REVIEWER 1: John Quilty

GENERAL COMMENTS

This paper introduces several variants of long short-term memory networks (LSTM) for the estimation of predictive uncertainty in rainfall-runoff modelling. The methods are explored using the CAMELS dataset (Catchment Attributes and MEteorology for Large-sample Studies) (Addor e t al ., 2017) as a means to provide a large-scale benchmark for the proposed methods as well as others that may be explored in the future. My takeaway is that both of these items are the main contributions of the paper, the latter, in my opinion, being the most relevant. In general, the paper is well-written and clear, the methodology is reasonable (however, some suggestions are included below), and the results seem promising. The main point I feel important to raise is that the authors adopt LSTM as the sole model and do not compare it against a plethora of other 'simpler' data-driven models that can provide estimates of the predictive uncertainty, likely with much less computational cost and onerous hyper-parameter tuning. Since the authors are looking to create a benchmark for comparing different predictive uncertainty estimation techniques in the context of rainfall-runoff modelling, it seems reasonable that very simple approaches be explored as a baseline before considering more complex methods. This critique is not meant to take anything away from the work that is done, since the reviewer understands LSTM may be a preferred method within this group (as is the case with process-based models within other groups), but more to seek justification for using more complicated (computationally intensive) models when it may suffice to use simpler ones that end up providing similar results, which seems relevant to consider from a benchmarking perspective. Aside form the above point, I consider most other comments below relatively minor but still important to address.If the authors can suitably address the comments noted below, I would be happy to recommend that the paper be published in Hydrology and Earth System Sciences.

Thank you for the thoughtful commentary. It was very helpful to us. We will respond to each of these issues in specific comments below, however it is worthwhile to address the larger point up front.

In general, we agree with the sentiment that complex approaches should be compared with a simpler reference. The lack thereof is exactly the reason why we proposed our baselines: We do not see any non-trivial method that would be easier than approaches that only require differentiable model-backbones able to provide the needed outputs. Especially if one considers that we gain fully distributional predictions from them (see Figure I of our answers), which are extremely rich representations that render them very flexible. That is, they can be used for almost all kinds of comparisons and benchmarking efforts in the future (e.g., we can analytically analyse the distribution as such, sample from them, derive point estimates such as the mean, the median or the maximum likelihood estimate, etc.; see also Figure 1 of our).

Further, just adding arbitrary methods to the paper just because we can do it from a computational standpoint would be antithetical with the idea of letting benchmarking become a community effort: Good benchmarking is not something that can be done in a responsible way in any single contribution, unless that paper itself is the outcome of a larger community effort (e.g., Best et al., 2015; Kratzert et al., 2019). Constructing new benchmarking approaches in a vacuum can be dangerous because they can easily become straw men (i.e., they might appear to be bad in isolation, but in reality the performance just reflects the choices or inabilities of the modellers). It can take a lot of knowledge, skill, and experience in any given method to use it correctly (even for “simple” methods). Because there is so much nuance to the current generation of ML and DL methods, because they take so much knowledge and skill to implement correctly, and because the hydrology community is just starting to build widespread expertise in these areas, it is all too easy to conduct flawed comparative studies in a vacuum. The way to guard against this is community benchmarking: We start with a set of self-contained baselines that are closed in themselves and openly share our framework, data, and methods Over time, we as a

community can then improve, replace, or add to them. Everyone runs the model or approach that they know best and results are compared at a community level.

As an example, in the specific comment 4 of reviewer 1 a particular method (QRF) is suggested to us. This somewhat undermines our intention of establishing DL-based baselines, since it is implied that practitioners can use it as a point of comparison. Imagine if we either naively or nefariously included the results from our QRF exploration (see our answer to specific comment #4) in the paper and showed that we could beat it. The end result would be that we would not have to push back against the request — and also our methods would look better by comparison. Further, the reviewer has not not used the method for the present setting (see specific comment #4), so he would have to trust our evaluation and make our road to publication easier. Despite this, we argue to not include QRF, since including it (and the corresponding outcomes) would be unrigorous . That is, the results might paint a wrong picture of the capabilities of the QRF, thus flawing the comparative benchmark and its potential adaptation.

This is why community benchmarking is so important. Community benchmarking is how a research community guards against this type of ad hoc and potentially misleading comparison. This study provides a UE benchmark on the CAMELS data, which is a large, publicly curated, open dataset that has been used frequently for model benchmarking. Our goal is to start a community benchmarking effort on this dataset for UE. In our view adding ad hoc benchmarks would be counterproductive.

