Referee's report on
Model calibration using ESEm v1.0.0 -an open, scalable Earth System Emulator
by Watson-Parris, Williams, Deaconu and Stier

Referee: Victoria Volodina

General comments

This paper presents ESEm, a Python library for emulating and calibrating Earth system models. Three widely used emulation techniques such as Gaussian process, Neural Network and Random Forest are provided as part of the tool. In addition, the ESEm also includes two calibration methods, specifically approximate Bayesian computation (ABC) and Markov-Chain Monte Carlo (MCMC). The authors highlight the importance of the proposed tool on three selected case studies. The goal of the authors is to propose an open-source and general tool for emulating and calibrating Earth system models.

Overall this is an interesting article. The authors managed to identify a gap in the process of tuning climate models and proposed a general tool to perform reproducible research. The authors managed to provide a range of examples to strengthen the case for using ESEm. However, I have a number of questions about the descriptions of emulation engines and calibration techniques outlined in the Specific comments section.

In addition, the authors claim that "no general-purpose toolset exists for model emulation in the Earth sciences", which is to the best of my knowledge is true. However, there has been extensive work done as part of the HIGH-TUNE project on tuning boundary-layer clouds parameterization with in-house developed High-Tune Explorer (htexplo) tool using GP emulators and multi-wave history matching [1, 2, 4]. The authors might find the description of the tools, the comparison with by-hand tuning and discussion about model discrepancy useful and constructive for their manuscript.

Specific comments

Further comments and questions are as follows.

1. In Section 3, it would be more instructive to provide a mathematical definition of the simulator and the respective surrogate model. In particular, I propose to move the mathematical formulation in Section 4, lines 277-279 to Section 3.

2. In Section 3.2, it would be helpful if you could provide a mathematical definition of the GP model. What form of the mean function are you using? Lines 167-170: you mentioned the reduction of input space using information criterion. Does it affect the number of terms in your mean function and/or kernel function? A helpful reference would be [1].

3. Line 204: for the demonstrator example, you used the 'Bias+Linear' kernel. Is there any connection between the Bias kernel and nugget term commonly specified for GP emulator? Is it a standard kernel choice?

4. In Section 3.3 Random Forests, I am left wondering whether Random Forest emulation would potentially be useful for approximating model responses with nonstationarity and discontinuity due to the binary partitions over the training data. Could you comment on this?

5. In Figure 2, there is a comparison between GP and CNN emulators. Could you also consider the Random Forests emulation strategy? If not, could you explain why?

6. In Section 4, line 278 you introduced a function $\mathcal{F}$ such that $\mathcal{F}(\theta) = Y$. However, in Equation (3), we observe that Y is itself a function of θ. Please revisit your notation.

7. In Section 4.1, I had some difficulties in following the description of Approximate Bayesian Computation (ABC). As I understand the authors are using ABC to approximate the likelihood $p(Y^0|\theta)$ in Equation (1) with samples from the simulator Y. In Equation (2), the authors defined

$$p(\theta|Y^0) \propto p(Y^0|Y)p(Y|\theta)p(\theta).$$

After comparing Equation (2) with Equation (1), we deduct that $p(Y^0|\theta) = p(Y^0|Y)p(Y|\theta)$, which cannot be right. Instead we require integration with respect to Y, i.e.

$$p(Y^0|\theta) = \int p(Y^0|Y)p(Y|\theta)dY$$

for this expression to be true. Therefore, Equation (2) becomes

$$p(\theta|Y^0) \propto \int p(Y^0|Y)p(Y|\theta)p(\theta)dY \approx \int I(\rho(Y^0, Y) \leq \epsilon)p(Y|\theta)p(\theta)dY.$$

I am not an expert in ABC, but in Equation (2) we have an approximate sign ($\approx$), because you approximate probability function $p(Y^0|Y)$ with $I(\rho(Y^0, Y) \leq \epsilon)$. Is it right? Perhaps it would be useful to provide readers with some ABC references.

8. I have difficulties in following Equation (3). In particular, the implausibility function commonly used in history matching is defined in terms of the first two moments, expectation and variance of the emulator. Instead in their implausibility computations, the authors use simulator output $Y(\theta)$ directly together with the emulator variance σ_E^2, which does not make sense. The implausibility function in Equation (3) should have the form

$$\rho(Y^0, Y(\theta)) = \frac{|Y^0 - \mu_E|}{\sqrt{\sigma_E^2 + \sigma_Y^2 + \sigma_R^2 + \sigma_S^2}},$$

where μ_E and σ_E^2 are the mean and variance of emulator respectively.

9. In Section 4.1, lines 333-343: the authors briefly discuss implausibilities for multiple observations. It would be useful to mention and reference multi-dimensional implausibility commonly used in history matching considered by Craig et al. (1996) and Vernon et al. (2010).

10. In Section 4.1, the authors provided an example to illustrate the ABC approach. I am curious to find out the percentage of input space that was retained, i.e. plausible space of parameters. This is a standard measure in history matching that could help to emphasise the importance of the proposed method.

11. In Section 4.2, lines 384-385: "...this discrepancy can be approximated as a normal distribution centred about zero... ". However, in Equation (5), $p(Y^0|Y)$ is a probability density function of a normal distribution centred around Y. Could you please clarify this point? Again, I am confused if the authors are using the simulator itself Y instead of the emulator's mean and variance?

Technical corrections

1. Line 65: could you decipher the abbreviation ML in "prevalent use in other areas of ML"?

2. Line 115: Please provide the reference to maximin latin-hypercube sampling [3] in "The parameter sets are created using maximin latin-hypercube sampling..."

3. Figure 4 is hard to follow. It would be helpful to remove inset plot and produce two separate plots next to each other.

4. Figure 5: CMIP6 ScenarioMIP outputs and the multi-model mean for each scenario is very hard to detect from the provided plot.

References

[1] Fleur Couvreux, Frédéric Hourdin, Daniel Williamson, Romain Roehrig, Victoria Volodina, Najda Villefranque, Catherine Rio, Olivier Audouin, James Salter, Eric Bazile, et al. Process-based climate model development harnessing machine learning: I. A calibration tool for parameterization improvement. *Journal of Advances in Modeling Earth Systems*, 13(3):e2020MS002217, 2021.

[2] Frédéric Hourdin, Daniel Williamson, Catherine Rio, Fleur Couvreux, Romain Roehrig, Najda Villefranque, Ionela Musat, Laurent Fairhead, F Binta Diallo, and Victoria Volodina. Process-based climate model development harnessing machine learning: II. Model calibration from single column to global. *Journal of Advances in Modeling Earth Systems*, 13(6):e2020MS002225, 2021.

[3] Max D Morris and Toby J Mitchell. Exploratory designs for computational experiments. *Journal of statistical planning and inference*, 43(3):381–402, 1995.

[4] Najda Villefranque, Stéphane Blanco, Fleur Couvreux, Richard Fournier, Jacques Gautrais, Robin J Hogan, Frédéric Hourdin, Victoria Volodina, and Daniel Williamson. Process-Based Climate Model Development Harnessing Machine Learning: III. The Representation of Cumulus Geometry and Their 3D Radiative Effects. *Journal of Advances in Modeling Earth Systems*, 13(4):e2020MS002423, 2021.