [20]. A drawback is that if the number of particles does not comfortably exceed the dimensionality of the state (e.g. if the state represents a large vector field collapsed into one dimension), then the PDF that the ensemble is supposed to approximate will be critically under-sampled. There comes the risk that the approximate covariance matrix (15) will be rank deficient, forging correlations between particles that don't actually exist and causing a compounding sampling bias. This is problematic in that it might lead to the true state not actually existing in the linear span of the ensemble, which is the only space eventually explored by the algorithm, as proven for

EKI in [13], thus removing all possibility of the algorithm meeting its objective of capturing the true underlying state $\boldsymbol{x}$ of the system. To ameliorate this issue, transformations to the empirical covariance can be introduced to try to remove spurious correlations in a technique known as covariance localization, where problem-specific assumptions about the covariance structure (e.g. short-range correlations) can be imposed to reduce the bias in the estimated covariance matrices. Covariance localization is explored for EKI in [28].

In this project, the rank deficiency problem is not of concern because we deal with small numbers of parameters. However, the sampling error and spurious correlations remain issues. A second technique known as covariance inflation is also used to address the poor statistics of small ensembles [13], [29]. Inspired by the observation that under-sampling leads to systematic underestimates of the posterior ensemble covariance, the simplest version of covariance inflation applies multiplicative or additive adjustments to the ensemble covariance following each analysis iteration by pushing each particle away from the ensemble mean. This prevents filter divergence, the phenomenon that occurs when the ensemble variance collapses so quickly that the filter effectively ignores observations, leading to under-fitting [30]. This is why covariance inflation is said to increase algorithm stability [19]. Schillings et al. [13] provide preliminary empirical evidence that covariance inflation and localization lead to more accurate state estimates given noise-ridden, nonlinear dynamics. However, as discussed in [30] and [31], incorporating these corrections brings new degrees of freedom to the filtering setup in the form of hyperparameters that can be expensive to tune. Lunderman et al. note that the hyperparameters for these two techniques need to be tuned jointly and suggest an adaptive localization [31]. In the work of [19] it is suggested that adaptive time stepping can be used to the same effect as covariance inflation.

Optimizations Inspired by Gradient Descent for Convergence Control

EKI and the EnKF are biased in that they are inconsistent for nonlinear forward maps, meaning the ensemble does not converge in expectation to the true posterior at pseudo-time one [32]. Moreover, the deleterious consequences of under-sampling raise doubts about EKI's applicability to problems where huge ensembles are infeasible. However, there is empirical evidence that EKI can perform well both in high-dimensional parameter spaces and for nonlinear forward maps. In a 2018 paper by Kovachki et al., an ensemble of 2000 particles was successfully used to pursue optimal values for more than 500,000 parameters in a convolutional neural network. The neural network was tuned to classify images of handwritten digits from the MNIST dataset, a popular and respected toy dataset used in computer vision. A relatively ad hoc form of randomization was used to try to overcome the subspace property. In particular, the particles were randomly perturbed about the ensemble mean at each iteration. With this optimization, after 50 epochs of training were performed using EKI on mini-batches of 600 images, the network correctly classified 97% of 10,000 images that it hadn't seen before, barely falling short of the 98% correctly classified by the same network trained using the traditional method of stochastic gradient descent with a learning rate of 0.1.

Curiously, the EKI dynamic follows a gradient flow in the linear setting. In [13] it is shown that by locally linearizing the forward map, the continuous-time approximation of EKI (29) degenerates to a preconditioned gradient descent,

$$\dot{\boldsymbol{\theta}}^{(j)} = -\boldsymbol{C}^{\boldsymbol{\theta}\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \Phi(\boldsymbol{\theta}^{(j)}) \tag{36}$$

where the preconditioning helpfully buffers the sensitivity of the descent to correlations and scale discrepancies in the parameter space. With particular attention to this result, in the works of [15] and [19] it is emphasized that techniques developed in the gradient descent literature might carry over to nonlinear optimization with EKI. For example, mini-batching can potentially be used to overcome memory constraints when the observations are large or many, e.g. in the case of [15]. Momentum-augmented approaches also explored in [15] may be useful for setting a search direction that is informed by the aggregated history of search directions. If the parameter space is high-dimensional, overfitting can occur where the predictive burden falls on a small subset of parameters or array features. Dropout, the strategy of randomly zeroing individual parameters or intermediate computed values during each training iteration, can help discourage such underutilization of parameters.

Arguably the most crucial insights to be borrowed from gradient descent relate to adaptive step size schemes. A generally effective strategy is to increase the time step as the iterations progress to compensate for the gradual shallowing out of the loss landscape, both to prevent convergence deceleration and to escape suboptimal minima at possible expense of stability. The approach of [15] is to rewrite the EKI discrete update (30) in terms of a matrix $\boldsymbol{D}$ as

$$\begin{aligned}\boldsymbol{X}_{n+1} = \boldsymbol{X}_n - \frac{\Delta t_n}{J}\boldsymbol{X}_n\boldsymbol{D}_n& \\ \text{where } \boldsymbol{X}_n = \begin{bmatrix}\boldsymbol{\theta}_n^{(1)} & \dots & \boldsymbol{\theta}_n^{(J)}\end{bmatrix}& \\ \boldsymbol{D}_{n,ij} = \langle \mathcal{G}(\boldsymbol{\theta}_n^{(j)}) - \overline{\mathbf{g}},\, {\boldsymbol{\Gamma}_{\boldsymbol{y}}}^{-1}(\mathcal{G}(\boldsymbol{\theta}_n^{(i)}) - \boldsymbol{y})\rangle&\end{aligned}$$

and let the adaptive time step for a given iteration be computed as

$$\Delta t_n = \frac{\Delta t_0}{\|\boldsymbol{D}_n\|_F + \epsilon}, \quad \|\boldsymbol{D}_n\|_F = \sqrt{\sum_i \sum_j |D_{n,ij}|^2}$$

where ϵ is a small value to prevent division by zero and Δt_0 is a reference time step which the authors suggest should be made as large as possible up before stability is compromised. This approach was used by Garbuno-Inigo et al. [33] and Chada et al. [27] and referenced in the work of [34]. An alternative idea is to use the determinant of the ensemble covariance matrix at each iteration as a proxy for the volume of the ensemble and to use fixed point iteration to set a step size that achieves a prescribed rate of collapse of this volume measure. A third approach, proposed by Iglesias et al. [34], is inspired by the adaptive-tempering Sequential Monte Carlo interpretation of EKI and attempts to control a “level of information" between successive intermediate densities as gauged by a metric called Jeffreys’ divergence. Without going into detail about how the various methods were implemented and how the hyperparameters for each were tuned, Figure 0-2 demonstrates that the choice of adaptive step size algorithm is pivotal for achieving a desired rate of convergence of the EKI ensemble mean estimate. Ultimately, EKI is an iterative solver for Bayesian inverse problems that first attempts to reconstruct a probability distribution of $\boldsymbol{\theta}$ that combines prior knowledge with the information from the collected data $\boldsymbol{y}$, then pursues the optimizer of this distribution. The rate of ensemble convergence affects the accuracy of this search.

Optimizations Specific to the Modeling Application at Hand

Uncalibrated dynamical models are prone to numerical instability. Occasionally an EKI particle will cause the model solution to explode, causing some entries to outgrow a 64 bit floating point representation, rendering the EKI update incalculable. We call such particles “failed"; all particles that successfully lead to finite numerical solutions are “successful". Sometimes particle failures can be prevented simply by decreasing the model integration time or increasing the time step size used for the integration. A strategy of gradually increasing model integration time or decreasing time step size as the EKI ensemble collapses might be fruitful. However, approaches for dealing with particle failures once they have occurred are still warranted. Following each given ensemble update, all failed particles can simply be removed from the ensemble and EKI can proceed with the reduced ensemble size. Alternatively, the failed particles can be re-sampled from a multivariate normal distribution parameterized by the ensemble mean and covariance computed either (i) among all particles, or (ii) among only the successful particles, under the condition that all re-sampled particles are successful. A third idea is to ignore the failed particles and update only the successful particles, then recover the original ensemble size by re-sampling from the mean and covariance of the updated ensemble. We chose to pursue this latter approach, where the algorithm terminates if the fraction of all particles that fail on a given iteration exceeds a specified threshold; we chose 20%. Tolerating a reasonable fraction of particle failures affords the flexibility to use wide parameter priors for thorough exploration of parameter space, at possible risk of incurring bias and non-Gaussianity to the failure-adjusted ensembles.

Particle failures occur as particles wander into unlikely regions of parameter space. Unfortunately, EKI requires Gaussian priors which are inherently unconstrained. One way to remedy this situation is to artificially impose bounds on the parameter space by employing constrained priors whose constituent samples can be one-to-one mapped to samples of unconstrained, Gaussian distributions. The “unconstrained" EKI particles, then, are just transformed versions of the true particles which can be confined to arbitrary intervals while obeying the normality requirement for EKI. We use scaled logit-normal priors that are confined to a specified interval $[a, b]$. Each “constrained" scalar sample θ_i can be mapped to a standard normal, “unconstrained" counterpart $\hat{\theta}_i$ by way of an analytic transformation,

$$\hat{\theta}_i = \log(\frac{\frac{\theta_i - a}{b-a}}{1 - \frac{\theta_i - a}{b-a}}) = \log\frac{\theta_i - a}{b - \theta_i}$$

and conversely, the samples $\hat{\theta}_i$ dealt with by EKI can be returned to the constrained space by the following inverse transformation.

$$\theta_i = \frac{\exp(\hat{\theta}_i)b + a}{1 + \exp(\hat{\theta}_i)}$$

Importantly the use of constrained logit-normal distributions has implications for how the uncertainty quantification is eventually carried out.

0.5 Uncertainty Quantification

Understanding how parameter uncertainty propagates from input to output allows model developers to concentrate their efforts on making structural improvements where improvements are the most needed. Using the Calibrate Emulate Sample [2] framework as a starting point, approximate posteriors can be generated for each parameter to gauge parametric uncertainty. Parameters that exhibit high levels of uncertainty can be considered for removal. Then, using variance-based sensitivity analysis, individual parameters can be characterized in terms of how much of the uncertainty in the model solution they account for. The greatest offenders can be prioritized as targets for future development. In turn, those parameters that exhibit high uncertainty and low sensitivity can be retired from the model.