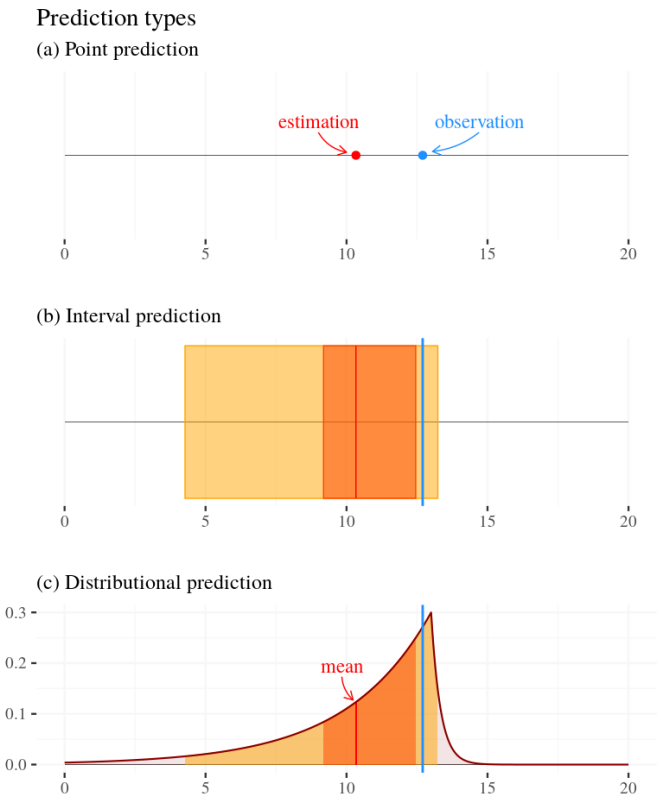

Figure 1. Different forms of predictions and their relation to each other. Note that the point prediction in plot (a) is the mean of the distribution prediction of plot (c).

Lastly we would like to comment on the presumption that our use of the LSTM perhaps is a subjective preference: While it might be true that some groups choose which model to use based on subjective preferences instead of objective criteria (e.g., Addor and Melsen, 2018), we do not view our choice as such. We use LSTMs for one reason: they are the best rainfall-runoff model that the hydrological sciences community has so far discovered by almost any metric. As mentioned above, the proposed UE approaches work with any differentiable model able to provide the required outputs. CMAL, UMAL, and GMM are just specific final layers in a DL network. MCD does not even require such a layer. If tomorrow some group developed a model that works better than the LSTM we would switch immediately. And, if said model would be based on the DL approach, say a transformer (Vasvani et al., 2017), the proposed UE baselines could simply be added as a layer on top. The LSTMs here are used only because they are currently the best rainfall-runoff model.

SPECIFIC COMMENTS

1. Terminology: throughout the paper the authors mention both 'prediction' and 'forecasting', which have different purposes and (potentially) different modelling setups. The authors should be consistent in their use of terminology throughout the manuscript. It appears the term 'prediction' is more appropriate considering their application.

Thank you for pointing this out. As mentioned in the original manuscript we tried to stick to the convention laid out by Beven and Young (2013) who suggest to use the word 'simulation' for a setting where the model uses a specific input for each time step but does not receive information about the observed outputs, and to use the term 'forecast' for settings where all information up to a given point in time is used to make a prediction. The term 'prediction' can be used for either situation. Thus, we should almost always use the term prediction or simulation here.

We used the term 'forecast' two times incorrectly in the paper - once in the abstract L.2 and once in the conclusions L.332. We will replace these occurrences with the term 'prediction' in the revised manuscript as proposed by the reviewer. This is unfortunate because we explicitly stated (and cited) our convention for this terminology in L.182 to L.186 of the original manuscript, and we appreciate the careful attention by the reviewer in catching this mistake.

2. Introduction: in L20-31 it would be good for the authors to acknowledge other established and recent methods for probabilistic prediction (and forecasting) in hydrology including: quantile regression-based neural networks (Cannon, 2018, 2011), copula-based approaches (Li et al., 2021; Liu et al., 2021), and (parametric/distributional) probabilistic decision tree methods (Başağaoğlu et al., 2021; Schlosser et al., 2019), the latter methods allow for the predictive distribution of the target variable to be easily estimated as part of the training procedure. Although I have yet to undertake such an analysis myself, I suspect the latter group of methods could estimate predictive uncertainty with much less computational expense than the LSTM variants adopted here.

