Author Response to RC2: "Computationally efficient subglacial drainage modelling using Gaussian Process emulators: GlaDS-GP v1.0"

Reviewer: Jacob Downs

Tim Hill, Derek Bingham, Gwenn E. Flowers, Matthew J. Hoffman

Reviewer comments are in black and we provide our responses in blue.

This work outlines a Gaussian Process based emulator of the GlaDS subglacial hydrology model. Inputs to the emulator consist of 8 scalar model parameters, while outputs include either spatio-temporal flotation fraction fields or scalar metrics describing bulk properties of the subglacial drainage system. Performance of the emulator is thoroughly examined by comparing emulators using different training sets as well as using different numbers of principal components to represent flotation fraction fields.

Overall, this manuscript is well written, the methods are clear, and I appreciate the rigorous evaluation of the emulator's accuracy. In light of this, I believe this work represents a good contribution to the field. However, I believe that this work would benefit from a more detailed discussion of how the GP emulator compares to existing emulators. In particular, I think the discussion should outline more of the key differences between this work and Verjans and Robel [2024], including differences in their potential use cases.

The authors should outline in specific terms where their emulator could be applied versus other existing methods. For instance, is this purely a tool for calibrating subglacial hydrology parameters or assessing sensitivity? Is there a path for using this as a substitute for GlaDS to predict effective pressure when coupled with an ice sheet model? What do the authors intend to do with this emulator, or what might future work on the emulator entail? As someone interested in this space, I was hoping the authors might delve more deeply into some of these topics in the discussion.

Thank you for taking the time to review our work in detail and for the constructive suggestions. We have responded to your comments individually below.

Line 52 Might benefit from rewording to"The Gaussian Process emulators we develop take subglacial drainage model parameters as inputs and predict spatially and seasonally resolved flotation fraction ... "
We have revised this wording as suggested to improve the clarity of this sentence.

Line 63 Maybe this could be reworded as "where the radius of channels is modeled as a balance between the creep closure of ice and opening by melt. "
We have revised this to say, "[...], with the channel ~~network emerging from~~ **radius determined by** the balance between creep closure of ice and opening by melt".

Line 65 "The continuum distributed (sheet) drainage system is defined on the mesh nodes, with possible channel locations defined by element edges."
This could be clarified. Maybe you could say that variables describing the distributed system are represented by linear finite elements with degrees of freedom on mesh nodes? This would help clarify that

the distributed drainage system isn't "confined" to the mesh nodes, but that's where the degrees of freedom are defined.
We have used the suggested, more accurate statement: "The governing equations, arising from conservation of mass and energy, are discretized on an unstructured triangular mesh. Variables describing the continuum distributed (sheet) drainage system are represented using finite elements with degrees of freedom located on the mesh nodes, with possible channel locations defined by element edges."

Line 83 "For details on Gaussian Processes"
Corrected, we have revised these citations to more precisely point the reader to the relevant references: "For background on Gaussian Processes see Jones et al. (1998) and Rasmussen and Williams (2005), and see Higdon et al. (2008) for a complete description of the emulators constructed here"

Line 130 Please mention why there are d+1 hyperparameters. Also, depending on the kernel, this number could vary.
Thank you for highlighting this. The reviewer is correct, while it is usually d+1 hyperparameters, the number can vary. Since we have not introduced a particular kernel function, it's not appropriate here to say how many hyperparameters there are yet. We have removed the number of hyperparameters from this statement: "The fact that the GP model is simple enough to allow for Bayesian inference of the emulator hyperparameter values θ, where uncertainty in the hyperparameters is reflected in the uncertainty in the emulator predictions, is a key advantage compared to a neural network for uncertainty quantification"

Line 170 That seems reasonable. If I understand, would it be fair to say that you hope that much of the spatial/temporal complexity is captured in the principal components while the response of the coefficients of the principal components to variations in parameters is expected to be smooth?
Yes, this is precisely what we are trying to say, and we have revised this section to read: "While the flotation fraction field need not be smooth in space and in time, the principal components w_{ij} (θ) tend to vary smoothly with respect to the GlaDS parameters since the the spatiotemporal complexity is captured by the principal component basis."