0.5.1 Emulation via Gaussian Process Regression

To enable efficient sampling of parameter posteriors, we train a Gaussian Process emulator to stand in for the true forward model. In the original CES paper [2], a multi-output emulation approach is used where, for an output in $\mathbb{R}^{d_y}$, d_y independent Gaussian processes are trained to emulate each coordinate in the output. A key challenge is that the various components of the model output are inevitably correlated, meaning that if the GPs are to produce accurate uncertainty estimates needed for sampling, the outputs must be decorrelated prior to emulation. The authors suggest that the decorrelation can be achieved based on the observation noise covariance matrix or by projecting the output data into an output space with linearly uncorrelated dimensions, e.g. an orthonormal basis computed by Principal Component Analysis (PCA) via singular vector decomposition (SVD). We adopt the SVD approach and additionally perform a dimensionality reduction by preserving only the first 20 principal components in order to save on GP training and evaluation time in an otherwise 2176-dimensional output space. Given N training samples $\{\boldsymbol{\theta}^{(n)}\}_{n=1}^{N}$ and corresponding forward map outputs $\{\mathcal{G}(\boldsymbol{\theta}^{(n)})\}_{n=1}^{N}$, PCA is executed on the column space of a matrix $\boldsymbol{G}$ of zero-centered outputs as follows [2]

$$\boldsymbol{G} = \hat{\boldsymbol{G}}\,\boldsymbol{\Sigma}\,\boldsymbol{V}^{\boldsymbol{T}} \tag{37}$$

$$\begin{bmatrix} (\mathcal{G}(\boldsymbol{\theta}^{(1)}) - \mathbf{m})^T \\ \vdots \\ (\mathcal{G}(\boldsymbol{\theta}^{(N)}) - \mathbf{m})^T \end{bmatrix} = \begin{bmatrix} (\hat{\boldsymbol{g}}^{(1)})^T \\ \vdots \\ (\hat{\boldsymbol{g}}^{(N)})^T \end{bmatrix} \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_{d_y} \end{bmatrix} \begin{bmatrix} (\boldsymbol{v}^{(1)})^T \\ \vdots \\ (\boldsymbol{v}^{(d_y)})^T \end{bmatrix} \tag{38}$$

where d_y is the length of the forward map output and $\mathbf{m} = \sum_{n=1}^{N} \mathcal{G}(\boldsymbol{\theta}^{(n)})$. Then the rows $(\hat{\boldsymbol{g}}^{(n)})^T$ of $\hat{\boldsymbol{G}}$ are the rows of $\boldsymbol{G}$ unscaled and projected onto the output space defined by the right-singular vectors $\boldsymbol{v}^{(1)}, \dots, \boldsymbol{v}^{(d_y)}$. To dimensionality-reduce the transformed outputs we eliminate all but the first $r < d_y$ principal components by preserving only dimensions corresponding to the r largest singular values, yielding a new decomposition in terms of truncated matrices distinguished with the subscript r,

$$\hat{\boldsymbol{G}}_r\,\boldsymbol{\Sigma}_r\,\boldsymbol{V}_r^{\boldsymbol{T}} = \begin{bmatrix} \hat{\boldsymbol{g}}_1^{(1)} & \dots & \hat{\boldsymbol{g}}_r^{(1)} \\ \vdots & \ddots & \vdots \\ \hat{\boldsymbol{g}}_1^{(N)} & \dots & \hat{\boldsymbol{g}}_r^{(N)} \end{bmatrix} \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_r \end{bmatrix} \begin{bmatrix} (\boldsymbol{v}^{(1)})^T \\ \vdots \\ (\boldsymbol{v}^{(r)})^T \end{bmatrix} \tag{39}$$

Because we have transformed the forward map outputs we correspondingly transform the observation $\boldsymbol{y}$ and associated uncertainty as

$$\hat{\boldsymbol{y}} = \boldsymbol{\Sigma}_r^{-1}\, \boldsymbol{V}_r^T\, (\boldsymbol{y} - \mathbf{m})$$
$$\hat{\boldsymbol{\Gamma}}_{\boldsymbol{y}} = \boldsymbol{\Sigma}_r^{-1}\, \boldsymbol{V}_r^T\, \boldsymbol{\Gamma}_{\boldsymbol{y}}\, \boldsymbol{V}_r\, (\boldsymbol{\Sigma}_r^{-1})^T$$

where the latter transformation assumes that $\boldsymbol{y} \approx \mathbf{m}$. To construct an emulator that maps from $\mathbb{R}^{17}$ to $\mathbb{R}^{20}$, we utilize 20 Gaussian Processes, one for each grid point in the transformed outputs and each one mapping $\mathbb{R}^{17}$ to $\mathbb{R}$ along with estimated uncertainties. Each Gaussian process computes a Gaussian probability distribution over functions that interpolate the training data, where the form of the interpolation functions is dictated by the choice of kernel function. If the training observations are noise-ridden the associated uncertainty can be incorporated into the GP typically leading to imperfect interpolation of the training data by the GP; here, we assume perfect data. Certain performance considerations are worth noting. Given a set $\boldsymbol{\Theta} = \{\boldsymbol{\theta}^{(n)}\}_{n=1}^{n=N}$ of training inputs and vector $\mathbf{f}_i = [\hat{\boldsymbol{g}}_i^{(1)}, \ldots, \hat{\boldsymbol{g}}_i^{(N)}]$ of corresponding training targets representing the i^{th} component of the transformed forward map outputs, the GP predictive mean and variance for a test input $\boldsymbol{\theta}_*$ are given by

$$m_i(\boldsymbol{\theta}_*) = \mathbb{E}[\hat{f_{i*}}|\boldsymbol{\Theta}, \mathbf{f}_i, \boldsymbol{\theta}_*] = \mathbf{k}_*(\mathbf{L}^T\backslash(\mathbf{L}\backslash\mathbf{f}_i)) \tag{40}$$

$$\mathrm{v}_i(\boldsymbol{\theta}_*) = \mathbb{V}[\hat{f_{i*}}|\boldsymbol{\Theta}, \mathbf{f}_i, \boldsymbol{\theta}_*] = k(\boldsymbol{\theta}_*, \boldsymbol{\theta}_*) - \mathbf{v}^T\mathbf{v} \tag{41}$$

where $\mathbf{k}_{*n} = k(\boldsymbol{\theta}_*, \boldsymbol{\theta}^{(n)})$ is an N-length vector of evaluations of the covariance function k on the test point and the N training inputs, and $\mathbf{L} = \text{Cholesky}(\mathbf{K})$ where $\mathbf{K}_{ij} = k(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(j)})$ is the Cholesky factorization of the $N \times N$ matrix of evaluations of the covariance function on the training inputs and themselves, and $\mathbf{v}$ is $\mathbf{L}\backslash\mathbf{k}_*$ [35]. The distribution over possible values of f_i* as described by the GP is $\mathcal{N}(m_i(\boldsymbol{\theta}_*), \mathrm{v}_i(\boldsymbol{\theta}_*))$. Taking all components together, we have that the overall predictive distribution over possible values of the transformed forward map output vector $\mathbf{f} = [\hat{\boldsymbol{g}}^{(1)}, \ldots, \hat{\boldsymbol{g}}^{(N)}]$ for a given parameter set $\boldsymbol{\theta}$ is

$$\hat{\mathcal{G}}(\boldsymbol{\theta}) \sim \mathcal{N}(\hat{m}(\boldsymbol{\theta}), \hat{\Gamma}_{GP}(\boldsymbol{\theta})) \tag{42}$$

where $\hat{m}(\boldsymbol{\theta}) = [m_1(\boldsymbol{\theta}), \ldots, m_r(\boldsymbol{\theta})]$ and $\hat{\Gamma}_{GP}(\boldsymbol{\theta}) = \text{diag}([v_1(\boldsymbol{\theta}), \ldots, v_r(\boldsymbol{\theta})]$. Inspecting Equation (40) reveals that predicting a transformed forward map output for a single test input $\boldsymbol{\theta}_*$ involves evaluating the kernel function $20N$ times, which can be computationally taxing if N is large. Additionally, carrying out the necessary matrix inversions can be both speed- and memory-intensive again for large N. Thus GPR does not scale well to large training datasets. In advanced applications, this would likely necessitate strategic subsampling of the available observations to procure a representative training set. Otherwise, an alternative universal function approximator such as a neural network might make for a more effective emulator.

For each constituent GP in the emulator, we non-dimensionalize the inputs and outputs using the mean and standard deviations of the training inputs and outputs. We impose a zero mean function and an ARD squared exponential kernel to model the decay of output correlations with respect to parameter distance,

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma^2 \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}')^T \mathrm{diag}(\boldsymbol{\ell})^{-2}(\boldsymbol{\theta} - \boldsymbol{\theta}')\right)$$

where the noise hyperparameter σ and all length-scale hyperparameters in $\boldsymbol{\ell} \in \mathbb{R}^{17}$ are initialized to 1 prior to optimization. Intuitively, the length-scale hyperparameters $\boldsymbol{\ell}$ modulate how the distances between two inputs in each dimension scale the covariance between the associated predictive outputs; the noise hyperparameter σ modulates the overall scale of these covariances. These kernel hyperparameters are optimized jointly on a log scale using BFGS gradient-based optimization, where the loss function is obtained by negating the log likelihood, $\log \pi(\mathbf{y}|\boldsymbol{\Theta}) = -\frac{1}{2}\mathbf{f}_i(\mathbf{L}^T \backslash (\mathbf{L} \backslash \mathbf{f}_i)) - \sum_n \mathbf{L}_{nn} - \frac{N}{2}\log(2\pi)$, of the training targets given the training inputs.