This paper is not intended as a literature review of probabilistic prediction in hydrology in general. There are dozens of methods that the review could have listed here as examples of probabilistic prediction. The original manuscript already covers many established and recent methods that are directly relevant to our study. There are enough "probabilistic prediction" methods in the hydrology literature that if someone wanted to do a comprehensive review, this would end up being a stand-alone review paper. We do not see anything particularly special or relevant about the suggested references. If there is a particular reason to cite a paper that is directly relevant to this study, then we definitely want to cite it — for example, the paper suggested in the next comment.

Regarding the performance: In our eyes, LSTMs are computationally cheap to train and run. As of now, they can be trained and run using large data sets on any modern computer with a GPU in a few hours. For comparison, take the QRF proposed in specific comment 4. It required more resources (CPU) to run at a similar scale (details provided in our answer to specific comment 4).

3. Introduction: the authors may want to acknowledge the recent paper by Althoff et al. (2021) whom also used MCD with LSTMs for daily streamflow forecasting.

Thank you for this reference. It is indeed very relevant and we were not aware of it (as can be seen from the arxiv submissions, we finished our paper before Althof et al. (2021) was published and did not see it in the aftermath). We will include it in the revised manuscript.

4. Section 2 and 2.3: while I appreciate the authors have been exploring LSTM in many former studies and are evolving a research program around this method, from my own experience, these methods are extremely computationally intensive (due to it's recurrent formulation, gradient-based learning, need for careful hyper-parameter tuning, etc.), and thus tend to be a 'turn-off' for those who do not have adequate computing resources to explore such methods. It would seem beneficial for the reader to have a 'simpler' non-LSTM baseline to compare against the proposed LSTM variants. If the results of the LSTM variants significantly outperform these

simpler methods, then it may serve as a motivation for other researchers to devote more time and resources to incorporating LSTM into their research endeavours. My feeling is that a simple benchmark method could include one of the many variants of quantile regression forests (see for example, https://scikit-garden.github.io/examples/QuantileRegressionForests/). This would not seem out of scope as the authors mention in Section 2 that their work is devoted to data-driven methods, of which LSTM are only a small (but relevant) fraction.

The answer to this comment is closely linked to our answer to the general comments of reviewer 1. While the answer there gives our arguments with regard to our vision (and is therefore more important), this answer can be seen as a technical addendum.

First, we would like to emphasize that we use the LSTM because it is the best model that we (or any other hydrology group) has found for rainfall-runoff estimation. This is the only reason. We care about providing information to research and operational groups who want to do the best job possible. Within this "big-data regime" our team has examined many different types of ML and DL models. For example, we have tried transformers, MLPs, boosted regressions (XGBoost), experimented with neural ODEs, and developed custom physics-informed neural network architectures (Hoedt et al., 2020). We will continue to try other approaches whenever possible. So far LSTMs simply work the best for the task of rainfall-runoff modeling.

Second, the LSTM does not constitute an uncertainty estimation approach, as such. In this paper it is used as a base model and the uncertainty approaches are independent of the LSTM in the sense that they could be applied to any type of ML model (they are just a layer in the deep learning network). If tomorrow we discovered that — for example — transformers were better than LSTMs for rainfall-runoff modeling, we would simply apply the UMAL/CMAL/GMM methods tested here to transformers.

Third, and irrespective of all of what we said above, we tested the quantile regression forest (QRF) that the reviewer suggested: In summary, the QRF performed worse with respect to the particular modeling problem. We did not anticipate these problems before starting with QRFs — it seems like a reasonable method — but after

running the method and understanding why it fails, it seems to us that it is not fit for this purpose. We have three things to say about this:

1) QRF doesn't work for regional streamflow modeling. QRF gives a median NSE of ~0.24 (Figure 2), whereas the lowest median NSE value in our work is from GMM at ~0.74. Because the QRF loss function is based on counting bins (it is the MSE of the predicted vs. actual number of observations that fall in each quantile bin), it can learn a perfect probability plot (Figure 3) without learning any dynamics of the system. In our results it learns some dynamics, but not enough to simulate realistic (probabilistic) hydrographs (Figure 4). The result is that while the average QQ plot looks good (because it is directly optimized for this metric), it has large biases from basin to basin (Figure 5). QRF is simply not designed for this type of modeling problem (Meinshausen, 2006).
2) The proposed QRF is computationally expensive (much more expensive than the LSTM). Used naively, it would take a long time to train the QRF on the same data for which we can train an LSTM in under 3 hours on a laptop with a single GPU, even when QRF fitting is fully parallelized on a node with 40 cores. It is possible that the specific implementation that the reviewer linked is poorly coded (there might be better ways to implement the QRF algorithm), but we are not experts in this method and we had to make several adaptations (in terms of hyperparameters) to get a performant variant to work.
3) This leads us to our third point, which is that using the QRF is not simple. There are many hyperparameters that must be set based on a combination of intuition, expert knowledge, and formal hyperparameter searching. Informative hyperparameter tuning would require millions of CPU-hours for QRF (as opposed to only hundreds of GPU hours for LSTM hypertuning). Additionally, we are not experts in QRF, so if we were to include any QRF results in this or any other paper, someone who is an expert could likely find criticism of our implementation. This is why community benchmarking is so important - groups who are actually invested in a method need to implement that method in a reproducible way on open community datasets so that results are directly comparable.