Line 285 Interesting, I like the commentary on the first couple principal components, and I think interpretability is a really nice advantage of this method.
Thank you for the nice comment, and we agree that the interpretability of this method is a nice feature.

Line 321 "Using 8 PCs reduces the height..."
Corrected.

Figure 4 I'm not sure what is meant by 95% prediction interval in c and f. Is this the prediction uncertainty of the emulator integrated through time and space? Also, when considering the prediction error, is this accounting for uncertainty in θ (sampled via MCMC) as well as the GP prediction uncertainty? Or is it just the latter, and you are using the most probable estimate for the hyperparameters? Presumably the advantage of using MCMC on θ as opposed to maximum likelihood estimation is to characterize its effect on uncertainty as well?
Fig. 4c and 4f show the prediction interval averaged through time and space. We have added an explanation of this figure to the text: "Fig.4a--c show the distribution of the RMSE, MAPE and the

spatiotemporally averaged 95% prediction interval across the test ensemble for emulators constructed using different numbers of PCs. Since GP prediction uncertainty varies across the space of emulator inputs depending on the distance to training runs, we assess the overall prediction uncertainty by computing a Monte Carlo integral across the space of GlaDS parameters, indicated as black circles in Fig. 4c, f)."

The 95% intervals include (1) uncertainty in the hyperparameters, (2) GP prediction uncertainty, and (3) uncertainty arising from the truncated basis via the error term and λ_{sim}. We have added a Figure providing an overview of the methods, including a summary of the algorithm used to draw posterior realizations from the GP. We hope this figure helps to clarify the uncertainties included in the emulator predictions.

Line 340 Before discussing results for the test cases with different levels of error, please introduce what the "high-error" and "median-error" simulations are. I think I found this information in the figure 6 and 7 captions, but it would be helpful to include it in the text.

Good suggestion, we have explained these test cases in the text: "To assess how emulator performance varies across the test set, we evaluate the performance on test simulations with 95th-percentile ("high error"), 50th-percentile ("median error") and 5th-percentile ("low error") RMSE."

Figure 9 It's difficult to see the differences in the median error across different numbers of simulations due to the scale of the outliers.

This is true, especially for the log-sheet transit time (row 2). We would also like to point out that the difficulty in seeing changes in the mean is a useful interpretation of the experiment. For these scalar metrics, it requires only ~32 simulations to obtain predictions with a median error (bias) near zero. Adding more simulations primarily reduces the error in the worst-case scenarios (outliers and the spread of the whiskers) and the uncertainty in predictions.

Section 6.4 I think some of this discussion could be expanded to highlight more of the nuances of the different approaches. For example, the emulator in Verjans and Robel [2024] aims to be fairly general purpose, and it's more of a one-to-one substitute for GLaDS. Hence, I see their neural network based approach as fundamentally different from yours in its intent. This also makes the comparison of the number of parameters difficult as the input / output spaces in Verjans and Robel [2024] is very broad, meaning more parameters are likely needed.

There are certainly a number of appealing elements to your GP approach, including interpretability, the speed at which it can be trained, and how you also obtain uncertainty estimates without additional work. But I feel like its difficult to directly compare your emulator to Verjans and Robel [2024], and I have a hard time seeing the two emulators being used in the same way. In this sense, I see your emulator as being far more comparable to Brinkerhoff et al. [2021].

Thank you for the suggestion, and we agree with your assessment that our approach is most similar in spirit and applications to Brinkerhoff et al. (2021). We have added a discussion about the utility of these three studies to Section 6.4:

