Response to RC2

We thank the anonymous reviewer for their reading and suggestions. We have made some changes in the manuscript to reflect the concerns, and address them below, with the referee comments in blue.

Comment:

> This paper presented the SPUX-MITgcm framework, an approach to the calibration of a hydrodynamic model for a highly spatiotemporally heterogeneous observational dataset. The current form of the paper contains a lot of information and model exercises but was not presented in the most suitable way (missing key information or details) to guide the readers to well understand the value of the work. As such there are some difficulties in evaluating the work.

Answer:

We have reworked the description of the methodology with additional details about the sampler and the particle filter, and enhanced Figure 2 to provide a comprehensive visualization of the data assimilation framework and the particle filter, and an explanation of the error model where the Bi-LSTM network is used. The paragraph describing the EMCEE sampler now reads:

> *In our model, we employ a particle Markov Chain Monte Carlo (MCMC) method which is highly suitable for non-linear problems (Andrieu et al., 2010; Šukys and Bacci, 2021). The MCMC algorithm is used to infer selected hydrodynamic model parameters (see Sect. 2.4.2), with ensemble affine invariant sampler (EMCEE) for the parameter acceptance/rejection criterion. A visualization of the process is shown in Fig. 2. The EMCEE sampler is particularly effective for poorly scaled distributions that become well-conditioned under affine transformations, and can be significantly faster than standard MCMC approaches on highly skewed distributions. A more thorough discussion of the advantages and disadvantages is given in Goodman and Weare (2010). Here, we only provide a brief overview of the mechanism. The EMCEE algorithm initializes with an ensemble of Markov chains (walkers), $\{X_i\}$, drawn from a prior probability distribution $\Pi(x)$, and split into two subsets, S_1 and S_2, and their marginal likelihood is estimated as explained in the paragraph below. First, we update all the walkers from S_1 using the stretch formula $X_{j,new} = X_j + Z[X_j - X_k]$, where X_k is a randomly chosen walker from S_2, and Z is a scaling variable. Each proposed update $X_{j,new}$ is confirmed in accordance with a Metropolis acceptance probability. The next step is to update S_2 walkers using the updated S_1 elements. We continue alternatively evolving walkers from S_1 and S_2 until a suitable convergence criterion is met.*

With regards to the particle filter, we now have:

> *For each EMCEE parameter, a particle filter (PF) is deployed to address the stochasticity of the weather predictions. The PF, implemented in Šukys and Bacci (2021), works as follows. For each parameter $\alpha^{(k)}_i$, we initialize m model states, M_j, and simulate until an observation is reached at time $t = t_p$ (see Fig. 2). At this point, model simulations are paused and all particles are resampled (bootstrapped) according to their observational likelihoods. Thus, certain model states will be deleted and replaced by another state from a different trajectory. Such a resampling algorithm significantly increases algorithmic and implementation complexity due to the required destruction and replication of existing particles. However, it also provides an efficient way of sampling "intermediate" posterior model states. To the authors best knowledge, this is the first application of such a filtering algorithm to a fully three-dimensional model. A particular benefit of this approach is that the stochasticity from the atmosphere is sufficient to generate trajectories that manage to track the observational data with proper model parameters. This is in contrast to*

the non-physical correction vector in many other DA schemes that are necessary to nudge trajectories toward the data. Aside from potentially causing instabilities, these latter approaches decrease the confidence in the fidelity of the underlying model, as the correction mechanism potentially also corrects a model deficiency. At the same time, as no model is perfect, the SPUX PF offers limited capability to handle biases that the sampler does not eliminate. In addition, the strict nature of our PF (model states cannot be modified!) significantly limits its performance, and thus it cannot expect to outperform the above-mentioned established alternatives.

Thus, the difference is largely in the different design goals of the particle filtering methods. The SPUX PF seeks a realistic trajectory (at least within the confines of the hydrodynamic model), while EnKF and 4-Var seek to minimize model prediction errors.

Comment:

Details how the PF is configured and why, sensitivity analysis, convergence rate, etc.

We added a reference to the source material (*Šukys and Bacci, 2021)* where the PF is described in detail. In this paper, we provided all the main points of the filtering algorithm, and provide the relevant configuration details in Section 2.4.2. The number of particles/chains was largely determined by the size of the computational allocation at the CSCS cluster, and the amount of time the inference would require. We do not find these considerations particularly relevant, and thus do not include them in the text.

Comment:

The same for LSTM, LSTM framework should be introduced, how does it work, describe training and validation process and evaluation.