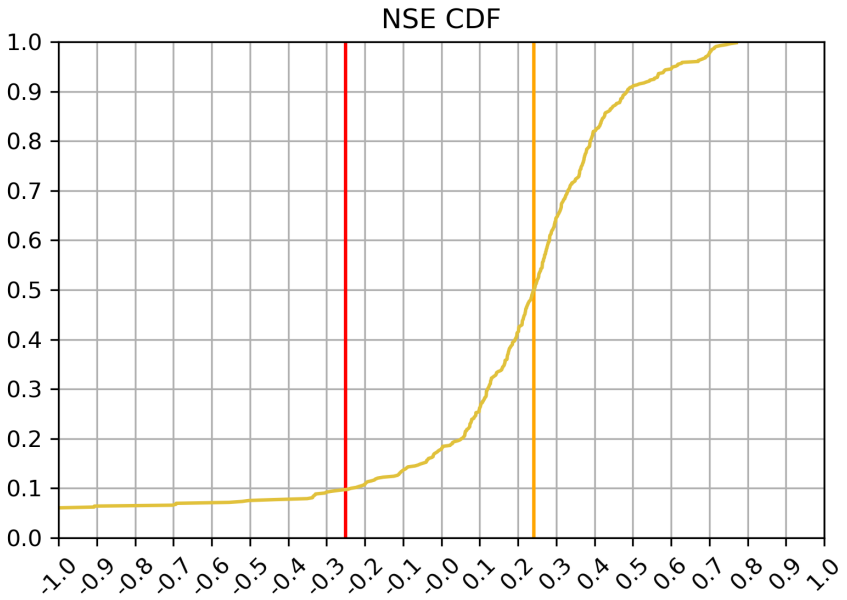

Figure 2. Empirical cumulative distribution function of the Nash-Sutcliffe Efficiencies (NSE) obtained by using the median of the Quantile Random Forest as a predictor. The x-axis is limited to -1 to 1, the mean NSE is depicted with a red vertical line, and the median NSE with an orange vertical line.

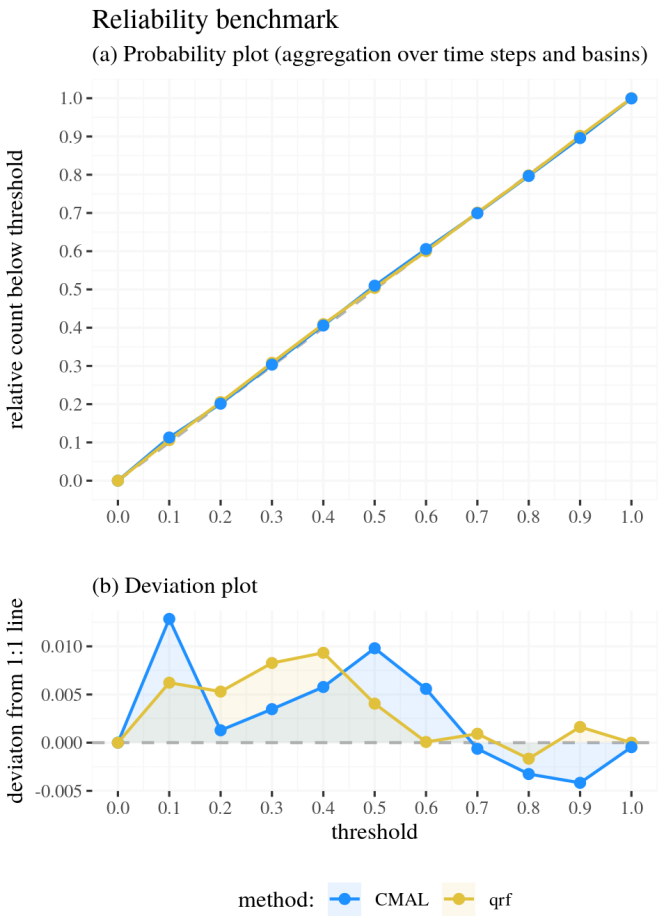

Figure 3. Reliability benchmark for the Quantile Random Forest (qrf) and the Countable Mixture of Asymmetric Laplacians (CMAL) in form of the probability plot and the deviation plot in the style of Figure 8 of the original manuscript.