0.5.2 Sampling

Following emulation, sampling is performed using the random walk Metropolis-Hastings algorithm in parallel chains initialized at the locations of the particles in the EKI initial ensemble. At each algorithm iteration for each given chain, the sample is randomly perturbed by an independent draw from a Gaussian distribution with mean zero and covariance computed as the collective covariance of all EKI samples, yielding a perturbed ‘proposal‘ sample. Then, with some probability the proposal sample is ‘accepted‘ and added to the chain; otherwise the original unperturbed sample is appended to the chain and carried over to the next iteration. Conceptually speaking, the samples should spend time at each location in proportion to the likelihood that the true parameters exist at that point. Accordingly, for a current sample $\boldsymbol{\theta}$ and proposal $\boldsymbol{\theta}'$, the acceptance probability $a(\boldsymbol{\theta}, \boldsymbol{\theta}')$ is a function of the ratio of the posterior probabilities $\hat{\pi}(\cdot|y)$ associated with each sample,

$$a(\boldsymbol{\theta}, \boldsymbol{\theta}') = \min\{1,\ \frac{\hat{\pi}(\boldsymbol{\theta}'|y)}{\hat{\pi}(\boldsymbol{\theta}|y)}\} = \min\{1,\ \exp(\hat{\Phi}(\boldsymbol{\theta}) - \hat{\Phi}(\boldsymbol{\theta}'))\}$$

If the sampler relies on evaluations of the true forward map, $\mathcal{G}$, then $\hat{\Phi}(\boldsymbol{\theta})$ is readily given by the regularized objective (10) as derived in section 0.4.1. As for the emulated forward map, $\hat{\Phi}(\boldsymbol{\theta})$ needs to take into account the predictive uncertainty (41) of the emulator. In particular, when $\hat{\mathcal{G}}$ replaces $\mathcal{G}$ as the forward model, we can reinterpret the inverse problem as

$$\hat{\boldsymbol{y}} = \mathcal{G}(\boldsymbol{\theta}^*) + \hat{\boldsymbol{\eta}} + \hat{\boldsymbol{\eta}}_{GP}(\boldsymbol{\theta})$$

where $\hat{\boldsymbol{\eta}} \sim \mathcal{N}(\mathbf{0}, \hat{\mathbf{\Gamma}}_{\boldsymbol{y}})$ and $\hat{\boldsymbol{\eta}}_{GP}(\boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{0}, \hat{\Gamma}_{GP}(\boldsymbol{\theta}))$. Taking into account the previously nonexistent $\boldsymbol{\theta}$-dependence of the noise, we write equation (8) in terms of the transformed variables from the previous section,

$$\pi(\hat{\boldsymbol{y}}|\boldsymbol{\theta}) = \frac{1}{(2\pi)^{\frac{r}{2}}|\hat{\Gamma}_{GP}(\boldsymbol{\theta}) + \hat{\mathbf{\Gamma}}_{\boldsymbol{y}}|^{\frac{1}{2}}} e^{-\frac{1}{2}\left\|(\hat{\Gamma}_{GP}(\boldsymbol{\theta})+\hat{\mathbf{\Gamma}}_{\boldsymbol{y}})^{-\frac{1}{2}}(\hat{\boldsymbol{y}}-\hat{\mathcal{G}}(\boldsymbol{\theta}))\right\|^2}$$

leading to a revised regularized objective function, [2]

$$\hat{\Phi}_{GP}(\boldsymbol{\theta}) = \frac{1}{2}\left\|(\hat{\Gamma}_{GP}(\boldsymbol{\theta}) + \hat{\mathbf{\Gamma}}_{\boldsymbol{y}})^{-\frac{1}{2}}(\hat{\boldsymbol{y}} - \hat{m}(\boldsymbol{\theta}))\right\|^2 + \frac{1}{2}\left\|\mathbf{\Gamma}_{\boldsymbol{\theta}}{}^{-\frac{1}{2}}(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})\right\|^2 + \frac{1}{2}\log\det(\hat{\Gamma}_{GP}(\boldsymbol{\theta}) + \hat{\mathbf{\Gamma}}_{\boldsymbol{y}}) \tag{43}$$

0.5.3 Sensitivity Analysis

Sensitivity estimates reveal the extent to which a given parameter's associated uncertainty is problematic for the model. Highly uncertain parameters can be inconsequential for a model if they exhibit low sensitivity. It is therefore helpful to supplement parametric uncertainty information with sensitivity information. Local sensitivity analysis approaches measure the effects of local parameter perturbations on the model output. We pursue a global approach called Sobol or “variance-based" sensitivity analysis. Ilya Sobol found that the variance of a model’s output can be decomposed into a sum of contributions from the various individual inputs and combinations thereof: [36]

$$V(\hat{\Phi}) = \sum_i V_i + \sum_i \sum_{j>i} + \cdots + V_{12\ldots d_\theta}$$

where $\hat{\Phi}$ is the model, $V(\hat{\Phi})$ is the model variance, and the various indices are parameter indices ranging from 1 to the number of parameters, d_θ. For a parameter $\boldsymbol{\theta}_i$ the first-order variance $V_i = \mathbb{V}_{\boldsymbol{\theta}_i}(\mathbb{E}_{\boldsymbol{\theta}_{\sim i}}(\hat{\Phi}|\boldsymbol{\theta}_i))$ measures the contribution to the output variance of $\boldsymbol{\theta}_i$ alone, averaging over the other inputs $\boldsymbol{\theta}_{\sim i}$; the second-order variances $V_{ij} = \mathbb{V}_{\boldsymbol{\theta}_{ij}}(\mathbb{E}_{\boldsymbol{\theta}_{\sim ij}}(\hat{\Phi}|\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)) - V_i - V_j$ measure the contribution to the output variance of only the interactions between $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$; and so on. The total order sensitivity index for $\boldsymbol{\theta}_i$ takes the contribution of $\boldsymbol{\theta}_i$ to the output variance together with those of all interactions involving $\boldsymbol{\theta}_i$ for an overall interaction-inclusive sensitivity measure on $\boldsymbol{\theta}_i$. The variance decomposition comes from viewing the model as a sum of constituent models each with a unique combination of the inputs,

$$\hat{\Phi}(\boldsymbol{\theta}) = f_0 + \sum_i f_i(\boldsymbol{\theta}_i) + \sum_i \sum_{j>i} f_{ij}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) + \cdots + f_{12\ldots d_\theta}(\boldsymbol{\theta})$$

where the parameters $\boldsymbol{\theta}$ inhabit a parameter space that has been shrunk or stretched along each dimension to fit a unit hypercube, i.e. $\boldsymbol{\theta}_i \in [0,1] \; \forall \; i$. By applying Sobol analysis to the GP-emulated objective $\hat{\Phi}_{GP}$ we can gauge how the variance in the loss is apportioned among the various parameters. To enable unit hypercube sampling, we impose bounds on the parameter space spanning the range of ±2 standard deviations from the means approximated through MCMC sampling of the $\hat{\Phi}_{GP}$. We then take approximate i.i.d. samples using Quasi-Monte Carlo sampling and finally apply the Jansen’s method [37] to estimate the first- and total-order sensitivity indices.

Part III

Results

Tikhonov-regularized EKI (0.4.2) was run for 10 iterations with 128 ensemble members in pursuit of a solution to the Bayesian inverse problem described in section 0.3. 1408 calls to the forward model $\mathcal{G}$ were made in this process and the outputs for the first 768 were saved for eventual use as training samples for emulation. Adaptive time steps were computed so as to prescribe a linearly decreasing rate of ensemble collapse, as measured by the ratio $|\boldsymbol{C}_{n+1}^{\boldsymbol{\theta\theta}}|/|\boldsymbol{C}_{n}^{\boldsymbol{\theta\theta}}|$ of the ensemble covariance matrix determinants at subsequent iterations, starting with 0.2 at iteration 0 to 0.01 at iteration 10. Figure 0-3 summarizes the parameter-wise evolution of the ensemble distribution. The ensemble variance for each parameter decreases near-monotonically as the iterations progress, indicating that the ensemble members collapsed toward a consensus in every dimension of the parameter space. Figure 0-4 shows the model-data misfit $\left\|\mathcal{G}(\overline{\boldsymbol{\theta}}_n) - \boldsymbol{y}\right\|^2$ associated with the ensemble mean at each iteration, both for the calibration suite responsible for the ensemble mean values and for 2- and 6-day suites which represent unfamiliar physical scenarios. The error plateaus around iteration 4 in all cases, indicating that essentially 640 calls to the forward model were needed to arrive at near-optimal parameters. Figures 0-6, 0-7 and 0-8 demonstrate the interpolation and extrapolation quality of the model, showing temperature and velocity profiles predicted by the calibrated model and LES for each suite of LES simulations.

Following calibration, 20 independent Gaussian processes were trained to emulate a dimensionality-reduced version of the forward map with linearly uncorrelated output dimensions. 98.4% of the variance in the original training samples was preserved in the process of dimensionality reducing the output space from $\mathbb{R}^{2176}$ to $\mathbb{R}^{20}$, as measured by the ratio of the traces of the full (2176×2176) and truncated (20×20) singular value matrices. 745 samples representing the ensemble members from iterations 0 to 5 of EKI were used as training data for the emulator; the remaining 23 samples were held out to assess the quality of the GP fit for each component of the output, as shown in Figure 0-9. For the most part, the GP mean predictions interpolate the validation targets well, and those samples which a GP predicts poorly are typically associated with high uncertainty as desired. The Pearson correlation coefficients between the true and predicted validation targets are either close to 1 or otherwise biased by a small number of uncertain outliers. Using Gaussian processes sped up the forward map evaluation 71-fold.

Following emulation, sampling was performed in 128 parallel chains using the random walk Metropolis-Hastings procedure described in section 0.5.2. The chain length for emulator sampling was 2000 with the first 200 assumed to represent transient samples and removed; the chain length for MCMC sampling was 1000 with a burn-in duration of 100. Figure 0-10 shows the superimposed marginal densities obtained by sampling the emulated objective $\hat{\Phi}_{GP}$ as compared to those obtained by sampling the true objective Φ_R. The considerable overlap between these densities, including of the sample means, suggests that EKI adequately sampled the true posteriors for each parameter and that the GP emulator successfully captured the likelihood information necessary for approximating parametric uncertainty to a reasonable degree of accuracy. Finally, in a departure from the purely Bayesian framework, Sobol sensitivity analysis was applied to 76,000 Quasi-Monte Carlo samples from a 17-dimensional unit hypercube scaled to span ±2 standard deviations from the means of the dimension-wise marginal distributions approximated by MCMC on the emulated likelihoods. The result as shown in Figure 0-12 shows considerable correlation between the normalized first- and total-order sensitivity indices suggesting that first-order effects take the most responsibility for the variability in the loss, and in particular the effects of parameters $C^K c^-$ and $C^K e^-$.

