Comments/Text of **Anonymous Referee 1 (AR1)** posted in blue, and our answers in black with old passages in red and new passages in green.

This very interesting paper of Kratzert, et al. compares the quality of the predictions of various hydrological models with three variants of the Long Short-Term Memory(LSTM) deep learning network. One of these variants, the novel EA-LSTM, is trained using meteorological data and catchment similarities as an additional input and is analysed in detail highlighting the superiority of such a network. In general the paper is very well written and it is worth to be published after some minor changes. Some comments:

1. Maybe you could explain better the differences of the analysis of the single model and the ensemble mean approach. On page 13, lines 317-320 you write: "To assess statistical significance for single models, the mean basin performance (e.g. mean NSE per basin and across all seeds) between two different model settings was compared between different model configurations." What's the difference between model settings and configuration? If I understood it correctly the difference in the verification of the single models and the ensemble mean is: Single model: From 8 ensemble model runs, you get 8 different predictions and you calculate the verification measures (e.g. NS values) for each of it and take the average (+/- Std? in Table 2); whereas in the Ensemble mean approach, for example this measure is calculated taking the mean of the 8 predictions?

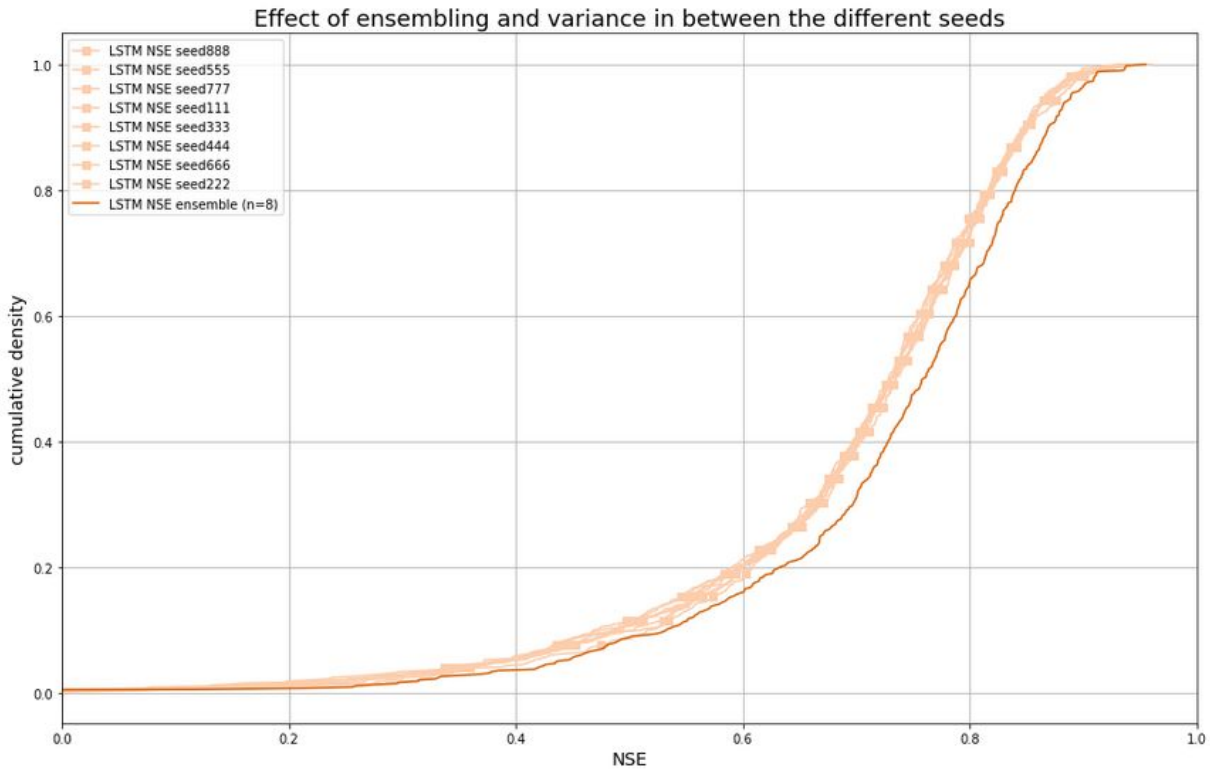
Thank you. We will rewrite this section of the description of methods (lines 317ff) to more clearly describe the ensemble approach and how the statistics of the single model are calculated.

Old passage:
To assess statistical significance for single models, the mean basin performance (e.g. mean NSE per basin and across all seeds) between two different model settings was compared between different model configurations. To assess statistical significance for ensemble means, the mean basin performance of the ensemble mean was compared between different model configuration.

New passage:
To assess statistical significance for single models, we first calculated the mean basin performance, i.e. the mean NSE per basin across the 8 repetitions. The so derived mean basin performance was then used for the test of significance. To assess statistical significance for ensemble means, the ensemble prediction (i.e. the mean discharge prediction of the 8 model repetitions) was used to compare between different model approaches.

2. For clarity reasons I would not include the single model outcome in Figure 3, because this a random outcome and would look different for each ensemble run.

We don’t agree with the reviewer on this point because we think it helps to underscore the potential of ensembling. As shown in Table 2 and Table 3 in the manuscript and the figure below, the overall CDF of the single models does have little variation between random seeds, especially in comparison to the benefit of ensembling. Therefore, we would like to keep these curves in the figure, since it helps to visualize the ensembling benefit and show by how much the CDF can be improved.



3. Nice to have the significance reported, which is most often not shown. Although the precision of these p-values is extremely high and the differences are probably rather neglectable caused by noise.

We agree with the reviewer that reporting significance measures is important.

4. Regarding the modified NSE. Wouldn’t it be easier to normalize the streamflow data (e.g. using the BoxCox transformation)? So you don’t have to event a new measureand adding a constant in order to achieve stable results.

The loss function we use isn’t really new. It’s just the basin-average NSE. The reason this is a little unusual in Hydrology is because we rarely (if ever) calibrate a single model to multiple basins. But this is standard practice in machine learning - where the overall loss function is the average loss over many samples. The only alternative is a single loss function calculated over concatenated data from multiple samples, which doesn’t work well (or at least not as

straightforward) for stochastic gradient descent, which randomizes and sub-samples the training samples.

Moreover, the reason we did not transform the data before training is because this affects model performance (besides normalizing to zero mean, unit variance). For example, an exponential-type transform like Box-Cox will generally under-emphasize peak flows (if the box-cox exponent is <1). We did try calibrating to log-transformed data, in an effort to normalize the streamflow data, but this does not work as well as using the natural streamflow data. The goal is to train to the non-transformed target data, but use a loss function that does not overemphasize any particular training sample (i.e., any particular or individual basin does not have out-sized influence on the training procedure).

5. Looking at the results, I would conclude that the EA-LSTM is very interesting fort his analysis, but for practical applications the LSTM with the coupled meteo data and catchment attributes is even more efficient and is less complex. That's why I would like to see the results of this model also in Figure 4, 5 and Table 3.

We will modify Figure 4, Figure 5 and Table 3 to include the results of the standard LSTM.

New passage added to the beginning of Sect. 3.2:
In this section, we concentrate on benchmarking the EA-LSTM, however for the sake of completeness, we added the results of the LSTM with static inputs to all figures and tables.

6. Are the catchment attributes kept static for all days of the year? For example the monthly mean of leaf area index could be easily varied depending on the month of the year?

Yes, the catchment attributes are static in this study. We are currently working on making these dynamic, including vegetation and climate indexes, soil moisture, snow cover from remote sensing, etc. This is however not a trivial extension and we do believe that the idea is worth being studied separately. Furthermore, for the sake of repeatability, we wanted to stick entirely to the CAMELS data set, which only includes static catchment attributes. Right now, in this paper, we are using long-term catchment attributes as indicators of differences between catchments (regional heterogeneity among catchments), not for assessing nonstationary catchment behaviors.