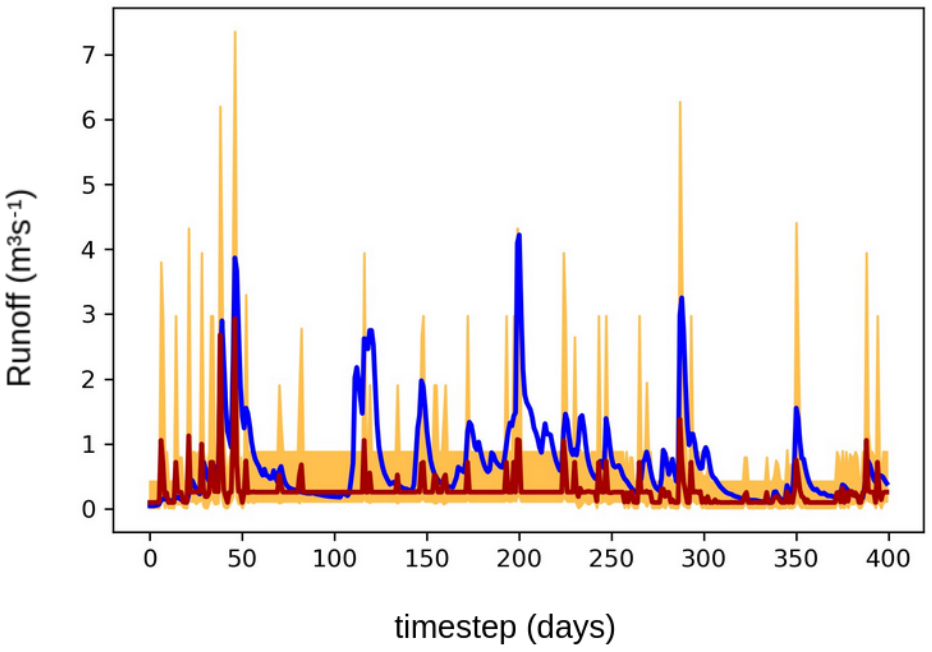

Figure 4. Working example hydrograph for a random basin. The observed streamflow is shown in blue, the median in red and the interquartile-range (the distance between the 25th and 75th percentiles) in orange.

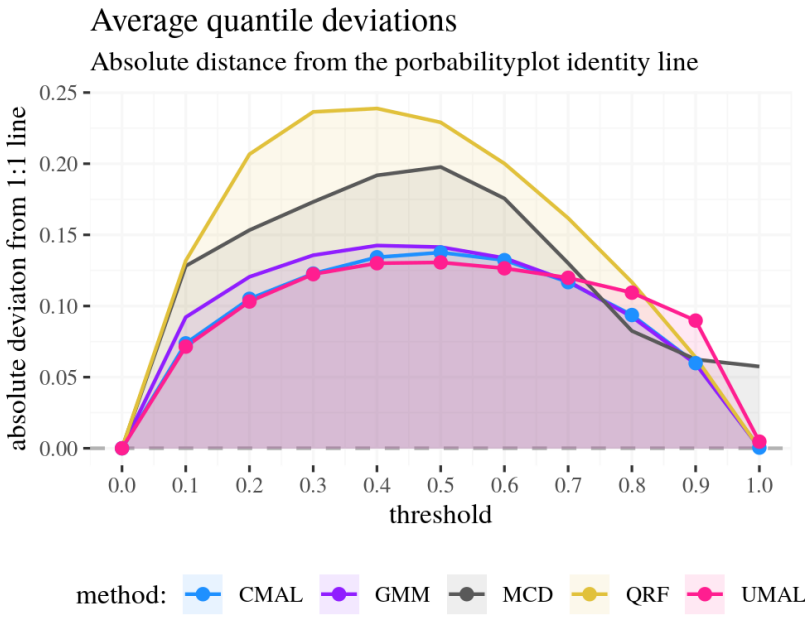

Figure 5. Demonstration of using the absolute deviations from the 1:1 line as an ad-hoc diagnostic, which does not allow trading-off performances between basins.

