R2

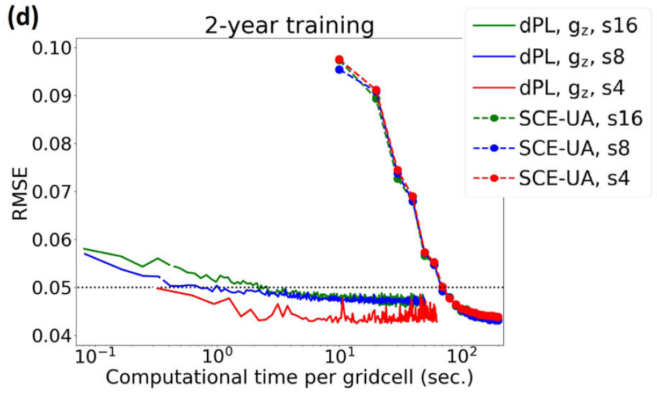
Overall, the manuscript is well-written with a clear research objective, innovative hybrid models, and solid results. The study compared the performance of the commonly used LSTM model with two differentiable hydrological models (static and dynamic parameters) with a temporal generalization experiment and conducted a comparative analysis for traditional “prediction in ungauged basins” problem. The model evaluation results provide valuable insights into improving the mechanism of the hydrologic models, confirming the strong localization and extrapolation capabilities of differentiable models. Below are some key comments and concerns:

Thank you for your evaluation.

L161-166: From my understanding, the original parameter calibration process of the HBV model has been replaced by the parameter calibration process of the g_A neural network. If this is the case, it is still necessary to run the HBV model. How does this approach compare to the traditional hydrological model calibration methods in terms of modeling speed? Has it resulted in time and labor savings in the modeling process?

Good question. It resulted in orders of magnitude of computational savings, and we’d argue labor saving as well, but it depends on the economy of scale. Below we will provide some background. We realize that many people have not read our previous paper (Tsai et al., 2021), and we will include some of discussion below in the revised manuscript.

There is an obvious time saving in parallelism. During training, we run models in parallel over a “minibatch”, which means somewhere ~100 basins at a time, before we calculate the loss and update the parameters. This is happening all in parallel due to easy GPU concurrency via PyTorch.
There is a less obvious economy of scale, which is that the training and the learned knowledge of neural networks are shared by all sites. This means the more sites participating in the training, the more efficient it becomes (see the figures below, from Tsai et al., 2021). Hence, if you run differentiable model training for one of a few sites, you won’t be as efficient as traditional optimization. If you run it for 10000s of sites, you will become orders of magnitudes more efficient. The number we cite in Tsai et al., 2021 is that, for the same calibration job over entire USA, a traditional evolutionary algorithm would need a 100-processor cluster to run 2-3 days, whereas our differentiable parameter learning needs 1 single GPU for 1 hour.



We argue it actually saves human effort as well, because you only need to set up the learning once and train it once, and you can produce parameter fields for the entire USA! We were never able to do this easily in the past! Again, obviously, this contrast depends on how many sites you are training together (Figure 5 from Tsai et al., 2021).

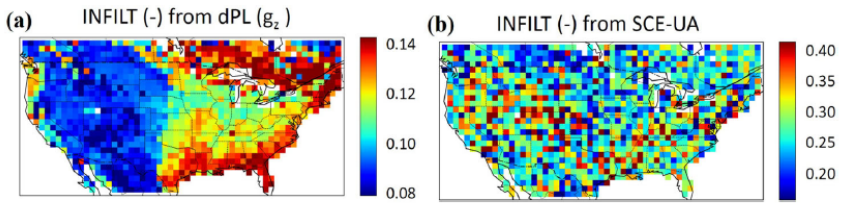


Fig. 5 Comparison of parameters generated by dPL and SCE-UA. The continuous, spatially representative patterns of **a** dPL-inferred parameters are noteworthy, especially in comparison to the discontinuous, random appearance of **b** SCE-UA-inferred parameters from site-by-site calibration. Both were trained with a $1/8^2$ sampling density.

In fact, the more sites that participate in training, the more robust (higher quality) the model becomes. There is a beneficial scaling relationship that is very alien to us geoscientists. We explored this in Tsai et al., 2021 as well.