7. I would suggest to delete the UMAP analysis, since the method is not explained and the results are a bit confusing.

On this point we don't agree with the reviewer and would like to keep this section. We see this as an interesting addition to the introduction of the new EA-LSTM and the benchmarking results. Specifically, we are using this analysis to illustrate the fact that the embedding layer (our static LSTM input gate) can 'learn' about catchment diversity in a physically meaningful way. This is a (fairly simple) form of explainable AI, and one of the goals of this paper is to work toward that

larger objective. The analysis of the embedding layer is important as an example of this larger purpose, and the UMAP analysis in particular is necessary for a reduced-dimension (i.e., graphical) analysis.

Although our paper is mainly intended as a modeling paper and we see the introduction of the EA-LSTM and benchmarking against various hydrological models as our main contributions, we think keeping Section 3.4 has the following benefits:

- It provides at least some feeling about what happened in the embedding layer of the input gate in the EA-LSTM (namely grouping of basins that match our expert knowledge).
- As such, it helps in our opinion to gain trust into LSTMs which are widely considered as black-box model.
- It shows possible analysis that are possible with the EA-LSTM in general and potentially open the door for many follow-up studies in the future. Such studies could either concentrate more on the interpretability of LSTM-based models or try to extract new hydrological understanding from the learned groupings.

We will extend the manuscript to include a short description of the UMAP method, however, would avoid a lengthy discussion on the method and point the interested reader to the official UMAP publication. In our manuscript it is simply take as (state-of-the-art) dimension reduction technique.

New passage describing UMAP:
Finally, we reduced the dimension of the input gate embedding layer (from R^256 to R^2) so as to be able to visualize dominant features in the input embedding. To do this we use a dimension reduction algorithm, called UMAP (McInnes et al., 2018) for "Uniform Manifold Approximation and Projection for Dimension Reduction". UMAP is based on neighbour graphs (while e.g., principle component analysis is based on matrix factorization), and it uses ideas from topological data analysis and manifold learning techniques to guarantee that information from the high dimensional space is preserved in the reduced space. For further details we refer the reader to the original publication by McInnes et al. (2018).

Comments/Text of **Hoshin Gupta (Reviewer 2)** posted in blue, and our answers in black with old passages in red and new passages in green.

([1-11] present a thoughtful summary of our manuscript)

[12] I believe that this paper represents a very significant contribution to the Earth System literature related to the development of Dynamical Environmental Systems Models (DESMs). I have alluded to some of the problems associated with the conventional CM approach in paragraphs [2-6] above. In this regard, there has been increasing community interest in the use of both "large sample" data sets and the use of "model-structural-correction-via-data-assimilation" (learning from data) to extract better understanding about the structure and functioning of hydrological systems, such as catchments.

[13] This paper bridges the challenges of learning from large sample data sets and learning how catchments structures/behaviors can differ at local to regional scales in a very meaningful way. While not addressing the problem of prediction in ungaged basins directly, the ability of the EA-LSTM to learn from and characterize differences in catchment functioning encoded in catchment attribute data is highly significant, and it would seem that a natural next step would be for the authors to demonstrate that potential by running experiments that seek to demonstrate that predictive ability learned from gaged locations can be transferred to ungaged locations. I look forward to reading more about this in the future.

We do have a short paper on this topic in WRR, that this reviewer is also currently reviewing. That paper, however, does not explore how the catchment-aware embedding presented here as an adaptation of the LSTM architecture helps in the PUB setting. This is for future work.

[14] As such, I have only a few suggestions to offer the authors. The first is that the current title "Benchmarking a Catchment-Aware Long Short-Term Memory Network (LSTM) for Large-Scale Hydrological Modeling" presents a rather technical front to what is arguably (in my opinion) a much more significant piece of work. I therefore offer up the possibility for the authors to consider that the introduction and discussion/conclusions sections be somewhat revamped/broadened to reflect the perspectives offered in my above summary of the paper. As indicated, I do think this paper is really more about the interesting challenges of learning and characterizing (via dynamical systems models) the "behavior and functioning" of hydrological systems at the catchment scale in such a manner that both universal (fundamentally hydrological) principles, and local-to-regional scale uniquenesses of such systems can be learned by accessing the patterns of information encoded in large sample data sets (Gupta et al 2014). In this regard the title could also then be generalized to reflect the nature of the conversation about "Learning Universal, Regional and Local Hydrological Behaviors via Machine-Learning applied to Large Sample data Sets". Or this more general discussion could be saved for a future publication.

Thank you for the suggestion about broadening the scope implied by the title. We will take the suggestion to change the title and update the discussion accordingly. However we don't feel confident enough to state that we already have a "universal" model, but rather that this work is a step in that direction. Thus, we would change the title to "*Towards Learning Universal, Regional, and Local Hydrological Behaviors via Machine-Learning Applied to Large-Sample Datasets*". We Furthermore adapted the first paragraphs of the discussion as follows:

New passage (beginning of discussion section):
The EA-LSTM is an example of what Razavi and Coulibaly (2013) called a model-independent method for regional modeling. We cited Besaw et al. (2010) as an earlier example this type of approach, since they used classical feed-forward neural networks. In our case, the EA-LSTM achieved state-of-the-art results, outperforming multiple locally- and regionally-calibrated benchmark models. These benchmarking results are arguably a pivotal part of this paper. The results of the experiments described above demonstrate that a single 'universal' deep learning model can learn both regionally-consistent and location-specific hydrologic behaviors. The innovation in this study – besides benchmarking the LSTM family of rainfall-runoff models – was to add a static embedding layer in the form of our EA-LSTM. This model offered similar performance as compared with a conventional LSTM (Sect. 3.1) but offers a level of interpretabiltiy about how the model learns to differentiate aspects of complex catchment-specific behaviors (Sect. 3.3 and Sect. 3.4). In a certain sense, this is similar to the aforementioned MPR approach, which links its model parameters to the given spatial characteristics (in a non-linear way, by using transfer functions), but has a fixed model structure to work with. In comparison, our EA-LSTM links catchment characteristics to the dynamics of specific sites and learns the overall model from the combined data of all catchments. Again, the critical take-away, in our opinion, is that the EA-LSTM learns a single model from large catchment data sets in a way that explicitly incorporates local (catchment) similarities and differences

[15] The second is that while the basin-average NSE* loss function does seem to serve the immediate needs of this study, I think that the ML-approach (and more generally hydrological learning from catchment data sets) can benefit from a more thoughtful approach to the problem of model performance metrics. In particular, the use of the observed output data "mean" as a benchmark for constructing the NSE itself, and the use of the output data variance to "normalize" across catchments to obtain somewhat comparable metric values to be averaged (or otherwise summarized in some statistical manner) seems, to me, problematic. In this regard, I think an Information Theoretic approach might ultimately prove to be more meaningful. I point out that the value of the metric, when used as the basis for assessing across different catchment locations, would be much enhanced if it somehow recognized the relative differences in complexity/difficulty associated with modeling the dynamical input-state-output behaviors at different locations (due to climatic, geological, and other factors). As discussed by Schaefli and Gupta (2007), the problem is at least partly one of appropriate benchmarking in order to make metric values meaningfully comparable. Some types of catchments (such as humid ones perhaps) are relatively easy to model to the level of obtaining high performance (e.g. NSE)

values, while others (such as arid ones perhaps) are much more difficult to model … potentially requiring more complex model structures, more data, and perhaps better data quality. Since the challenge here is learning hydrological principles from the data, and some catchment systems are easier to characterize using simpler model structures, it would seem prudent to figure out how to account for this knowledge in the designs of our learning systems, which includes the metrics used as the filter through which information is being extracted.