> The GP emulator approach that we have described is closest in spirit and in practical applications to that of Brinkerhoff et al. (2021). By emulating model outputs for different model parameter values, the GP emulator constructed in this study and the Brinkerhoff et al. (2021) neural network emulator are well-suited for quantifying parametric uncertainty, calibrating model parameters

given data and exploring parameter sensitivity (e.g., Fig. 11). Both approaches use a principal component decomposition that nicely introduces interpretability for the emulator (e.g., Fig. 3, 11). Aside from structural differences in the type of emulator, the major differences between our work and that of Brinkerhoff et al. (2021) is that we explicitly resolve subglacial water pressure and drainage characteristics and we obtain a built-in prediction uncertainty estimate, whereas Brinkerhoff et al. (2021) implicitly represent subglacial conditions through the influence on surface velocities and take extra steps to estimate prediction uncertainty. Both approaches are tied to a particular study area, limiting their utility for large-scale forward modelling. On the other hand, Verjans and Robel (2024) use a convolutional neural network that can generalize to arbitrary melt forcing and study areas, making it an ideal tool for forward modelling of ice-sheet evolution forced with a basal boundary condition that is influenced by the hydrology emulator. Since Verjans and Robel (2024) do not predict water pressure for different model parameters, their emulator is not ideally suited for uncertainty quantification, calibration of drainage model parameters or sensitivity analysis.

Line 481 To me, saying that IGM enforces conservation of momentum sounds as if it is enforced strictly. Although IGM uses a physics-based loss function, conservation of momentum is only upheld approximately. Contrast this with something like Horie and Mitsume [2024], in which a value is strictly conserved in a neural network.
This is a good point, we have updated this wording to clarify that "Jouvet et al. (2023) use a loss function that is based on conservation of momentum as part of a neural network ice-flow velocity emulator", rather than strictly enforcing their PDE constraint.

Section 6.6
I think this manuscript would significantly benefit from elaborating on specific use cases for this emulator. For instance, do you see the emulator being used more or less as is, or do you think the value of this work is in the general approach that you present? You mention uncertainty in future sea level rise, but not a clear application of the emulator for this purpose.

There is a pretty clear use case for using the emulator to do Bayesian calibration of subglacial hydrology model parameters, but what other use cases might it have? Do you see a path forward for coupling effective pressure fields from the emulator to an ice sheet model? Clarifying the intended use of this emulator or more concrete pathways for other applications would really strengthen the discussion.

Thank you for the suggestion to improve this discussion. We have added two paragraphs to Section 6.6 to more clearly lay out where we see this methodology being used:

- Calibrating parameters of the subglacial drainage model using observations of quantities corresponding to GlaDS outputs, such as borehole water pressure or moulin water level, tracer transit times, or channel characteristics inferred from passive seismic measurements.
- Using emulated effective-pressure fields as inputs to a sliding law to characterize the sensitivity and related uncertainty of ice flow (e.g., solid-ice discharge to the oceans) to drainage model parameters.

- More broadly, the GP methods have not been extensively tailored to the subglacial drainage application, they have a place as part of uncertainty quantification across a range of glaciological processes.

For the reviewer's information, we would like to highlight that we have a recent preprint that applies the GP approach described in this manuscript to the calibration task: https://doi.org/10.31223/X5GQ68. We have added a reference to this preprint when we point to Bayesian calibration of drainage model parameters as an extension of the present work.

References

Douglas Brinkerhoff, Andy Aschwanden, and Mark Fahnestock. Constraining subglacial processes from surface velocity observations using surrogate-based bayesian inference. 67 (263):385–403, 2021. ISSN 0022-1430, 1727-5652. doi: 10.1017/jog.2020.112. URL https://www.cambridge.org/core/journals/journal-of-glaciology/article/constraining-subglacial-processes-from-surface-velocity-observations-using-surrogatebased-bayesian-inference/1E03CA805D8CE6A0C310108540D9457E. Publisher: Cambridge University Press.

Masanobu Horie and Naoto Mitsume. Graph neural PDE solvers with conservation and similarity-equivariance. 2024. URL https://openreview.net/forum?id=WajJf47TUi.

Vincent Verjans and Alexander Robel. Accelerating subglacial hydrology for ice sheet models with deep learning methods. 51(2):e2023GL105281, 2024. ISSN 1944-8007. doi: 10.1029/2023GL105281. URL https://onlinelibrary.wiley.com/doi/abs/10.1029/2023GL105281.
eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2023GL105281.