5. L177-180: This is somewhat confusing. If I understand correctly (based on the Althoff et al. (2021) paper), dropout is used during training at each iteration but it does not create a separate model at each iteration, only a 'thinned' network. However, performing dropout during testing (or model implementation, evaluation, or whatever other term you wish to use), each time you make a prediction you simply turn on/off nodes according to the pre-specified probabilities used during training and you repeat this as many times as you desire, creating a number of 'sister' predictions. Again, the model does not change, you simply 'thin' the network each time you create a 'sister' prediction. If this is how MCD was used in the experiments described in this paper, it is not apparent and would be helpful to clarify.

We are not sure if we can follow why the reviewer finds these lines confusing.

The reviewer's interpretation of the dropout mechanism is correct, however this is exactly what is described in lines 177-180. The reviewer also correctly assesses that a thinned network is not a separate model in the sense that all thinned networks use the same tensor network. Thus, the reviewer might be objecting to our use of the term "sub-model" in this sentence, however this is the term used in the original dropout paper (Srivastava et al., 2014) to describe the concept: *"The central idea of dropout is to take a large model that overfits easily and repeatedly sample and train smaller sub-models from it."* (Srivastava et al., 2014). In our eyes it is very intuitive to think of dropout as a way of building up an implicit ensemble (as a matter of fact, a very particular one as shown in Gal and Ghahramani, 2016).

Alternatively, the reviewer's confusion might arise due to the term 'sister prediction', which seems to be used incorrectly by Althoff et al. (2021). This term was not used in the original papers on dropout (Srivastava et al., 2014) or Monte Carlo Dropout (Gal and Ghahramani, 2016). The oldest source we could find was Liu et al. (2017): *"Sister forecasts are predictions generated from the same family of models, or sister models. While sister models maintain a similar structure, each of them is built based on different variable selection process, such as different lengths of calibration window and different group analysis settings."* If this is the working definition of "sister-predictions" then the thinned models resulting from dropout would not be sister models and the resulting forecasts are not sister forecasts.

We hope that this helps. Either way, the description of dropout in the text of our paper is correct and uses language that is in agreement with primary sources.

6. L193-4: why not use a regularized squared loss function? Is it not a standard practice to perform L2 (and potentially L1) regularization to improve LSTM performance and reduce overfitting? Was this considered? If not, why?

There are many regularization techniques for neural networks. For example, we use dropout as a regularization. This is quite common. L1 and L2 regularization can yield good results in specific contexts, but they are not as trivially applicable as one might think — for example, we did indeed try L1 and L2 regularizations in some previous experiments, but did not obtain good results. For the use of L2 regularization specifically, it is now often implicitly used/approximated in the form of weight-decay.

7. L195-6: what dataset partition was used to select optimal hyper-parameters?

As is customary, we used the training- and validation-sets as defined in the appendix. We will mention this explicitly, since— as the reviewer rightfully points out — we cannot assume that readers would know.

8. Section 2.4.3: the authors should formally define first and second order uncertainties.

We will answer this together with comment 12.

9. Table 3: it's not clear how the 'obs' data is to be used to 'contextualize' the results from the different models. More detail should be provided (perhaps with an example in the relevant section).

We are not sure what the confusion is. Statistics from the observation sampling distribution are given as a reference for statistics of model error distributions. We are not sure what details could be provided or how there could be confusion about this. Basically, variances and quantile widths mean relatively little on their own

without a point of reference, and comparing to the observation distribution shows improvement due to using the UE approaches.

10. L286-289: it would seem like a good idea for the authors to investigate how best to use the CMAL and UMAL variants for improving predictive performance at low and high (tail) flows (from the point prediction point of view), as this tends to be a major impetus for creating models with uncertainty assessment capabilities. In other words, what's the point of going through the trouble of designing these more sophisticated methods if they cannot outperform the base approach (LSTMp) when assessed on highly relevant metrics. Please don't get me wrong, I am not trying to downplay the very interesting work done by the authors, I am simply trying to help them more fully explore the merits of their work and better 'sell' their approach to a 'skeptic'.