We absolutely agree that it will be critical, going forward, to understand carefully and in detail how different loss functions affect the training of deep learning Hydrology models. We've done some work with this - trying to emphasize peak and low flows, working with probabilistic loss functions, etc. None of that work is mature enough to publish at this point. This has been a much bigger challenge that we perhaps originally expected, and we appreciate the reviewer's advice. Hopefully we will have something more meaningful or helpful to say about this in a future publication.

[16] Finally, I think that the aforementioned issue may also relate to the fact that certain catchment attributes tend to be dominant indicators of differences in catchment behaviors, while others seem to show "lower importance" (sensitivity). It is been well known that "climate" (and one would reasonably expect also "topography") is the dominant indicator of catchment similarity, but this does not really help us to understand what structural differences in catchments drive differences in their behaviors. The finding that soil and vegetation characteristics are low on the "importance" list is interesting, as it suggests that the existing catchment attributes being used may not be sufficiently informative about catchment-scale soil and vegetation contributions to hydrological behaviors. So, is it a problem of poorly encoded soils and vegetation information at the catchment scale, or is really the case that such soils and vegetation do not play as big a role in hydrological behavior as we might expect? It would be interesting to consider how this issue could be better investigated using the ML approach.

First we would like to clarify that we do not say that soil and vegetation indices are not "important" but just that climatic and topographic attributes are *more* important. As the reviewer mentions, this follows hydrological literature and also the intuition of the reviewer. In our case, this could potentially result in two basins with similar climatic and topographic attributes that are distinguished primarily by their soil/vegetation properties. However, in the larger context it is the set of climate and topographic attributes that "separates" the most basins and thus the larger sensitivity/importance.

Regarding the question about what we might be able to learn from these results for hydrological modeling. This seems in-line with the well-known idea that the first-order trends in most models are Budyko-type effects (i.e., climate related). This is not especially new, but also encouraging that the LSTM behaves as expected. One thing we might take away from this is due to the comparison with MPR regionalization. MPR uses topographic, geologic, soil and land use attributes as inputs. We show that our regionalization approach outperforms the two MPR calibrated models (VIC and mHM). This could indicate that (the conventional use of) MPR might

be improved by including climatic attributes in the regionalization scheme.. I guess the question is about the extent to which it is meaningful for a model to ‘react’ directly to climate indexes rather than just to meteorological forcings. The regressions in typical regionalization strategies could use climate indexes as regressors or inputs.

Another point regarding the relative unimportance of geological and vegetation features indicated in our sensitivity analysis is that probably catchment averaged soil properties, as well as vegetation indices contain too much noise compared to the relatively noise-free climatic and topographic attributes. This is what is probably meant by the reviewer with “poorly encoded” information of those features, in which case we would agree. We don’t, however, prove this in the paper - all we show in the paper is that there is enough information in the indexes to at least help with catchment differentiation, and that traditional approaches do not utilize all of this information.

Comments/Text of **Anonymous Referee 3 (AR3)** posted in blue, and our answers in black with old passages in red and new passages in green.

Benchmarking LSTM

Overall, this paper stands at the forefront of hydrology. There are three aspects of the paper that I like. First, this work show state-of-the-art performance in terms of large-scale streamflow prediction accuracy. This would serve to push hydrologic science forward. Second, the authors implemented a novel LSTM structure to enable a static layer through which they could examine the impacts of different static catchment attributes. Third, they investigated network internal embeddings which is the first time in hydrology which I have seen, and provided some insights (not so perfect, as I would expand on later). These are all novel and I believe the paper should eventually be accepted.

Upon deeper examination I indeed found some issues related to potentially un-robust analysis, points of confusion and lack of clarity, need for more hydrologic insights, and somewhat superficial discussion in the exploration of embeddings. Some relevant citations are also missing. Thus I rate the manuscript a moderate revision. The comments below are not to cast the paper in a negative way, but they are in the hope of helping the authors improve the paper to a strong state before publication.

Major comments:

1. Hydrologic understanding: the discussion of the clustering and embeddings was, shall I say, not entirely satisfying. I liked the novelty of the visualization and the construct of the LSTM to enable this. It helped us understand a bit more about how LSTM works. However, I craved for a bit more hydrologic understanding. The discussion in section 3.4 was a bit sporadic and not so memorable. The take-home message appears to be “the EA-LSTM is able to learn complex interactions between catchment attributes that allows for grouping different basins”. Stopping here does not help with the long-standing criticism of machine learning as a blackbox. I had hoped to gain some deeper hydrologic insights, e.g., why different basins were grouped together? What is the characteristic of each cluster and how are these clusters different from previous catchment clustering schemes, e.g., (Berghuijs et al., 2014; Carrillo et al., 2011; Fang & Shen, 2017; Sawicz et al., 2011; Toth, 2013; Troch et al., 2013)? To go deeper it may not need additional work, but more thoughts about the results.

To avoid misunderstandings, we would like to clarify that we see the main contribution of this paper as i) demonstrating of the LSTM-based modeling approach for large-scale hydrological modeling in general (building upon the results of Kratzert et al., 2018) ii) introducing the EA-LSTM and iii) benchmarking vs. a large set of well-established hydrological models.

With this premise we would like to address the comment regarding Section 3.4: Within our EA-LSTM, we include an embedding layer (the static LSTM input gate), that can 'learn' about catchment diversity purely from discharge data. Analyzing the (physically) meaningfulness of the learned embedding can be seen as a (fairly simple) form of explainable AI. Although, as said above, our paper is mainly intended as a modeling paper, we think that Section 3.4 has the following benefits (copied from our answers to Reviewer #1):

- The blackbox i somewhat opened. The section provides at least some intuition about what happened in the embedding layer of the input gate in the EA-LSTM (namely grouping of basins in a way that matches expectations).
- It provides an example of the kind of analysis that are possible with the EA-LSTM in general and potentially opens the door for many follow-up studies in the future. Such studies could either concentrate more on the interpretability of LSTM-based models or try to extract new hydrological understanding from the learned groupings.

Given the above mentioned scope and the benefits of the section, we would like to avoid extending the hydrological interpretation of Section 3.4. Especially, because here we are not analyzing the model performance, but rather just examine the intrinsic properties of the model. Additionally, as the reviewer herself/himself cites, doing a full-fledged cluster analysis is the work of many individual publications themselves and would clearly be out of scope here.

2. More robustness: I'm afraid many of the attributes in Table 4 are correlated in space and it may be not very robust to draw conclusions from them especially for attributes that are not the highest ranking. For example, does geological permeability really stand position #9? Can we take it that permeability is the second important factor amongst non-climatic factors? This is somewhat surprising and is worth more discussion, but I'm afraid it might just be due to coincidence. To see so the authors could remove some basins (randomly or removing a spatial cluster) or attributes (as the factors tend to have interaction in these kinds of factor analysis) and train again and see how this table react to the perturbation.

First off, we would like to state that any spatial correlation in physical catchment features is real information that can and should be leveraged by regional models. We even explicitly did not include latitude/longitude inputs to our model in this study so that only real, physically-based information is leveraged directly by the EA-LSTM.
This is discussed, for example, by Addor et al (2018), and our findings are in line with the results of said publication. The results may thus be less surprising than indicated in this comment. In this context we would also like to mention that we did an independent robustness analysis by perturbing the features with gaussian noise (see L395ff), which shows the reliance (and robustness) of the model with respect to changes of the features.

We do however agree that the results do not form a particularly strong ranking. Regarding this point, it is important for us to emphasize that in the original contribution did not claim anywhere that the absolute rank of any particular feature has a meaning. This was a model sensitivity

analysis, which is common for modeling studies. The only conclusions that we drew from this sensitivity analysis are:

- "*..the most sensitive catchment attributes are topological features [...] and climate indices [...].*" (L 422f)
- "*Certain groups of catchment attributes did not typically provide much additional information. These include vegetation indices [...], as well as the annual vegetation differences. Most soil features were at the lower end of the feature ranking*" (L 423ff)

These seem to be valid conclusions of a sensitivity analysis like this.