We modified the text to provide a more thorough description of the BiLSTM network we use in the model, together with a description of the data that we used to train it in Section 2.4.3. We also added Fig. 3, which provides a visualization of the operational BiLSTM module that we use. We also added a reference to a paper that provides an extension of the framework to accoutn for spatial features (Stalder et al., 2021). The relevant portion of Section 2.4.3 now reads:

[…] We implement a neural network using Bi-directional Long Short-Term Memory (BiLSTM) blocks that use the 27-hour history of 18 feature inputs to make a skin temperature prediction. 16 of the features come from the means and their respective spreads of the MeteoSwiss weather predictions (air temperature, cloud cover fraction, wind velocity, relative humidity, precipitation, short-wave and long-wave radiation). The last two features are the hydrodynamic model temperature predictions and hour of the day. The model was trained using data from 2018 and 2020, with the bulk water temperature predictions extracted from the Meteolakes model (Baracchini et al., 2020b). Data from 2019 was separated from the training for benchmarking purposes. A schematic of a BiLSTM block and the neural network is shown in Fig. 3. Input to the neural network is provided as a (27 time-step, 18 channel) tensor. An initial fully connected layer maps these 18 channels to 32 channels. The output is then sequentially passed through three separate BiLSTM cell blocks. Finally, the output of the last BiLSTM block is linearly mapped to a two-channel output, which corresponds to temporal predictions of the skin temperature and their predictive log-variances. In our experiments, using LSTMs were crucial in order to exploit historical patterns in the input features when predicting the skin temperature. In a recent work, an extension of the LSTMs to a spatially-dependent model is presented by Stalder et al. (2021).

For the particle filter, uncertainty quantification of the predicted skin temperature is also necessary. Accordingly, the BiLSTM blocks in our neural network randomly disables 30% of the LSTM units to induce stochasticity. This allows our model to also implement additional methods to quantify epistemic and aleatoric uncertainty (Kendall and Gal, 2017). Monte Carlo dropout approximates predictions from an ensemble that can be used to quantify epistemic uncertainty. On the other hand, using the negative log-likelihood of a normal distribution as the objective function in training allows BiLSTM to also estimate predicted variance that can be used to quantify aleatoric uncertainty. Specifically, for a given input of 18 feature observations over a 27-hour history, the neural network predicts the corresponding 27-hour skin temperature estimations as well as their estimated logarithmic variances. During the optimization phase of the neural network, negative log-likelihood of a normal distribution as in Kendall and Gal (2017) (eqn 8), is used with skin temperature estimations ($\hat{y}_i$) and corresponding logarithmic variance estimation (s_i). At test time, we generate model predictions for 19 times for each input while keeping dropouts within BiLSTMs active, yielding 19 different prediction vectors, similar to an ensemble model. While the mean of estimated skin surface temperature are used for mean estimates, we use the variance of the 19 predictions for epistemic uncertainty. Aleatoric uncertainty is computed from the predicted variance estimates by taking their average after mapping the predicted logarithmic variance into variance. The total scalar variance is computed as the sum of the two variances. Accordingly, we construct a normal distribution with the computed total variance centered at the mean skin temperature prediction to be evaluated against the LSWT measurement.

Comment:

how does EMCEE Sampler work, how did you evaluate its performance?

Section 2.4.1 has been rewritten to provide greater detail on the EMCEE sampler, and Figure 2 has been greatly expanded to visualize the process of sampler initialization, parameter proposal via the "stretch" move, and an example of how the parameters might evolve.

The performance of the EMCEE sampler is evaluated in Section 3.1, where we conclude that the sampler has converged based on the evolution of the Markov chain parameters, and therefore the sampler has performed satisfactorily. On the other hand, in the supplementary material Section 3 we demonstrate a case of the sampler diverging in an attempt to calibrate dynamic model parameters (eddy diffusivity/viscosity). Thus, the sampler is not guaranteed to work. In the conclusion, we state:

In addition, as discussed in the supplementary material, the sampler has significant difficulty with calibrating certain model parameters (although this issue could potentially be mitigated with a better error model). Therefore, we feel that a computationally cheaper method for parameter estimation (for example, an optimization algorithm instead of a sampler) might be the more productive approach. At the same time, an improved version of the particle filtering approach could provide a powerful option for operational forecasting models.

In the end, we conclude is that the approach will require more work to be viable for 3D data assimilation problems.

Comment:

> I am also not quite sure about the BiLSTM's role in the proposed framework.

We improved the presentation of the methodology to be clearer as to how the BiLSTM network fits into the framework. As shown in the revised Figure 2, the purpose of the network is to provide an observation operator, which maps the predicted bulk temperature from the hydrodynamic model onto skin temperature to be compared to LSWT data.

Comment:

> A better highlighting of the novelty and achievement of the work in the context with comparison to a similar or alternative approach. My impression is that it is a novel framework applied to the 3-D model, but not fully sure how effective and efficient it improve the simulation results. I think this can be significantly improved if the authors can re-structure the manuscript to highlight key information about the models used in the study.

In terms of parameter calibration, we agree that a comparison to a different MCMC type of sampler or a completely different calibration methodology (such as DUD) could be quite beneficial. However, such a comparison would require the implementation of such a sampler in SPUX which would also be compatible with the hydrodynamic model, MITgcm. This would require a significant time investment, and thus was not completed. At the end of the paper, we conclude that further developments, such as a more effective PF, are necessary before the framework becomes usable.