Part IV

Discussion

Given sufficient computational resources and an adequate Bayesian formulation of the model-data relationship, data-driven approaches for systematic parameter inference and uncertainty quantification can be used to automate the tuning of parameterized dynamical models and provide empirical feedback to inform future model development. We have shown that Ensemble Kalman Inversion can be used successfully to tune the 17 parameters of a sub-grid closure within an OSBL column model to bring the model into close agreement with in- and out-of-sample horizontally averaged solutions of high-fidelity idealized numerical simulations. We then used the parameter-output pairs generated by EKI to train a Gaussian process emulator as a computationally inexpensive surrogate for the true forward model, and showed that the marginal distributions and parameter correlations estimated through sampling the emulated likelihoods successfully approximated those estimated by sampling the true likelihoods.

In spite of the linearity and normality assumptions inherited by EKI from the Kalman filter, to say nothing of the finite sampling and discrete time approximations required for use of EKI in practice, we have added to the limited body of empirical evidence that EKI can succeed in nonlinear settings. With EKI we have obtained a calibration result that we believe provides an accurate qualitative sense of how CATKE can perform when operating at its full potential, subject to the given model resolution and other such fixed settings. Our procedure's demonstrated success in uncertainty quantification has important implications for EKI as well. First, it suggests that EKI is a suitable replacement for Ensemble Kalman Sampling in the Calibrate Emulate Sample framework. Ensemble Kalman Sampling has the disadvantage that the theoretical derivation of the algorithm prescribes a natural time step for the dynamic, and the number of iterations required for the dynamic to reach equilibrium at the posterior can be prohibitively large, precluding the use of expensive forward models. Second, we suggest that EKI is better than a last resort meant for situations where forward model derivatives are unobtainable; many gradient-based approaches seek only to optimize and cannot offer any semblance of a Bayesian description of the distribution over possible parameter values. The use of EKI enables subsequent likelihood emulation, removing the barrier to state-of-the-art MCMC methods for expensive forward models. In the same vein, Gaussian processes have a distinct advantage over many alternative emulation approaches in that they can provide information about emulator uncertainty which enables a more faithful MCMC sampling of the true parameter posteriors and as well as a more stable MCMC dynamic. In addition to being useful for facilitating model development, the ability to derive posterior estimates was in our work an end in and of itself, and the posterior estimates obtained via MCMC can be repurposed as priors when CATKE is eventually embedded into a global model.

There are many possible directions for future work pertaining to every aspect of the CES procedure. Training and evaluation time for the GP emulator can be prohibitive if many samples are available for training. Such restrictions may warrant further work on robust adaptive time stepping schemes that promise to cater a minimally redundant and high-discrepancy set of input-output pairs. Work on fine-tuning the time stepping procedures may also be fruitful for addressing other shortcomings of GP emulation. For example, if the ensemble converges too quickly along a given dimension, the GP will tend to underestimate the uncertainty for that parameter as the GP predictive uncertainty will be highly inflated outside the narrow domain of data for that parameter. Emulation may be improved by use of alternative kernel functions, new methods of decorrelating and compressing output coordinates, or alternative multi-output emulation approaches. Analytical methods for deriving first- and second-order Sobol sensitivity indices directly from tuned GP models [38] may eventually be useful if a single GP is able to be trained to interpolate the objective function directly. Crucially, in this work, access to gradient information of the GP emulator has only been used for kernel hyperparameter optimization, but may also be exploited to improve the sampling procedure, for example by replacing the random walk sampling scheme with a Hamiltonian Monte Carlo sampler.

Part V

Figures

Table 1. Large Eddy Simulation Constants

	Momentum flux $\overline{U'W'}_{z=0}\,/\mathrm{m^2\,s^{-2}}$			Buoyancy flux $\overline{W'T'}_{z=0}\,/\mathrm{m\,s^{-1}\,K}$			Coriolis parameter $f\,/\mathrm{s^{-1}}$		
Length (days) Scenario	2	4	6	2	4	6	2	4	6
Strong cooling	0	0	0	1.2E-7	7E-8	5E-8	-1E-4	-1E-4	-1E-4
Strong wind	-1E-3	-8E-4	-7E-4	0	0	0	-1E-4	-1E-4	-1E-4
Strong wind, weak cooling	-7E-4	-6.5E-4	-5.5E-4	6E-8	4E-8	3E-8	-1E-4	-1E-4	-1E-4
Weak wind, strong cooling	-3.3E-4	-3E-4	-2.2E-4	1.1E-7	7E-8	5E-8	-1E-4	-1E-4	-1E-4
Strong wind, no rotation	-2E-4	-1E-4	-7E-5	0	0	0	0	0	0

Table 1: Simulation-specific constants for the 2-day, 4-day and 6-day suites of high-resolution large eddy simulations used to provide the ground truth data for the profile evolutions of $\overline{\boldsymbol{u}}$, $\overline{\boldsymbol{v}}$, $\overline{\boldsymbol{c_\theta}}$, and $\boldsymbol{e}$ over a course of 2, 4 and 6 days, respectively. CATKE is calibrated against the 4-day simulations; the 2- and 6-day simulations are reserved for validation. The five simulations within each suite differ in the surface momentum flux, surface buoyancy flux (positive values indicate cooling and negative values heating), and rotation. The particular values were chosen so that the temperature mixed layer depth reaches about 50% of the vertical domain (100-120 meters) by the end of each simulation. The simulations begin at rest ($\overline{\boldsymbol{u}}|_{t=0} = \overline{\boldsymbol{v}}|_{t=0} = 0$) and with an identical temperature initial condition that has three layers, where in each layer the temperature increases linearly with proximity to the surface, and the stratification is the steepest at the extremes.

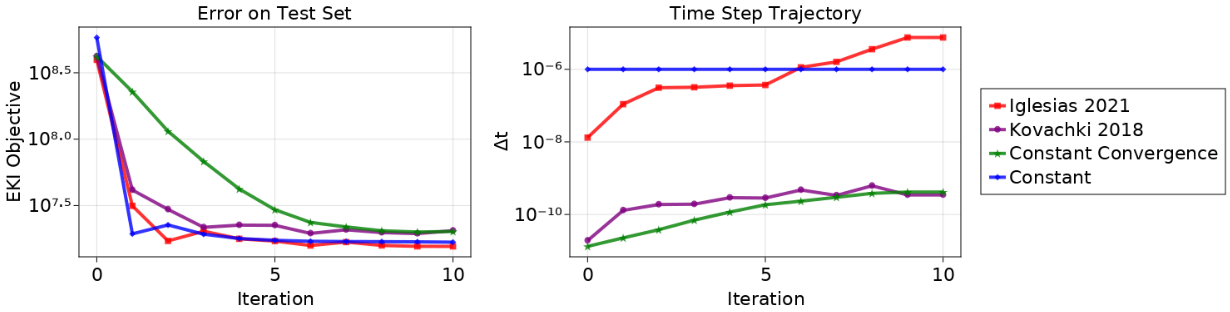

Figure 0-2: **(Left)** Evaluations of $\Phi_R(\overline{\boldsymbol{\theta}}_n)$ (10) as a function of n for the 6-day “testing" suite of observations, where $\overline{\boldsymbol{\theta}}_n$ is the ensemble mean obtained by running EKI on the 4-day “training" suite. Iglesias 2021 is the hyperparameter-free approach of [34]; Kovachki 2018 is the approach of [15]; Constant Convergence is the strategy of using fixed point iteration to achieve a prescribed ratio of determinants of the ensemble covariance at subsequent iterations; Constant is the strategy of using a fixed step size. Hyperparameters were chosen via grid search to minimize the error $\Phi_R(\overline{\boldsymbol{\theta}}_{10})$ of the final ensemble mean on the 2-day “validation" suite. **(Right)** EKI pseudo-time step as a function of iteration, where the time step is understood in the sense of equation (27). Because we are unconfident in the hyperparameter optimization for each of these methods we do not consider the result a performance comparison between adaptive step size schemes but rather empirical proof of the significant influence that the step size trajectory has on the properties of the ensemble convergence. Thus details about algorithm implementations are beyond the scope of this report and we refer interested readers to ParameterEstimocean.jl [17] where the various schemes are implemented.