That said, our experiments suggest that the obtained qualitative ranking of the feature groups (like climatic, topological, soil and vegetation) is rather robust. To strengthen upon this statement, we added below the results of the same analysis for all 8 repetitions of the same model settings (the EA-LSTM optimized with the basin average NSE) as used in Table 4. As we can see from these tables, the qualitative ranking of these feature groups remained similar. We hope that the reader does focus on exact rankings or exact sensitivity values of any particular feature but rather on the overall image of Table 4 - which is why we grouped these into categories in the first place.

We also agree with the reviewer that this might not be clear from the way the manuscript is currently written, and thus the results could be questioned as being a *coincidence.* We will therefore update the manuscript to clarify regarding this point.

Newly added passage:
Note that the results between the 8 model repetitions (not shown here) vary slightly in terms of sensitivity values and ranks. However, the quantitative ranking is robust between all 8 repetitions, meaning that climate indices (e.g. aridity and mean precipitation) and topological features (e.g. catchment area and mean catchment elevation) are always ranked highest, while soil and vegetation features are of less importance and are ranked lower. It is worth noting that our rankings qualitatively agree with much of the analysis by Addor et al. (2018).

```
p_mean                  0.682248
aridity                 0.564276
area_gages2             0.504591
elev_mean               0.459893
high_prec_dur           0.406671
frac_snow               0.405564
high_prec_freq          0.382006
slope_mean              0.370855
geol_permeability       0.352949
carbonate_rocks_frac    0.339022
clay_frac               0.330383
pet_mean                0.310769
low_prec_freq           0.299585
soil_depth_pelletier    0.273934
p_seasonality           0.272786
frac_forest             0.267421
sand_frac               0.255156
soil_conductivity       0.243641
low_prec_dur            0.219104
gvf_max                 0.213809
gvf_diff                0.212412
lai_diff                0.208096
soil_porosity           0.194036
soil_depth_statsgo      0.191936
lai_max                 0.190274
silt_frac               0.183365
max_water_content       0.158722
```

```
p_mean                  0.737930
elev_mean               0.620351
area_gages2             0.520789
frac_snow               0.481682
clay_frac               0.471752
aridity                 0.407523
gvf_max                 0.335463
geol_permeability       0.296405
soil_depth_pelletier    0.291952
pet_mean                0.290072
slope_mean              0.282145
low_prec_freq           0.280580
soil_depth_statsgo      0.274308
soil_conductivity       0.274178
silt_frac               0.259220
high_prec_dur           0.259150
p_seasonality           0.250047
low_prec_dur            0.248930
sand_frac               0.243023
carbonate_rocks_frac    0.235574
frac_forest             0.232315
gvf_diff                0.222452
high_prec_freq          0.202322
lai_diff                0.192319
lai_max                 0.168555
soil_porosity           0.167466
max_water_content       0.142825
```

```
elev_mean               0.592299
p_mean                  0.576201
aridity                 0.506481
area_gages2             0.474934
frac_snow               0.447427
clay_frac               0.443380
carbonate_rocks_frac    0.432280
slope_mean              0.400347
geol_permeability       0.368472
pet_mean                0.368429
soil_depth_pelletier    0.342860
gvf_max                 0.329133
sand_frac               0.314407
high_prec_freq          0.301476
soil_conductivity       0.295581
high_prec_dur           0.279304
gvf_diff                0.279032
p_seasonality           0.276549
silt_frac               0.267486
low_prec_freq           0.233236
soil_porosity           0.212450
soil_depth_statsgo      0.212345
frac_forest             0.193351
lai_max                 0.192644
lai_diff                0.185409
max_water_content       0.171502
low_prec_dur            0.167153
```

```
elev_mean               0.687521
p_mean                  0.675504
aridity                 0.524162
low_prec_dur            0.427454
soil_depth_pelletier    0.403478
clay_frac               0.401946
frac_snow               0.396246
area_gages2             0.384397
high_prec_dur           0.383352
slope_mean              0.378002
gvf_max                 0.345176
low_prec_freq           0.342597
pet_mean                0.327042
geol_permeability       0.323855
p_seasonality           0.322263
frac_forest             0.320437
silt_frac               0.292820
high_prec_freq          0.273888
max_water_content       0.245695
soil_depth_statsgo      0.245633
sand_frac               0.203782
soil_conductivity       0.201373
gvf_diff                0.195265
carbonate_rocks_frac    0.188165
lai_diff                0.173871
lai_max                 0.141322
soil_porosity           0.113825
dtype: float64
```

p_mean	0.683777	elev_mean	0.614620
elev_mean	0.535805	p_mean	0.600607
aridity	0.474985	frac_snow	0.515789
area_gages2	0.473146	aridity	0.506338
frac_snow	0.430077	area_gages2	0.439643
high_prec_freq	0.429197	soil_depth_pelletier	0.366733
slope_mean	0.406997	slope_mean	0.346615
soil_depth_pelletier	0.387183	clay_frac	0.343064
carbonate_rocks_frac	0.375872	carbonate_rocks_frac	0.329496
clay_frac	0.357481	gvf_max	0.318784
geol_permeability	0.344788	high_prec_freq	0.314248
gvf_max	0.327458	p_seasonality	0.309494
gvf_diff	0.324827	low_prec_freq	0.305372
pet_mean	0.320391	geol_permeability	0.285597
low_prec_freq	0.310324	sand_frac	0.285117
p_seasonality	0.291569	high_prec_dur	0.272177
high_prec_dur	0.273754	low_prec_dur	0.235999
silt_frac	0.272043	pet_mean	0.231612
low_prec_dur	0.241519	silt_frac	0.230806
soil_depth_statsgo	0.226876	gvf_diff	0.225315
max_water_content	0.219675	soil_conductivity	0.222055
sand_frac	0.215917	frac_forest	0.194465
soil_conductivity	0.214915	soil_depth_statsgo	0.189208
frac_forest	0.196826	soil_porosity	0.177579
soil_porosity	0.189546	lai_diff	0.166245
lai_max	0.173699	lai_max	0.157240
lai_diff	0.155421	max_water_content	0.143066

elev_mean	0.624811	p_mean	0.690424
p_mean	0.607064	elev_mean	0.563552
aridity	0.483825	frac_snow	0.557419
area_gages2	0.437059	aridity	0.513616
p_seasonality	0.421215	area_gages2	0.464579
slope_mean	0.390884	high_prec_freq	0.374508
frac_snow	0.390787	high_prec_dur	0.359950
high_prec_freq	0.382907	gvf_diff	0.333755
high_prec_dur	0.349156	soil_depth_pelletier	0.325056
gvf_max	0.349014	geol_permeability	0.324826
geol_permeability	0.335745	pet_mean	0.324743
soil_depth_pelletier	0.323830	clay_frac	0.322207
carbonate_rocks_frac	0.309022	slope_mean	0.317398
gvf_diff	0.288544	gvf_max	0.311988
pet_mean	0.287186	sand_frac	0.305872
clay_frac	0.284030	low_prec_dur	0.281842
low_prec_freq	0.245233	p_seasonality	0.278623
frac_forest	0.224514	silt_frac	0.271964
soil_conductivity	0.219839	carbonate_rocks_frac	0.267376
silt_frac	0.212018	soil_conductivity	0.259286
low_prec_dur	0.208801	low_prec_freq	0.237494
sand_frac	0.183467	lai_diff	0.178141
soil_depth_statsgo	0.180952	frac_forest	0.177963
lai_diff	0.178751	lai_max	0.177861
max_water_content	0.166533	soil_porosity	0.176054
soil_porosity	0.146432	soil_depth_statsgo	0.158091
lai_max	0.145553	max_water_content	0.127484

3. Details for reproducibility: one of the selling points of the paper was the high performance. Hence it imperative that the results are reproducible. Are the transformations applied for input and output? How many layers of LSTM were used (in comparison with authors' HESS 2018 paper, this choice seemed ad hoc?)? How was the ranking for Table 4 done indeed? This was a local method, so what is the origin for perturbation?