There is likely some sort of confusion here from the reviewer. All of the reported point metrics are relatively good in comparison (see e.g. Kratzert et al., 2019). The low and high-flow metrics are however measuring a form of bias that assumes a symmetric error distribution. And, the point of these sentences is that the mean of the distributional predictions is biased in low and high flow regimes, since we can expect that the underlying distributions are indeed asymmetric (e.g. there are now flows below 0). This is a necessity: There is an inherent (not incidental) tradeoff between predicting asymmetric distributions (hydrological uncertainty is well-known to be asymmetric) and using the mean of an asymmetric prediction as a point-estimate (compare Figure I of our answers). This is a fundamental attribute of probabilities that we just wanted to point out explicitly. It's not about a difference between models (no model can fix this problem), it is a fundamental artifact of probabilistic prediction. If necessary, we can revise this sentence in the manuscript, but it is not a matter of a poorly performing model or bad metric.

11. Section 3.2.2 is very interesting!

Thank you. This is much appreciated.

12. Section 3.2.3: once first and second order uncertainties are formally defined (see comment 8), this section should give a good description of what is shown in Figure

11 but it is unclear what message the authors expect the reader to take-away from this figure. What's the relevance of this figure and why should the reader 'care' about it?

We agree with the critique.

Formally, if a variable y is described by a family of distributions $y = f_0(y; \theta_1)$ that are functions of possibly multidimensional parameters θ_1, we call it first order uncertainty. If the potential distribution of said parameters are estimated — that is, if the parameters are "*stochasticized*" — as in $\theta_1 = f_1(\theta_1; \theta_2)$ one calls it second order uncertainty. As usual, higher orders can be derived recursively so that $\theta_n = f_n(\theta_n; \theta_{n+1})$. First order uncertainty is related to aleatoric uncertainty, and second- or higher-order uncertainties to epistemic uncertainty (we are however in a very restricted setting here and the approaches do not necessarily disentangle the different kinds of uncertainties). Similarly, the MDNs estimate first-order uncertainties, while MCD estimates second-order uncertainty.

That said, we do not believe that a formalization is the right way to go at this point of the paper. The above formalism will be familiar in one way or another to many. We think the problem here lies in our unclear usage of language. That is, a more thoughtful description is warranted. We will thus expand the section in the following way:

> "In this experiment we want to demonstrate an avenue for studying higher-order uncertainties with CMAL. Intuitively, the distributional predictions are estimations themselves and thus subject to uncertainty. And, since the distributional predictions do already provide estimates for the prediction uncertainty we can think about the uncertainty regarding parameters and weights of the components as a second-order uncertainty. In theory even higher-order uncertainties can be thought of. Here, as already described in the method-section we use MCD on top of the CMAL approach to "stochasticize" the weights and parameters and expose the uncertainty of the estimations. Figure 11 illustrates the procedure: The upper part shows a

hydrograph with the 25%–75% quantiles and 5%–95% quantiles from CMAL. This is the main prediction. The lower plots show kernel density estimates for particular points of the hydrograph (marked in the upper part with black ovals labeled ‘a’, ‘b’ and ‘c’, and shown in red in the lower subplots). These three specific points represent different portions of the hydrograph with different predicted distributional shapes and are thus well suited for showcasing the technique. These kernel densities (in red) are superimposed with 25 sampled estimations derived after applying MCD on top of the CMAL model (shown in lighter tones behind the first order estimate). These densities are the MCD-perturbed estimations and thus a gauge for how second order uncertainty influences the distributional predictions.”

13. Section 3.3: my understanding is that this is the time needed to make predictions with a trained model. What is the training time for the different models? Can the authors provide an example calculation for the overall run-time in Appendix A (or at the very least in their reply, it’s not clear how the 365 and 174 days were calculated)?
This is correct and the reviewer did indeed spot an error here. Training is much faster, since no sampling is necessary! If we have one batch-size of 256, 531 basins, 10 years, with 365 days each, we have *256 x 531 x 10 x 365 = 1938150* data-points. Here, the batch-size tells us how many can be processed in parallel. We need approximately 7 minutes per epoch — i.e. to make a prediction for each data-point — since each batch takes ~0.055 seconds to compute. For, say 30 epochs, this would yield a model training-phase of 3.5 hours. The different hyper-parameter runs can of course also be parallelized.

When using MCD for sampling, the samples are generated by repeatedly re-executing the model. In the example, we originally wanted to take 75,000 samples for the 531 basins over 10 years. This means that we would need to generate *75,000 x 531 x 10 x 365* points. For illustrative purposes we assumed a batch-size of 256, as above. For practical purposes one could of course use much larger batch-sizes (whatever fits on the GPU) at no additional cost. To get the number of days from this we compute:

(75,000 x 531 x 10 x 365 x 0.055)//(256 x 60 x 60 x 24) ≈ 361.46 days

That is, approximately 360 days (providing a simple estimate, and accounting for errors in the computation and numerical imprecisions etc.). The 174 are obtained by replacing 0.055 with 0.026 in the above computation and rounding up.