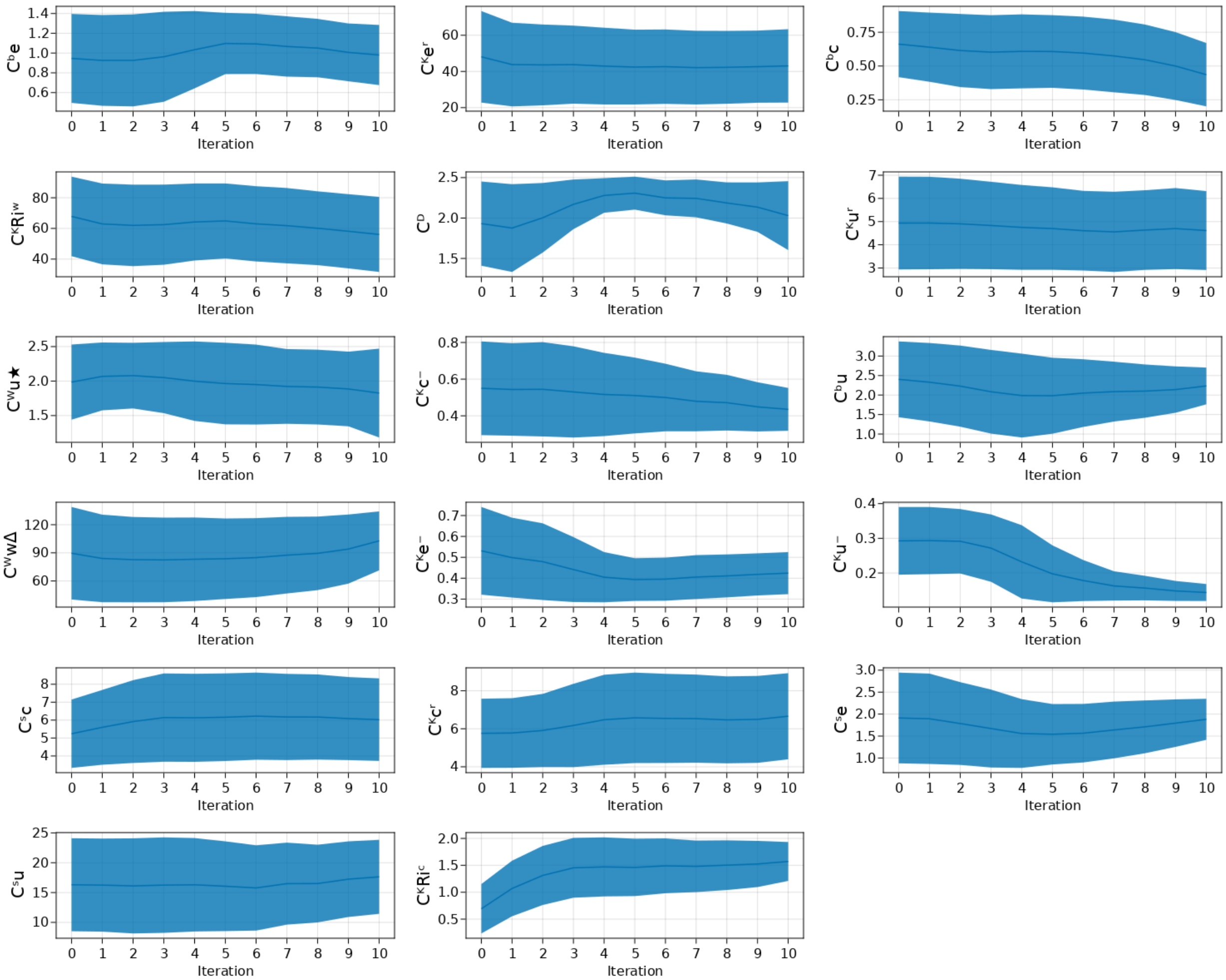

Figure 0-3: Marginal ensemble mean and variance of each parameter at each EKI iteration.

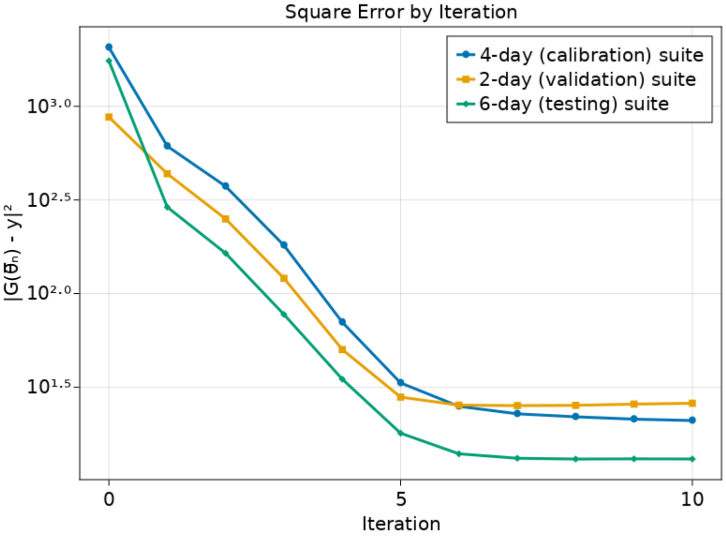

Figure 0-4: Model-data misfit $\left\|\mathcal{G}(\overline{\boldsymbol{\theta}}_n) - \boldsymbol{y}\right\|^2$ by iteration given in-sample and out-of-sample observations where $\overline{\boldsymbol{\theta}}_n$ is the ensemble mean of the EKI ensemble evolved based on the 4-day calibration suite of observations. The forward map outputs for the 2-day validation suite represent the model solution captured at days 1, 1.5 and 2; the forward map outputs for the 6-day testing suite represent the model solution captured at days 3 and 6.

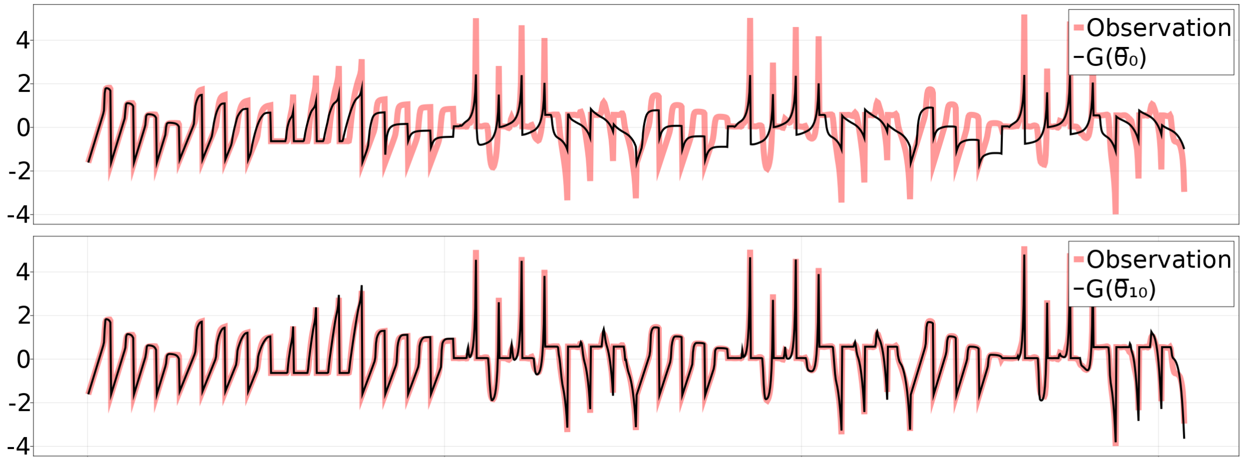

Figure 0-5: Forward map output $\mathcal{G}(\overline{\boldsymbol{\theta}})$ superimposed on the observation $\boldsymbol{y}$ where $\overline{\boldsymbol{\theta}}$ is the ensemble mean at iteration 0 (top) or iteration 10 (bottom). Evidently calibration brings the model into close agreement with the observation.

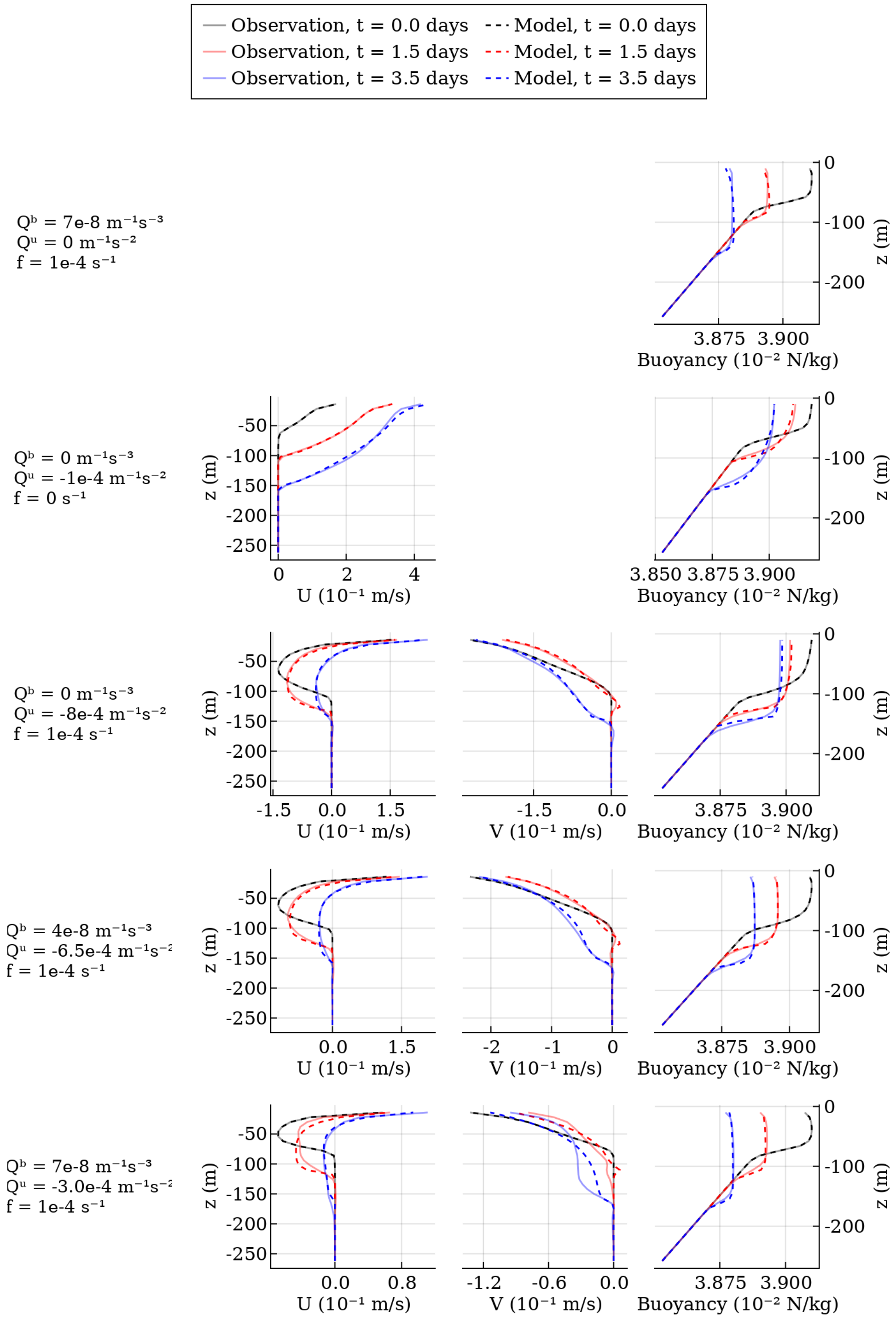

Figure 0-6: Predictive performance of the calibrated model compared to the LES truth at representative time snapshots from the 4-day suite of observations. Q^b is surface buoyancy flux, Q^u is surface momentum flux, and f is the Coriolis parameter.