All information demanded by the reviewer are already reported in the manuscript:

- "Are the transformations applied for input and output?" L 247 "*All input features (both static and dynamic) were standardized (zero mean, unit variance) before training*". However, we agree that such an information should probably be placed in the data section (Section 2.4) and will update the manuscript accordingly.
- "How many layers of LSTM were used [...]?" The number of LSTM layers (one LSTM layer) is specified alongside the other network details in the Appendix B (L 562).
- "How was the ranking for Table 4 done[...]? " The details on how to derive the feature ranking is explained exhaustively in Section 2.6.2 "Robustness and Feature Ranking". Concretely, regarding the ranking of Table 4: "*Further, since we predict one time step of discharge at the time, we obtain this sensitivity measure for each static input for each day in the validation period. A global sensitivity measure for each basin and each feature is then derived from taking the average absolute gradient (Saltelli et al., 2004).*" and then L. 420f "*Table 4 provides an overall ranking of dominant sensitivities. These were derived by normalizing the sensitivity measures per basin to the range (0,1) and then calculating the overall mean across all features*"
- "... what is the origin for perturbation?" We used the optimized parameters as starting value. There might be some confusion here. We do not solve the gradient computation by numerical approximation, but rather calculate the gradients analytically through backpropagation. So if at all, the true values for the static catchment attributes can be seen as the origin of perturbation. In the original manuscript this is explained in 2.6.2 "Robustness and Feature Ranking" L.251f.

4. Share more experience please: there are many choices which were unexplained, and the community would benefit from the authors providing more discussion of what worked and what did not during their experiments. How did other objective functions do? What if you don't do ensemble averaging? How large are the impacts of hyperparameters, e.g., hidden layers and learning rates? These do not necessarily need figures and could be answered by a couple of sentences. Some minor points below are related to this.

Sadly, we do not know how we can do this. We tried to provide as much information as possible. And, to our knowledge, no choices in our network architecture or training procedure remained unexplained. Appendix B explains the hyperparameter search settings. We did not experiment with different learning rates and can't share any experiences on this question.

Furthermore, we did not test any other objective functions than the two reported in this paper. Hyperparameter search was performed using MSE (the machine learning community standard for regression tasks).

The only thing that comes to mind is that we did not report the results of all considered configurations and if wished we can update the Appendix B accordingly with a short description. As a short summary: The median model performance (across the basins) remains more or less stable between most configurations, while the most variance can be observed in the mean NSE. Two layers did not provide any meaningful improvement, that would justify the additional computational cost. However, our hyperparameter search was not exhaustive and at no point in the manuscript we claim to have found the best possible architecture for this task.

5. The authors should also expand on why climatic factors showed up on top of table 4. It appears other static basin physical attributes were not important at all. Does this suggest catchment co-evolution? A potential indication of overfitting (to climatic factors that obviously vary), and more discussion is begging to be done here.

The climatic factors show up on the top of the table, since through the method of Morris they have the highest gradient. We don't know of any experiment that would tell us *why* climate factors appear there (i.e. why they have the highest gradient), except hydrological intuition. (This is not different than any sensitivity analysis for any type of hydrologic model - sensitivity analyses do not answer questions about 'why' certain features are more sensitive) As such, these results in isolution do not suggest catchment co-evolution. They tell us that the model uses certain features more heavily than others. However, these findings are also in line with the results reported by Addor et al. (2018), as we state in L 428 "*It is worth noting that our rankings qualitatively agree with much of the analysis by Addor et al. (2018).*"

Also, this table doesn't suggest that physical attributes are unimportant, just that they are not as important as climate features. Again, this agrees with previous literature, as cited. This intuition that climate-related factors are the dominant drivers of hydrological systems, for example, models are often tested in terms of their ability to predict departures from the Budyko curve. We therefore do not see any indication of overfitting from this analysis.

Minor points:

1. I'm at a loss to understand the opening statement about streamflow being an out-standing problem. At what point is this problem solved vs not solved? Is there a hard threshold? Did the present work solve this problem?

To clarify: The sentence in question reads: "*Regional rainfall-runoff modeling is an old but still mostly out-standing problem in Hydrological Sciences"*. Here, *out-standing* is referring to *regional* modeling, not to streamflow modeling in general. There is no hard threshold to determine when a problem like this is solved (and we believe that the sentence does not imply

that either; as a matter of fact we added the word “mostly” to avoid such a conclusion). However, the benchmarking in our paper with state-of-the-art regionalization methods and the fact that the proposed LSTM-based modeling approach significantly (and by far margins) outperforms these models, suggest that there is (or at least was) still significant room to improve how the community addresses this problem.
We believe that most readers will not be puzzled by the provided formulation and will therefore leave it unchanged.

2. L73, “which part of the network are used for a given basin”—this sentence is difficult to interpret at this point. What does “used for” mean here.

We added some clarity to this sentence: “*Concretely, we propose an adaption of the LSTM where catchment attributes explicitly control which parts of the LSTM state space are used for a given basin*”

Old passage:
Concretely, we propose an adaption of the LSTM where catchment attributes explicitly control which parts of the network are used for a given basin.

New passage:
Concretely, we propose an adaption of the LSTM where catchment attributes explicitly control which parts of the LSTM state space are used for a given basin.

3. L76, “similarly behaving”. Is this referring streamflow responses or attributes? (only the former would be called a behavior, but this work didn’t seem to include streamflow response in the clustering part)

“Behavior” here refers to the similarity in the rainfall-runoff dynamics, as suggested by the reviewer. This is also stated implicitly in the two sentences directly preceding the one in question (L74f) “*...it can learn how to combine different parts of the network to simulate different types of rainfall-runoff behaviors. In principle, the approach explicitly allows for sharing parts of the networks for similarly behaving basins...*”

4. L78, “embedding”. This is a natural language processing jargon. Quite difficult for hydrologists to comprehend. I think it would be reader friendly if the authors spend two sentences explaining this word. My understanding is that embeddings are not just hidden layer activations, but a mapping of inputs to an ordered hidden space that has meanings. For example, the hidden layers of machine translation layers form an embedding. Each ranked item in the embedding in NLP can be related to a linguistic concept.

Historically, “embedding” is not a term from the field of natural language processing, but rather a general mathematical concept. Maybe the reviewer is confusing this term with “word embeddings”, which is a term-of-art from natural language processing, but is not what we are

referring to. More importantly, L. 77f defines the term embedding exactly: "*..our adaptation provides a mapping from catchment attribute space into a learned, high-dimensional space, i.e. a so-called embedding*".

5. L117 "some amount of information" is fuzzy. Is it about catchment attributes or about streamflow responses? This is critically important as the two have very different meaning regarding what would be done. From reading the later parts, here you seem to refer to static Attributes.

We changed the previous sentence to: "*..our objective is to build a network that learns to extract information that is relevant to rainfall-runoff behaviors from observable catchment attributes.*" so that the context is hopefully clearer.

Old passage:
To reiterate from the introduction, our objective is to build a network that learns catchment similarities directly from rainfall-runoff data in multiple basins.

New passage:
To reiterate from the introduction, our objective is to build a network that learns to extract information that is relevant to rainfall-runoff behaviors from observable catchment attributes.