In retrospect we should have stuck with the original 7500 (which would have yielded 36.1 and 17.4 days, respectively) that we used throughout the manuscript. We will correct this for the revised version of the manuscript. and larger batch-sizes are possible in practice.

14. Section 4: after L335 the authors may wish to very briefly summarize the adopted models and datasets used in the study before continuing with L336 onwards. This should help the interested reader with a short 'time-budget', who may only jump from the abstract to the conclusion, get a decent idea of the methods and dataset involved (the dataset being one of the key strengths of this paper).

This is a very good proposal. We will do so for the revised manuscript version.

15. Each equation in the appendices (B1, B2, etc.) should be properly cited (the rule is to cite all equations that are not developed by the authors in the paper).

We agree that all equations should be properly cited. And, we believe that we did so: The GMM mechanism stems from Bishop (1999), the UMAL from Brando et al. (2020) and MCD from Gal and Ghahramani (2016). All of these are referenced. CMAL was introduced by us, so no further references are necessary. Further, we adapted the entire notation to put the equations into the present context and make it easier to see where the approaches are similar or divergent. Also here, no further references are necessary.

TECHNICAL CORRECTIONS

[...]

We will adapt all proposed technical corrections. Thank you for pointing these out.

References

- Addor, N., & Melsen, L. A. (2019). Legacy, rather than adequacy, drives the selection of hydrological models. *Water resources research*, *55*(1), 378-390.
- Althoff, D., Rodrigues, L. N., & Bazame, H. C. (2021). Uncertainty quantification for hydrological models based on neural networks: the dropout ensemble. *Stochastic Environmental Research and Risk Assessment*, *35*(5), 1051-1067.
- Başağaoğlu, H., Chakraborty, D., & Winterle, J. (2021). Reliable Evapotranspiration Predictions with a Probabilistic Machine Learning Framework. *Water*, *13*(4), 557.
- Beven, K., & Young, P. (2013). A guide to good practice in modeling semantics for authors and referees. *Water Resources Research*, *49*(8), 5092-5098.
- Cannon, A. J. (2018). Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes. *Stochastic environmental research and risk assessment*, *32*(11), 3207-3225.
- Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning* (pp. 1050-1059). PMLR.
- Gauch, M., Mai, J., & Lin, J. (2021). The proper care and feeding of CAMELS: How limited training data affects streamflow prediction. *Environmental Modelling & Software*, *135*, 104926.
- Li, H., Huang, G., Li, Y., Sun, J., & Gao, P. (2021). A C-Vine Copula-Based Quantile Regression Method for Streamflow Forecasting in Xiangxi River Basin, China. *Sustainability*, *13*(9), 4627.
- Liu, B., Nowotarski, J., Hong, T., & Weron, R. (2015). Probabilistic load forecasting via quantile regression averaging on sister forecasts. *IEEE Transactions on Smart Grid*, *8*(2), 730-737.
- Liu, Z., Cheng, L., Lin, K., & Cai, H. (2021). A hybrid bayesian vine model for water level prediction. *Environmental Modelling & Software*, *142*, 105075.
- Meinshausen, N., & Ridgeway, G. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*(6).

- Papacharalampous, G., Tyralis, H., Langousis, A., Jayawardena, A. W., Sivakumar, B., Mamassis, N., ... & Koutsoyiannis, D. (2019). Probabilistic hydrological post-processing at scale: Why and how to apply machine-learning quantile regression algorithms. *Water*, *11*(10), 2126.
- Renard, B., Kavetski, D., Kuczera, G., Thyer, M., & Franks, S. W. (2010). Understanding predictive uncertainty in hydrologic modeling: The challenge of identifying input and structural errors. *Water Resources Research*, *46*(5).
- Schlosser, L., Hothorn, T., Stauffer, R., & Zeileis, A. (2019). Distributional regression forests for probabilistic precipitation forecasting in complex terrain. *Annals of Applied Statistics*, *13*(3), 1564-1589.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).
- Website: Tim Wiliams 2018 on stackoverflow: https://stackoverflow.com/questions/51483951/quantile-random-forests-from-scikit-garden-very-slow-at-making-predictions, last accessed at 22.June.2021.