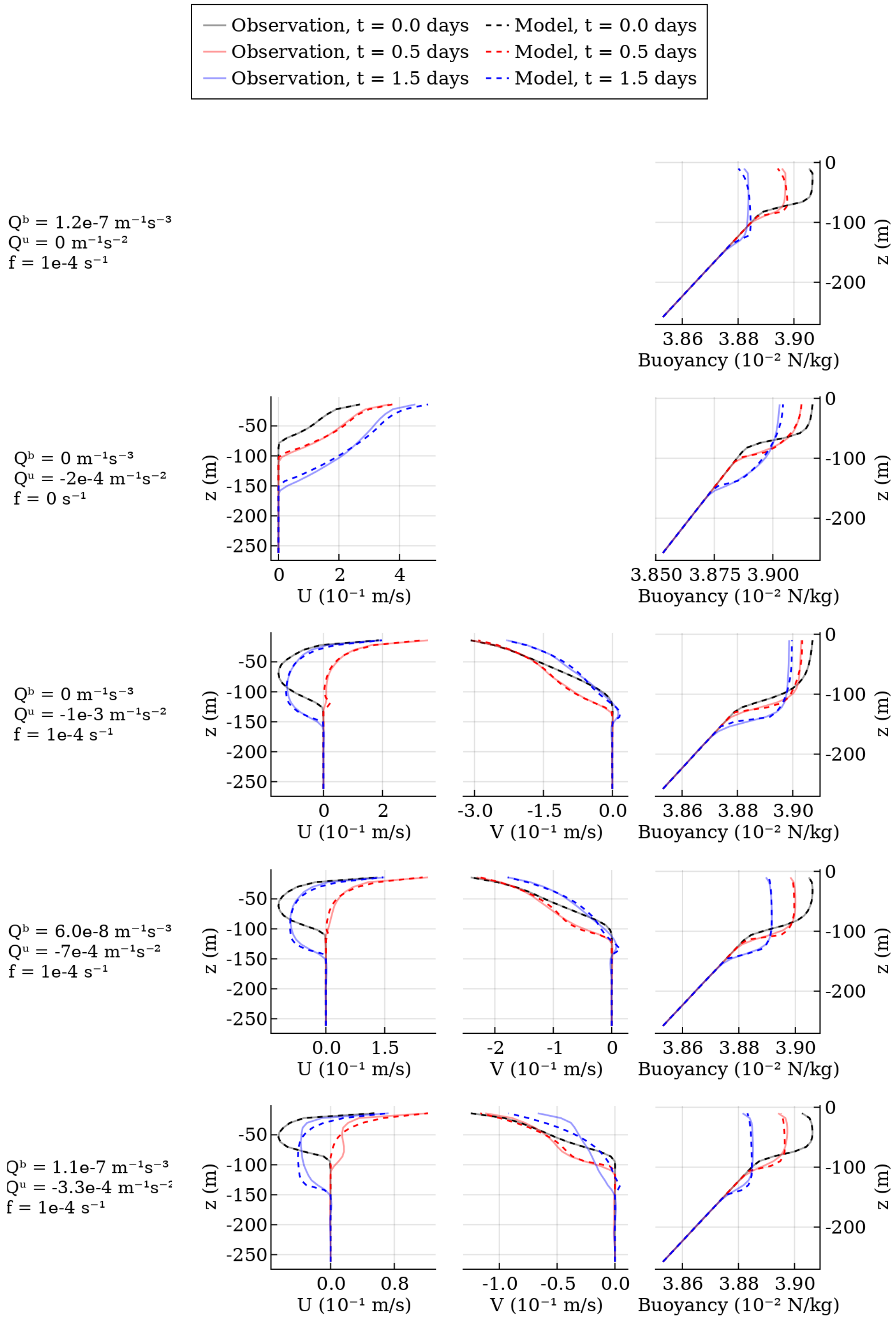

Figure 0-7: Predictive performance of the calibrated model compared to the LES truth at representative time snapshots from the 2-day suite of observations not employed for calibration.

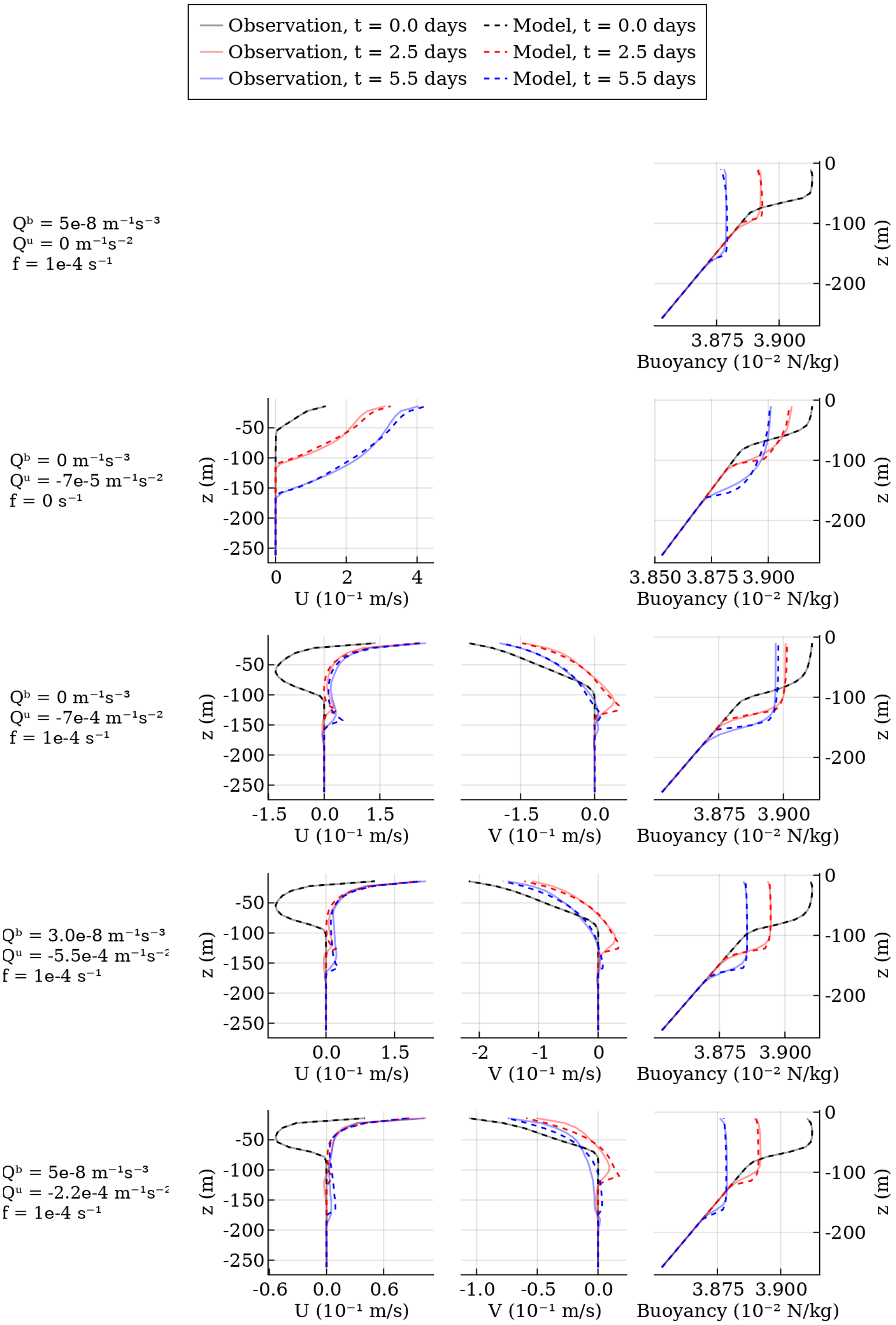

Figure 0-8: Predictive performance of the calibrated model compared to the LES truth at representative time snapshots from the 6-day suite of observations not employed for calibration.