6. L122, regarding using static attributes as a constant array. It would be relevant to cite (Fang et al., 2017) which used this setup and was already distinguishing different landscapes using static attributes as inputs to LSTM. It occurs this paper should at least be mentioned in the present one.

Using static attributes as constant input is not something we are claiming is novel. More specifically, this method has been applied many times before in the field of machine learning (e.g. Karpathy and Fei-Fei 2014, Wen et al. 2015, Wen et al. 2016). The technique was not originally proposed by Fang et al. (2017) and their manuscript is not working on the same topic as our manuscript (rainfall-runoff modeling), we therefore do not see this as an especially appropriate reference to cite in this case.

7. L134-135. This is an interesting setup. It's worth mentioning that, from Eq 9 & 11, what was selected by the input gate were not only x_d but also h from the last step.

It is not entirely clear what the reviewer wants to suggest. If this refers to the fact that that h[t-1] is used in the forget and output gate, as well as the cell update (g[t]), then they are right. The input gate however, does not get any information of x_d[t] in our proposed EA-LSTM and neither from h[t-1]. We hope by changing the following sentence "*..while the dynamic and recurrent inputs control what information is written..*" we can resolve the confusion.

Old passage:
The static features control, through input gate (i), which parts of the LSTM are activated for any individual catchment, while the dynamic inputs control what information is written into the memory (g[t]), what is deleted (f[t]), and what of the stored information to output (o[t]) at the current time step t.

New passage:
The static features control, through input gate (i), which parts of the LSTM are activated for any individual catchment, while the dynamic and recurrent inputs control what information is written into the memory (g[t]), what is deleted (f[t]), and what of the stored information to output (o[t]) at the current time step t.

8. L158 – what happened when you used other loss functions?

We are unsure about the exact intent of this question. We used two loss functions in this manuscript and compared the results. From a hydrological modelling perspective it seems obvious that different loss functions might provide different optimization results. Designing and choosing (good/correct) objective functions is an old and important problem in hydrology. It is highly non-trivial, yet unsolved and surrounded by many discussions. However, it is also not the focus of this contribution and we therefore view the testing of more loss functions as out of scope.

9. L171 “25,000 km2” – is it really appropriate to model those with an area of 25,000 km2 the same as other smaller basins?

Although results of experiments not shown in this manuscript suggest there is no problem with doing so, in this manuscript only basins with an area smaller than 2000km² were used. As stated in L. 174 we use the same 531 basins as Newman et al. (2017): to cite their manuscript: “*We subset the complete Newman et al. (2014) basin list to remove...basins larger than 2000 km²*”. That said, we agree that we missed to state this clearly in our manuscript and therefore adapt L 176 to add the following sentence “*Furthermore, out of the 671 basins, only those with an area smaller than 2000km² were kept.*”

Old passage:
We used the same 531 basins from the CAMELS data set as Newman et al. (2017). The basins are mapped in Fig. 2. These basins were chosen out of the full set because some of the basins have a large (>10 %) discrepancy between different strategies for calculating the basin area, and incorrect basin area would introduce significant uncertainty into a modeling study. The basin selection and subset is described by Newman et al. (2017).

New passage:
We used the same subselection of 531 basins from the CAMELS data set that was used by Newman et al. (2017). These basins are mapped in Fig. 2, and were chosen (by Newman et al. (2017)) out of the full set because some of the basins have a large (>10 %) discrepancy between different strategies for calculating the basin area, and incorrect basin area would introduce
significant uncertainty into a modeling study. Furthermore, only basins with a catchment area smaller than 2000 km^2 were kept.

10. L194 – "favor of"

Corrected, thank you.

Old passage:
We chose to use existing model runs so to not bias the calibration of the benchmarks to possibly favor of our own model.

New passage:
We chose to use existing model runs so to not bias the calibration of the benchmarks to possibly favor our own model.

11. L222, regarding the ensemble averaging, readers deserve to know, how big is the spread? What if you don't take the average? Sometimes the ensemble mean gets a better R2 but it misses peaks.

Yes, it is true that taking the ensemble mean will reduce variance. We've not explored more complex ensemble techniques, of which many exist. However, we see testing different ensembling strategies as out-of-scope for this paper.

12. It is unclear what "six different settings and eight different models" are.

This is explained in the preceding sentences.

- Regarding the "six different settings" L 219f "*All three model configurations were trained using the squared-error performance metrics discussed in Sect. 2.3 (MSE and NSE*). This resulted in six different model/training configurations.*"
- Regarding the "eight different models" L 221f "*To account for stochasticity in the network initialization and in the optimization procedure (we used stochastic gradient descent), all networks were trained with n = 8 different random seeds*"

The phrase quoted by the reviewer from L. 224 (immediately following the two sentences quoted) pulls these sentences together "*In total, we trained and tested six different settings and eight different models per setting for a total of 48 different trained LSTM-type models*"

13. L261. might be useful to say you extracted gradients from the learned network after training (correct?), as some readers are unfamiliar with how this is done. However, these gradients are time-step dependent.

Indeed, we calculated the gradients w.r.t. the static inputs from the trained model, since we are interested in analyzing the robustness and feature ranking of a trained network, not of a randomly initialized one. In L. 244 we stated this fact for the model robustness "*To estimate the robustness of the trained model to uncertainty in the catchment attributes…*". We will add a similar sentence to the feature ranking to avoid possible confusion around analyzing untrained models.

Old passage:
To provide a simple estimate of the most important static features, we used the method of Morris (Morris, 1991).

New passage:
To provide a simple estimate of the most important static features of the trained model, we used the method of Morris (Morris, 1991).

14. Also, why is it called global sensitivity test? It is also local, around a origin for perturbation.

Citing Campolongo et al. (2015) from their introduction "*The Morris method is simple to understand and implement, and its results are easily interpreted. Furthermore it is economic in the sense that it requires a number of model evaluations is in the number of model factors. The method can be regarded as global as the final measure is obtained by averaging a number of local measures (the elementary effects), computed at different points of the input space.*" To clarify the result of Eq. 14 (or Eq. 15 in our case), this is not a global measure in the sense that the entire space of possible values is considered, but in the sense that more points are considered to derive the sensitivity (see Saltelli et al., 2004). This is reflected in our statement in L. 263f: "*A global sensitivity measure for each basin and each feature is then derived from taking the average absolute gradient (Saltelli et al., 2004)*"

15. L264 better say "the average of absolute gradients across all basins and all time steps", and----why absolute?

Here, we are still referring to a global sensitivity measure for each individual basin. Therefore, "*for each basin*" is correct in this sentence. The averaging across multiple basins is then applied to derive the values in Table 4 (see answer to major comment #3), after normalizing the sensitivity measures to the range (0,1) per basin. Absolute, because otherwise oscillating (positive, negative) gradients, have the potential to cancel and (erroneously) suggest that the respective feature(s) are unimportant. Furthermore, taking absolute values is the proposed

method for deriving the global sensitivity measure from these local points and is referred to as μ* in the literature (e.g. Saltelli, 2004; Campolongo et al. 2011).

16. L267-268 “represent xxx into xxx"? the sentence does not make grammar sense. please fix. This is obviously an expansion of from 27 to 256. Why would this be really necessary?

We do not see a grammatical error in this sentence. Embedding can be used as a noun, which makes the phrase “*[this] vector […] represents an embedding of xxx into yyy*” grammatically correct.

This transformation is necessary, since the resulting input gate must be a vector of 256-dimensions - the same size as the LSTM has cell states. This is basically the same as in every other gate where e.g., the 5-dimensional dynamic inputs (the 5 meteorological variables) have to be transformed into a vector of 256-dimensions for the forget and output gate and the cell update respectively.

17. Table 2. this value is indeed the highest I have seen. Good work!

Thank you.

