In the first rounds of reviews, I commented on three topics:

1. Model setup (single basin LSTMs vs regional LSTMs).
2. Selection of input variables.
3. Data splits.

In most parts, the authors argued that my concerns do not require any change to their manuscript apart from adding a new section on “Limitations and Outlooks” that basically lists these concerns.
In one point (selection of input variables), I think it is a pity that the authors limit themselves to their selection and do not report potential novel insights (see detailed comment below). On another point (model setup), I do not agree with the argumentation of the authors, why they think that an inferior model calibration procedure is justified, but in the end it is their decision.
The last point (data splits), is probably the most critical. Reading that “*the model was used for statistical purposes instead of a prediction task”*, which according to the authors justifies the use of a data split that is not independent of the training data is in my opinion a red flag for any kind of study that involves a model. Even if the results would not change (what I sincerely hope for the authors) it is a scientifically wrong thing to do. And to justify it by saying a per-gauge time split is more complicated is in my opinion a really bad argument.

I added some specific comments to the three points below but in general, I have nothing much to add to what I said in the first round. Ultimately, it is up to the editor to decide how to proceed.

1. Model Setup

The topic of discussion is, whether it is necessary/recommended or not for this study, to use a regional training setup with LSTMs rather than the per-basin calibrated LSTMs the authors chose to use. Here you say

“*However, as you mentioned as well in your comment, the choice of which strategy to apply (local model vs. regional model) should be in line with the research purpose.*”

No, this is not what I have mentioned. I deeply believe that no matter what you want to do with a model, you should always set up a model in the state-of-the-art for whatever model you use. Again, you would probably not calibrate a conceptual hydrology model in a regional fashion if you are interested in a single basin.
There is no reason to believe that a model that is consistently and significantly better than another model (or the same model but trained with a different setup) should not have a better understanding of the underlying physical processes (i.e. flood generation).

LSTMs trained to local basins suffer from two problems:

1. They witness saturation effects (which is easily visible when plotting discharge time series and e.g. comparing the max peak the model is able to output during the training period vs validation period).

2. They are unable to model any process that did not happen in that particular basin during the training period.

The solution to both of these problems is training LSTMs on a multi-basin dataset, which afaik should be the one and only way how LSTMs should be applied for rainfall-runoff modeling. Looking at e.g. Frame et al. (2022), one can see how the regionally trained LSTM is able to predict unprecedented events in individual basins (e.g. being trained only on data with q < HQ5 and then tasked to predict > HQ100), which a locally trained model will never be able to do.

"*With the local models, since meteorological variables are the only inputs, we are able to focus on how they explain the temporal variation of discharges. In comparison, for a regional model that is supposed to capture both temporal and spatial variations in discharge peaks, it is challenging to distinguish how meteorological variables contribute to temporal variation in flooding within catchments from how they contribute to the spatial variation in flooding across catchments.*"

I think I disagree with everything you have said here. First, just because something is "more challenging", that doesn't mean in my opinion that the easier (but potentially wrong/biased) approach is justified. And in more detail, I also don't see the problem of looking at flood generating processes through integrated gradients, if static features are also model input arguments. Sure, the static arguments influence the model dynamics, but so does the fact that you have different model weights for every catchment. Just as an example, imagine you have a regionally calibrated model and two different basins A and B. Now we force the model with the same timeseries of meteorological features but using the different sets of static features, corresponding to basin A and basin B. Now imagine that in one basin, the model would produce a flood peak that, by looking at the integrated gradients of the meteorological features, is driven by the precipitation of the last few days, while in the other basin it produces a flood peak that is driven by temperature of the recent days and precipitation of some weeks ago. The difference here is certainly, because the static attributes are different. But does this have an influence on your analysis? I think not, and most likely this model has a much better understanding of the underlying processes. In your case, you would identify these two peaks as generated by different processes, which is most likely correct. So sure, static features have an influence on the model dynamics, but your analysis will still tell you which meteorological inputs are the most important for the peak flow. To be entirely honest, I fail to see what is the real challenge here.