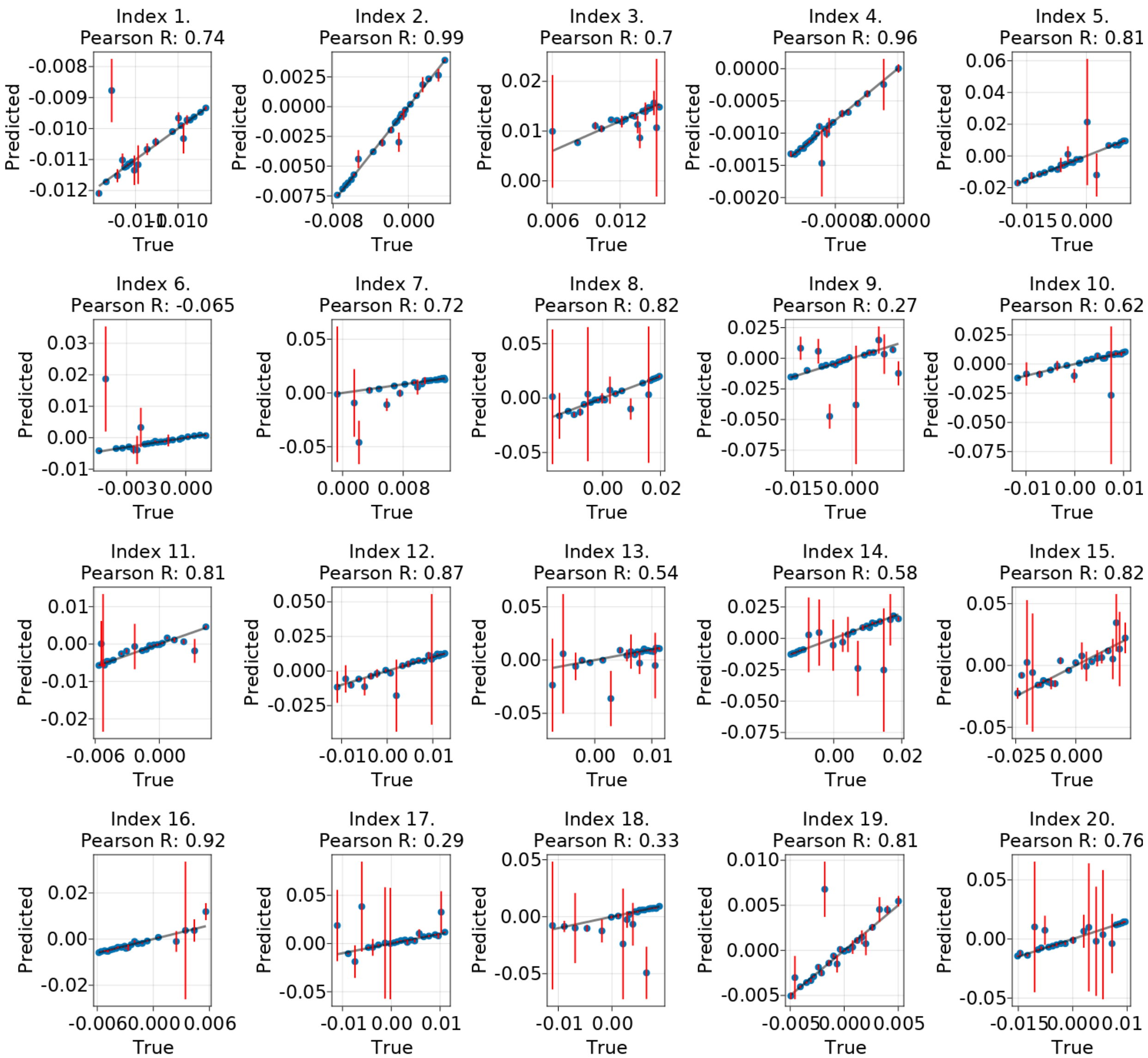

Figure 0-9: Performance of GP emulator components on 23 held-out training samples representing 3% of the overall training set. For given input $\boldsymbol{\theta}^{(n)}$ in the held-out samples, "true" denotes the true value $\hat{\boldsymbol{g}}_i^{(n)}$ of the i^{th} component of the transformed output generated by the true forward model $\mathcal{G}(\boldsymbol{\theta}^{(n)})$ applied to $\boldsymbol{\theta}^{(n)}$; "predicted" denotes the emulator mean prediction $m_i(\boldsymbol{\theta}^{(n)})$. The error bars represent the GP predictive uncertainty $v_i(\boldsymbol{\theta}^{(n)})$. Validation samples were selected not at random but by sampling at even intervals within the inter-quartile range of the sorted training targets for each given GP.

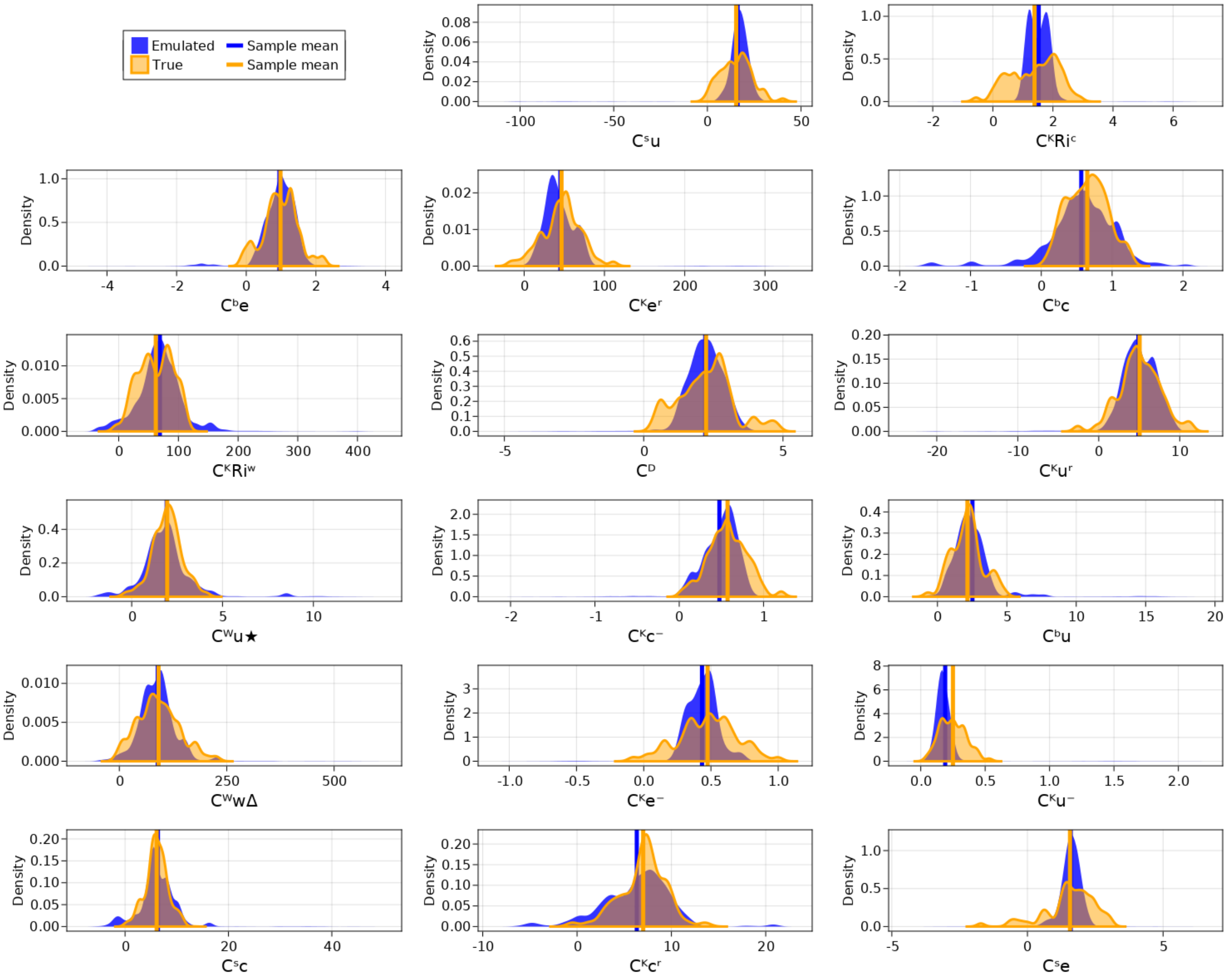

Figure 0-10: Marginal density distribution for samples generated by MCMC on the emulated objective (blue) and true objective (yellow).

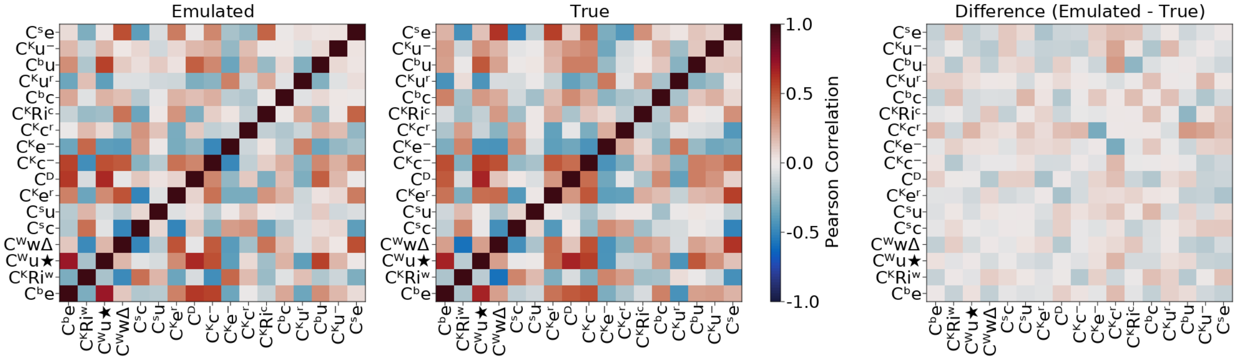

Figure 0-11: Pearson correlations between model parameters as estimated based on the samples generated by MCMC on the emulated objective (left) and the true objective (middle), along with the subtracted coefficients (right).

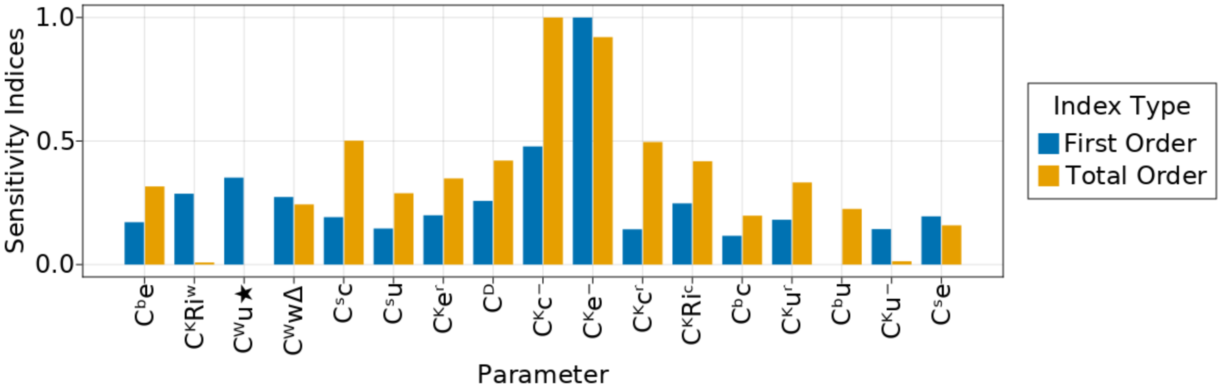

Figure 0-12: Min-max normalized Sobol sensitivity indices estimated based on 76,000 Quasi-Monte Carlo samples of the emulated objective function $\hat{\Phi}_{GP}$.