18. L380. Why 447 basins now? What are missing?

The first sentence in Section 3.2 (L.363) explains this: “*The results in this section are calculated from 447 basins that were modeled by all benchmark models*”.

It’s important to reiterate that we used benchmark models that were run by the respective model development groups. We did not run our own benchmark models. This is critical because we want to give the benchmark models the highest possible chance of success - the presumption being that the respective model development groups are the most well-qualified to run their own models. Notice that this is a common strategy in model intercomparison and model benchmarking studies (e.g., Best et al., 2015)

19. L410 Unsure how this answers the question if the network just remembers. The logic is Confusing.

We think that the general results of the robustness analysis (as shown in the boxplot in Fig. 6) indeed address whether the network is simply remembering basins. If we understand the reviewer correctly, s/he is referring to pure overfitting against the static attributes and that the LSTM simply remembers all 531 catchments individually. If the LSTM simply remembers all 531 catchments individually, there would not be slow degradation in performance (as seen as increase in the variance of the boxplot over increasing level of additive noise) but rather a more drastic performance drop, when not using the exact catchment attributes for each basin. We will add a sentence that better describes this result.

Old passage:
As expected, the model performance degrades with increasing noise in the static inputs. However, the degradation does not happen abruptly but smoothly with increasing levels of noise.

New passage:
As expected, the model performance degrades with increasing noise in the static inputs. However, the degradation does not happen abruptly but smoothly with increasing levels of noise, which is an indication that the LSTM is not over-fitting on the static catchment attributes. That is, it is not remembering each basin with its set of attributes exactly, but rather learns a smooth mapping between attributes and model output.

20. L414, mean precipitation, etc --- aren' these supposed to be climatic inputs rather than static? (can we not let the network generalize it from the forcing data)?

Mean precipitation, high precipitation duration etc. are indeed climatic inputs, but also static inputs since these are aggregated values over the time series (see Addor et al. 2017). The network would only be able to derive statistics like mean precipitations internally from the time length we derive as the input for predicting a single day (here we use an input sequence length of 365 days).

21. Table 4. Echoing a major point raised above. What further conclusions can be drawn from the fact that climatic attributes take the most important positions? catchment Co-evolution theory?

Indeed, what could be inferred here? It's a good question. Certainly this type of speculation is far outside the scope of this paper. We are not prepared to speculate on climate-driven catchment co-evolution, but we suspect that the 30-year data record in CAMELS is not long enough to address this question

22. L454 "before vs. after the transformation into the embedding layer". This is a good comparison, although later there didn't seem to be much comment on this comparison

There are a few comparisons made throughout the analysis:

- L 453ff "*In all cases with cluster sizes less than 15, we see that clustering by the values of the embedding layer provides more distinct catchment clusters than when clustering by the raw catchment attributes*"
- L 462ff "*In both the k = 5 and k = 6 cluster examples, clustering by the EA-LSTM embedding layer reduced variance in the hydrological signatures by more or*

approximately the same amount as by clustering on the raw catchment attributes. The exception to this was the hfd-mean date, which represents an annual timing process (i.e., the day of year when the catchment releases half of its annual flow). This indicates that the EA-LSTM embedding layer is largely preserving the information content about hydrological behaviors, while overall increasing distinctions between groups of similar catchments"

- L 471ff "*Although latitude and longitude were not part of the catchment attributes vector that was used as input into the embedding layer, both the raw catchment attributes and the embedding layer clearly delineated catchments that correspond to different geographical regions within the CONUS*"

For each of the steps of the cluster analysis (silhouette plots, variance reduction and cluster results shown on the map of the USA), we actually gave a direct comparisons between the results using the embedding of the EA-LSTM or using the raw catchment attributes. We are unsure what kind of additional comments are expected from the reviewer.

23. UMAP—might be good to briefly explain what it does. Is it just PCA?

We agree that the explanation of the UMAP method could be extended in Section 2.6.3 and will update the manuscript accordingly.

New passage describing UMAP:
Finally, we reduced the dimension of the input gate embedding layer (from R^256 to R^2) so as to be able to visualize dominant features in the input embedding. To do this we use a dimension reduction algorithm, called UMAP (McInnes et al., 2018) for "Uniform Manifold Approximation and Projection for Dimension Reduction". UMAP is based on neighbour graphs (while e.g., principle component analysis is based on matrix factorization), and it uses ideas from topological data analysis and manifold learning techniques to guarantee that information from the high dimensional space is preserved in the reduced space. For further details we refer the reader to the original publication by McInnes et al. (2018).

24. L479 Honestly, it's not that easy to see which cluster you are talking about. could use some annotation on the plot.

This is a good idea and we will update the plot accordingly.

25. L489 I found this discussion, as a take-home message, to be somewhat superficial, and unsurprising. I'd appreciate somewhat more in-depth discussion about the hydrology.

We understand and appreciate the reviewer's perspective (more is usually better), however it's hard to see from this comment what the reviewer finds missing in our analysis. We did give quite a lot of hydrological discussion in the context of analyzing the embedding layer - does the reviewer see something in our analysis that is missing? Is there something they might hope to

learn that we didn't explore? We would love suggestions about how to improve this analysis, but just asking for more is not really an actionable suggestion.

Regarding the reviewers' suggestion that our conclusions were not surprising, I guess surprising is somewhat subjective. We were generally happy that (a) the model performed as well as it did against benchmarks, and (b) that the similarity analysis generally agreed with previous literature. This means that the model at least appears to be giving the right answers for the right reasons.

26. Figure 11 these colors do not mean anything. It is a bit confusing. Why not use a at least partially consistent color scheme?

These colors actually do mean something (and it was actually somewhat difficult to get the colors to match on the various plots). These colors present the results of several clustering analyses, and are categorical labels. Therefore, we chose to color the basins in a categorical color palette, where each color reflects one cluster class. This makes a categorical color-scheme necessary, since there is no intrinsic ordering (excluding continuous, sequential and diverging color schemes). Furthermore, we made sure that the clusters between the different subplots are more or less colored similarly, so it is easier to compare between the subplots. What else does the reviewer meant by "partially consistent color scheme"? Consistent with what? Certainly the color scheme is consistent between subplots in the figure.

27. Figure 12 better annotate axes even if they don't mean much

We decided to exclude the 2D-coordinates of the UMAP embedding because, as the reviewer suggested herself/himself, they do not mean anything. We would therefore argue that they are probably more confusing and distracting and the reader could ask what why this basin has an embedding coordinate of (4,2) and the other basin only of (0,-0.5) (both are arbitrary sets).

28. L524. I am confused why this is called regional, as the LSTM was trained with all basins over CONUS. What would constitute a model that is not regional?

Regional modeling in Hydrology has a very specific meaning. The alternative is a local model (i.e., one that is calibrated to a specific basin). The second reviewer suggested that this is potentially a universal rainfall-runoff model that could be applied to basin groups of any scale (small regions, US scale, continental scale or even globally), but our intent was to draw a connection with what is a named (and well-defined) problem in Hydrology.

29. L529-530. It either goes against a belief or it does not. Can't go "somewhat against". And, the logic here is not quite clear. This paper is not about parameter identification. The fact that the network works does not imply that parameters can be identified. First the LSTM parameters cannot be interpreted. Second, even very different parameters could give you similar predictions.

First, it's actually possible for two opinions to partially disagree or somewhat disagree, however we changed this part to: "*This result challenges the idea that runoff time series alone only contain enough information…"*

Secondly, we are not sure if we understand the comment about parameter identification:

- The technical correctness of the statement "*the LSTM parameters cannot be interpreted*" depends on one's understanding of interpretation (for a lengthy discussion on this topic we refer to Lipton, 2016). The LSTM parameters are (maybe) not one-to-one translatable into physical properties as some of the hydrological model parameters, however this criticism is no less valid for conceptual models: it might be questionable what a e.g., catchment-wide infiltration value represents.