"*...some studies have suggested that training a regional model for all catchments at once may be a better practice (e.g., Nearing et al., 2021).*"

I come back to this point, because it sounds like this is just a "suggestion". Let me tell you from experience from the operational side (not only at Google but various companies/agencies) but also from literature: I am not aware of anyone that applies LSTMs that is not using regionally trained LSTMs. In my view, there is really no discussion happening around whether you should use regionally or single basin calibrated models. Again, it is not *only* about better performance but also *why* the model can achieve this higher performance (see list above). I don't want to

step on someone's toes but from my point of view, the reason why people are sticking to single basin LSTMs is because it was hammered into their head for decades that single basin models are better at capturing local processes than regional models, which is true for conceptual models but it is not anymore for LSTM-based models.

2. Input feature selection

"*We agree that including more inputs is beneficial to uncovering patterns related to flood mechanisms that are likely to be overlooked. During our preliminary tests, we had run models with daily averaged sea level pressure, relative humidity, and radiation as additional inputs, which did indeed lead to more clusters in terms of feature importance patterns*"

Reading that you already did these experiments, which led to different results, but decided to not include them because they probably don't agree with the "known" patterns is in my opinion simply sad. I agree with the first reviewer on something he mentioned in a slightly different context, which is that you artificially limit yourself in this study. You have the potential to report new findings and even if you can't explain them, you could still present them and potentially start a new discussion in this area. Instead you limit yourself and the model to only check if your method identifies the same patterns as previous studies with different approaches.

"*The performance did not drop much by not including more variables*"

I think this is not important here, but as you said yourself, the clusters of flood generating processes changed. So which one is "the truth"?

I have nothing to add to this point and ultimately, it is the decision of the authors. I think it is just a missed opportunity and generally, I am not happy with the fact that all major reviewer comments are put into a new "Limitations and outlooks" section and that this suffices.

2. Data split

I think this is probably the most worrisome point to me. I think the applied data splitting (random in time) introduces a severe data leakage. You are not testing your method on unseen data, not even in these cases where you predict on a "test sample". The reason is that if you randomly select timesteps to be train/test data then three adjacent timesteps could e.g. be "train - test - train". Since each timestep is predicted from an input sequence of 180 days, e.g. the first train sample and the test sample only differ in a single time step of data. And without any doubt, the discharge data is highly auto-correlated. In Figure S1-S3, it is e.g. possible that for the models for which this time step appears to be in the test data (dashed lines) the previous and next timestep of that event is a training step. And in this case you can not at all argue that this is

independent test data. And that these plots show that the signal is the same for all 10-folds could be just because of this effect, because there is not really any point in your input time series that wasn't seen during training of any of these models.

"*Firstly, runoff data available in the GRDC dataset is not temporally complete in many catchments in Europe, with missing data sometimes occurring for several months or years irregularly. This complicates carrying out a unified temporal k-fold cross-validation across these catchments.*"

This is related to what I wrote above: Just because something is "more complicated", this doesn't mean you shouldn't do it. In fact, it isn't that hard to loop over basins and do time series splits per basin. How difficult it is to include different data splits for each basin in your training pipeline depends on your code. It is a built-in function in the open source library NeuralHydrology (https://github.com/neuralhydrology/neuralhydrology/ Disclaimer: I am one of the developers) and would work out of the box if you would use this for training your models.

"*We should emphasize that the model was used for statistical purposes instead of a prediction task, thus the split of the training dataset and the testing dataset is only to ensure the model has learned a generalizable relationship between variables.*"

I don't understand your point here. For any kind of statistical analysis with any model (data driven or not) you want to make the analysis on an independent dataset. The point is, that your test dataset is not independent of the training dataset as there is data leakage.

"*The generalizable relationship should hold not only for the testing dataset but also for the training dataset.*"

This is not necessarily true. First, in the extreme case you could overfit on the training data, meaning your model remembers every sample and thus is not generalizing at all. Second, have you ever looked at e.g. the NSE of an LSTM during the training period and compared this to the test period (with a non-random splitting). You will see that the LSTM achieves a much higher NSE during the training period and I would be more than cautious to draw any conclusions from this on the models generalization capabilities.

References:

Frame, J. M., Kratzert, F., Klotz, D., Gauch, M., Shelev, G., Gilon, O., Qualls, L. M., Gupta, H. V., and Nearing, G. S.: Deep learning rainfall–runoff predictions of extreme events, Hydrol. Earth Syst. Sci., 26, 3377–3392, https://doi.org/10.5194/hess-26-3377-2022, 2022.