Bibliography

[1] M. A. Iglesias, K. J. H. Law, and A. M. Stuart, "Ensemble kalman methods for inverse problems," *Inverse Problems*, vol. 29, no. 4, p. 045 001, Mar. 2013. DOI: 10.1088/0266-5611/29/4/045001.

[2] E. Cleary, A. Garbuno-Inigo, S. Lan, T. Schneider, and A. M. Stuart, "Calibrate emulate sample," *Journal of Computational Physics*, 2021.

[3] L. Zanna and T. Bolton, "Data-driven equation discovery of ocean mesoscale closures," *Geophysical Research Letters*, no. 47, 2020.

[4] J. Yuval, P. A. O'Gorman, and C. N. Hill, "Use of neural networks for stable, accurate and physically consistent parameterization of subgrid atmospheric processes with good performance at reduced precision," *Geophysical Research Letters*, vol. 48, no. 6, e2020GL091363, 2021, e2020GL091363 2020GL091363. DOI: https://doi.org/10.1029/2020GL091363.

[5] A. Ramadhan, J. Marshall, A. Souza, G. L. Wagner, M. Ponnapati, and C. Rackauckas, "Capturing missing physics in climate model parameterizations using neural differential equations," *arXiv preprint arXiv:2010.12559*, 2020.

[6] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Science Advances*, vol. 3, no. 4, e1602614, 2017. DOI: 10.1126/sciadv.1602614.

[7] M. Dashti and A. M. Stuart, "The bayesian approach to inverse problems," in *Handbook of Uncertainty Quantification*, R. Ghanem, D. Higdon, and H. Owhadi, Eds. Cham: Springer International Publishing, 2016, pp. 1–118, ISBN: 978-3-319-11259-6. DOI: 10.

1007/978-3-319-11259-6_7-1. [Online]. Available: https://doi.org/10.1007/978-3-319-11259-6_7-1.

[8] F. Hourdin, T. Mauritsen, A. Gettelman, *et al.*, "The art and science of climate model tuning," *Bulletin of the American Meteorological Society*, vol. 98, no. 3, pp. 589–602, 2017. DOI: 10.1175/BAMS-D-15-00135.1. [Online]. Available: https://journals.ametsoc.org/view/journals/bams/98/3/bams-d-15-00135.1.xml.

[9] D. A. Yoder and P. D. Orkwis, "On the use of optimization techniques for turbulence model calibration," *Computers & Fluids*, vol. 214, p. 104 752, 2021, ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2020.104752. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045793020303224.

[10] B. Williamson D. and C. e. a. A.T. Hampton, "Identifying and removing structural biases in climate models with history matching," *Clim Dyn*, vol. 45, 2015. DOI: https://doi.org/10.1007/s00382-014-2378-z.

[11] F. Couvreux, F. Hourdin, D. Williamson, *et al.*, "Process-based climate model development harnessing machine learning: I. a calibration tool for parameterization improvement," *Journal of Advances in Modeling Earth Systems*, vol. 13, Mar. 2021. DOI: 10.1029/2020MS002217.

[12] J.-C. Golaz, V. E. Larson, J. A. Hansen, D. P. Schanen, and B. M. Griffin, "Elucidating model inadequacies in a cloud parameterization by use of an ensemble-based calibration framework," *Monthly Weather Review*, vol. 135, no. 12, pp. 4077–4096, 2007. DOI: 10.1175/2007MWR2008.1. [Online]. Available: https://journals.ametsoc.org/view/journals/mwre/135/12/2007mwr2008.1.xml.

[13] C. Schillings and A. Stuart, "Analysis of the ensemble kalman filter for inverse problems," *SIAM Journal on Numerical Analysis*, vol. 55, Feb. 2016. DOI: 10.1137/16M105959X.

[14] Z. Ding and Q. Li, *Ensemble kalman inversion: Mean-field limit and convergence analysis*, 2019. DOI: 10.48550/ARXIV.1908.05575. [Online]. Available: https://arxiv.org/abs/1908.05575.

[15] N. Kovachki and A. Stuart, "Ensemble kalman inversion: A derivative-free technique for machine learning tasks," Aug. 2018.

[16] W. D. Blaker, A. H. C., and et al., "Calibration and uncertainty quantification of convective parameters in an idealized gcm," *Journal of Advances in Modeling Earth Systems*, vol. 13, 9 2021. DOI: `https://doi.org/10.1029/2020MS002454`.

[17] A. Ramadhan, G. L. Wagner, C. Hill, *et al.*, "Oceananigans. jl: Fast and friendly geophysical fluid dynamics on gpus," *Journal of Open Source Software*, vol. 5, no. 53, p. 2018, 2020.

[18] A. Arakawa and V. R. Lamb, "Computational design of the basic dynamical processes of the ucla general circulation model," *General circulation models of the atmosphere*, vol. 17, no. Supplement C, pp. 173–265, 1977.

[19] D. S.-A. Neil K. Chada Yuming Chen, "Convergence acceleration of ensemble kalman inversion in nonlinear settings," *arXiv preprint arXiv:1911.02424*, 2021.

[20] M. Katzfuss, J. R. Stroud, and C. K. Wikle, "Understanding the ensemble kalman filter," *The American Statistician*, vol. 70, no. 4, pp. 350–357, 2016. DOI: `10.1080/00031305.2016.1141709`. eprint: `https://doi.org/10.1080/00031305.2016.1141709`. [Online]. Available: `https://doi.org/10.1080/00031305.2016.1141709`.

[21] J. Lei, P. Bickel, and C. Snyder, "Comparison of ensemble kalman filters under non-gaussianity," *Monthly Weather Review*, vol. 138, no. 4, pp. 1293–1306, 2010. DOI: `10.1175/2009MWR3133.1`. [Online]. Available: `https://journals.ametsoc.org/view/journals/mwre/138/4/2009mwr3133.1.xml`.

[22] "A dynamical systems framework for intermittent data assimilation," *BIT*, vol. 51, pp. 235–249, Mar. 2011. DOI: `10.1007/s10543-010-0302-4`.

[23] J. L. Anderson and S. L. Anderson, "A monte carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts," *Monthly Weather Review*, vol. 127, no. 12, pp. 2741–2758, 1999. DOI: `10.1175/1520-0493(1999)127<2741:AMCIOT>2.0.CO;2`. [Online]. Available: `https://journals.ametsoc.org/view/journals/mwre/127/12/1520-0493_1999_127_2741_amciot_2.0.co_2.xml`.

[24] C. Cotter and S. Reich, "Ensemble filter techniques for intermittent data assimilation - a survey," Aug. 2012.

[25] M. Mlikota and F. Schorfheide, "Sequential monte carlo with model tempering," 2022. DOI: 10.48550/ARXIV.2202.07070. [Online]. Available: https://arxiv.org/abs/2202.07070.

[26] M. A. Iglesias, M. Park, and M. V. Tretyakov, "Bayesian inversion in resin transfer molding," *Inverse Problems*, 2018.

[27] N. K. Chada, A. M. Stuart, and X. T. Tong, "Tikhonov regularization within ensemble kalman inversion," *SIAM J. Numer. Anal.*, vol. 58, pp. 1263–1294, 2020.

[28] X. Tong and M. Morzfeld, "Localization in ensemble kalman inversion," *Geophysical Research Letters*, 2022. DOI: https://doi.org/10.48550/arXiv.2201.10821.

[29] S. Reich and C. Cotter, *Probabilistic Forecasting and Bayesian Data Assimilation.* Cambridge University Press, 2015. DOI: 10.1017/CBO9781107706804.

[30] J. L. ANDERSON, "An adaptive covariance inflation error correction algorithm for ensemble filters," *Tellus A*, vol. 59, no. 2, pp. 210–224, 2007. DOI: https://doi.org/10.1111/j.1600-0870.2006.00216.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1600-0870.2006.00216.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1600-0870.2006.00216.x.

[31] S. Lunderman, M. Morzfeld, and D. J. Posselt, "Using global bayesian optimization in ensemble data assimilation: Parameter estimation, tuning localization and inflation, or all of the above," *Tellus A: Dynamic Meteorology and Oceanography*, vol. 73, no. 1, pp. 1–16, 2021. DOI: 10.1080/16000870.2021.1924952. eprint: https://doi.org/10.1080/16000870.2021.1924952. [Online]. Available: https://doi.org/10.1080/16000870.2021.1924952.

[32] Z. Ding, Q. Li, and J. Lu, "Ensemble kalman inversion for nonlinear problems: Weights, consistency, and variance bounds," *Foundations of Data Science*, vol. 3, no. 3, pp. 371–411, 2021.

[33] A. Garbuno-Inigo, F. Hoffmann, W. Li, and A. M. Stuart, "Interacting langevin diffusions: Gradient structure and ensemble kalman sampler," *SIAM Journal on Applied Dynamical Systems*, vol. 19, no. 1, pp. 412–441, 2020. DOI: 10.1137/19M1251655. eprint: https://doi.org/10.1137/19M1251655. [Online]. Available: https://doi.org/10.1137/19M1251655.

[34] M. Iglesias and Y. Yang, "Adaptive regularisation for ensemble kalman inversion," *Inverse Problems*, vol. 37, no. 2, p. 025 008, Jan. 2021. DOI: 10.1088/1361-6420/abd29b. [Online]. Available: https://doi.org/10.1088/1361-6420/abd29b.

[35] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning.* 2006, p. 19. [Online]. Available: http://www.gaussianprocess.org/gpml/chapters/RW2.pdf.

[36] I. Sobol, "Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates," *Mathematics and Computers in Simulation*, vol. 55, no. 1, pp. 271–280, 2001, The Second IMACS Seminar on Monte Carlo Methods, ISSN: 0378-4754. DOI: https://doi.org/10.1016/S0378-4754(00)00270-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378475400002706.

[37] M. J. Jansen, "Analysis of variance designs for model output," *Computer Physics Communications*, vol. 117, no. 1, pp. 35–43, 1999, ISSN: 0010-4655. DOI: https://doi.org/10.1016/S0010-4655(98)00154-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010465598001544.

[38] O. Roustant, "Analytical computation of sobol indices with a gaussian process metamodel," Nov. 2009.