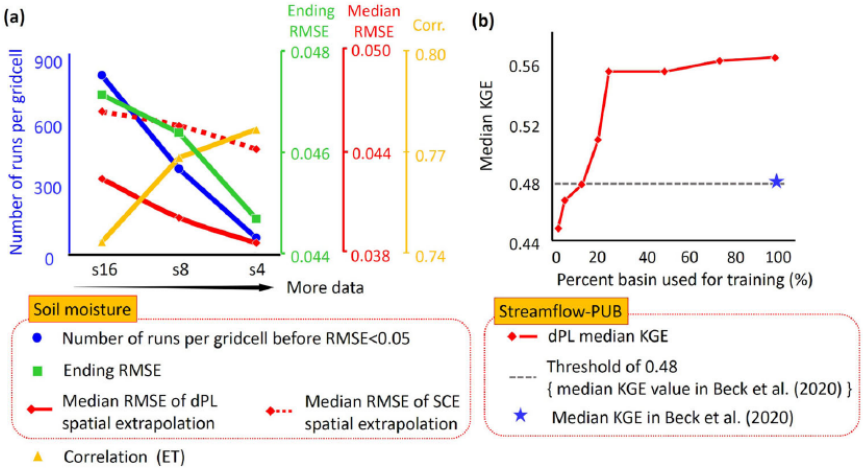


Fig. 7 Scaling curves of dPL. a Training data sampling density increases from s16 to s4. Dashed and solid red curves share the same red *y* axis (dashed line is for SCE-UA, to enable comparison). **b** Scaling curve for the spatial extrapolation (PUB) test with the Beck20 global headwater catchment dataset.

Wen-Ping Tsai*, Dapeng Feng*, Ming Pan, Hylke Beck, Yuan Yang, Kathryn Lawson*, Jiangtao Liu*, and Chaopeng Shen. From calibration to parameter learning: Harnessing the scaling

effects of big data in geoscientific modeling. *Nature Communications*, (2021). doi: 10.1038/s41467-021-26107-z.

L200-205: Employing FHV and FLV is a thoughtful model evaluation strategy. In contrast to relying solely on integrated metrics like KGE, FHV and FLV offer a more thorough evaluation of model performance. Has the author considered whether integrated metrics such as KGE are suitable for capturing the distribution characteristics of state variables? Additionally, is the loss function used during the neural network parameter determination suitable for the data's distribution characteristics? This is crucial, as KGE includes a term related to Pearson correlation coefficient, which may not be applicable to distributions beyond normal. Such considerations are essential to avoid potential misguidance in the model calibration process.

We will add some comments regarding this point in the revision. In our experience, KGE tends to emphasize the peaks even more than NSE. Please note that KGE was used as an evaluation metric in the paper but the main calibration objective was not KGE --- it was a mixture of mean squared error and root-mean-squred error, as described in the paper. We are aware of the distribution issue raised by the reviewer. As described in an earlier paper (Feng et al., 2020), we applied a logarithmic transformation which made the residual more normal. However, later on this was found to be not as crucial as we initially thought, because NNs are good at minimizing any errors that is minimization and the tradeoff is not that strong.

it should be a separate paper that discuss the selection of metrics, for example, see this one https://hess.copernicus.org/articles/23/4323/2019/.

L208-212 & Figure 3: Are the evaluation results corresponding to the training set, the testing set, or the overall results from both? The assessment outcomes should be provided separately for the training set and the testing set to enable a clearer evaluation of the model's performance and identification of any issues within the model.

Metrics are reported for the validation data. We will clarify this.

L222-223: The structural issues might explain the errors with peak flow in differentiable models. However, it raises a question as to why LSTM also exhibits errors in peak flow. As mentioned earlier, the utilization of inappropriate evaluation metrics could contribute to the low FHV across all models. It requires a more in-depth consideration of how metrics impact the calibration results.

We will add some comments in the revised manuscript. Our hypotheses are (i) underestimation of precipitation inputs especially for large storms; (ii) to capture extremes, we need smaller basins and probably run models on hourly time steps to capture the spatial heterogeneity in rainfall and nonlinear effects; (iii) extremes are not well observed so the training of NNs may be problematic for extremes; (iv) as you mentioned, some consideration with calibration metrics as well, but some previous effort including from Kratzert et al., have studied this pretty extensively

and they were not able to find a way to improve extreme metrics. We think much more effort is needed to improve extreme representation and we welcome community effort in study this topic.