- However, the function of each individual parameter in the trained LSTM could indeed be interpreted, to see if a certain weight e.g., thresholds to specific temperatures in the input. The huge number of parameters however, makes such work difficult. This is not really related to the point of the sentence in question, however, which is about deficiencies in hydrology models, not about the interpretability of LSTM parameters.

- We are also not sure if we understand the second point of the review in this context. Indeed, different parameters can give similar predictions and overall performances, as we have shown in the paper. However, what is the point here regarding our statement that we think traditional large-scale hydrological models can be structurally improved?

References:

Addor, N., Newman, A.J., Mizukami, N. and Clark, M.P., (2017). The CAMELS data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences (HESS)*, *21*(10), pp.5293-5313.

Addor, N., Nearing, G., Prieto, C., Newman, A.J., Le Vine, N. and Clark, M.P., (2018). A ranking of hydrological signatures based on their predictability in space. *Water Resources Research*, *54*(11), pp.8792-8812.

Best, M. J., Abramowitz, G., Johnson, H. R., Pitman, A. J., Balsamo, G., Boone, A., ... & Ek, M. (2015). The plumbing of land surface models: benchmarking model performance. *Journal of Hydrometeorology*, *16*(3), 1425-1442.

Campolongo, F., Cariboni, J., Saltelli, A., & Schoutens, W. (2005). Enhancing the Morris method. *Sensitivity Analysis of Model Output. Proceedings of the 4th International Conference on Sensitivity Analysis of Model Output (SAMO 2004)* (pp. 369-379).

Campolongo, F., Saltelli, A., & Cariboni, J. (2011). From screening to quantitative sensitivity analysis. A unified approach. *Computer Physics Communications*, *182*(4), 978-988.

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3128-3137).

Lipton, Z. C. (2016). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.

Newman, A.J., Mizukami, N., Clark, M.P., Wood, A.W., Nijssen, B. and Nearing, G., (2017). Benchmarking of a physically based hydrologic model. *Journal of Hydrometeorology*, *18*(8), pp.2215-2225.

Saltelli, A., Tarantola, S., Campolongo, F., and Ratto, M. (2004): Sensitivity analysis in practice: a guide to assessing scientific models. *pp. 94–100, Wiley Online Library*.

Wen, T. H., Gasic, M., Mrkšić, N., Su, P. H., Vandyke, D., & Young, S. (2015, September). Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1711-1721).

Wen, T. H., Gasic, M., Mrkšić, N., Barahona, L. M. R., Su, P. H., Ultes, S., ... & Young, S. (2016, November). Conditional Generation and Snapshot Learning in Neural Dialogue Systems. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 2153-2162).

~~Benchmarking a Catchment-Aware Long Short-Term Memory Network (LSTM) for Large-Scale~~ Towards Learning Universal, Regional, and Local Hydrological ~~Modeling~~Behaviors via Machine-Learning Applied to Large-Sample Datasets

Frederik Kratzert[1], Daniel Klotz[1], Guy Shalev[2], Günter Klambauer[1], Sepp Hochreiter[1,*], and Grey Nearing[3,*]

[1]LIT AI Lab & Institute for Machine Learning, Johannes Kepler University Linz, Austria
[2]Google Research
[3]Department of Geological Sciences, University of Alabama, Tuscaloosa, AL United States
[*]These authors contributed equally to this work.

Correspondence: Frederik Kratzert (kratzert@ml.jku.at)

Abstract. Regional rainfall-runoff modeling is an old but still mostly out-standing problem in Hydrological Sciences. The problem currently is that traditional hydrological models degrade significantly in performance when calibrated for multiple basins together instead of for a single basin alone. In this paper, we propose a novel, data-driven approach using Long Short-Term Memory networks (LSTMs), and demonstrate that under a 'big data' paradigm, this is not necessarily the case. By training a single LSTM model on 531 basins from the CAMELS data set using meteorological time series data and static catchment attributes, we were able to significantly improve performance compared to a set of several different hydrological benchmark models. Our proposed approach not only significantly outperforms hydrological models that were calibrated regionally but also achieves better performance than hydrological models that were calibrated for each basin individually. Furthermore, we propose an adaption to the standard LSTM architecture, which we call an Entity-Aware-LSTM (EA-LSTM), that allows for learning, and embedding as a feature layer in a deep learning model, catchment similarities. We show that this learned catchment similarity corresponds well with what we would expect from prior hydrological understanding.

1 Introduction

A longstanding problem in the Hydrological Sciences is about how to use one model, or one set of models, to provide spatially continuous hydrological simulations across large areas (e.g., regional, continental, global). This is the so-called *regional modeling problem*, and the central challenge is about how to extrapolate hydrologic information from one area to another – e.g., from gauged to ungauged watersheds, from instrumented to non-instrumented hillslopes, from areas with flux towers to areas without, etc. (Blöschl and Sivapalan, 1995). Often this is done using ancillary data (e.g. soil maps, remote sensing, digital elevation maps, etc.) to help understand similarities and differences between different areas. The regional modeling problem is thus closely related to the problem of prediction in ungauged basins (Blöschl et al., 2013; Sivapalan et al., 2003). This problem

is well-documented in several review papers, therefore we point the interested reader to the comprehensive reviews by Razavi and Coulibaly (2013) and Hrachowitz et al. (2013), and to the more recent review in the introduction by Prieto et al. (2019).

Currently, the most successful hydrological models are calibrated to one specific basin, whereas a regional model must be somehow 'aware' of differences between hydrologic behaviors in different catchments (e.g., ecology, geology, pedology, topography, geometry, etc.). The challenge of regional modeling is to learn and encode these differences so that differences in catchment characteristics translate into appropriately heterogeneous hydrologic behavior. Razavi and Coulibaly (2013) recognize two primary types of strategies for regional modeling: *model-dependent* methods and *model-independent* (data-driven) methods. Here, model-dependent denotes approaches where regionalization explicitly depends on a pre-defined hydrological model (e.g., classical process-based models), while model-independent denotes data-driven approaches that do not include a specific model. The critical difference is that the first tries to derive hydrologic parameters that can be used to run simulation models from available data (i.e., observable catchment characteristics). In this case, the central challenge is the fact that there is typically strong interaction between individual model parameters (e.g., between soil porosity and soil depth, or between saturated conductivity and an infiltration rate parameter), such that any meaningful joint probability distribution over model parameters will be complex and multi-modal. This is closely related to the problem of equifinality (Beven and Freer, 2001).

Model-dependent regionalization has enjoyed major attention from the hydrological community, so that today a large variety of approaches exist. To give a few selective examples, Seibert (1999) calibrated a conceptual model for 11 catchments and regressed them against the available catchment characteristics. The regionalization capacity was tested against seven other catchments, where the reported performance ranged between an Nash-Sutcliffe Efficiency (NSE) of 0.42 and 0.76. Samaniego et al. (2010) proposed a multiscale parameter regionalization (MPR) method, which simultaneously sets up the model and a regionalization scheme by regressing the global parameters of a set of a-priori defined transfer functions that map from ancillary data like soil properties to hydrological model parameters. Beck et al. (2016) calibrated a conceptual model for 1787 catchments around the globe and used these as a catalog of 'donor catchments', and then extend this library to new catchments by identifying the ten most similar catchments from the library in terms of climatic and physiographic characteristics to parameterize a simulation ensemble. Prieto et al. (2019) first regionalized hydrologic signatures (Gupta et al., 2008) using a regression model (random forests), and then calibrated a rainfall runoff model to the regionalized hydrologic signatures.

Model-independent methods, in contrast, do not rely on prior knowledge of the hydrological system. Instead, these methods learn the entire mapping from ancillary data and meteorological inputs to streamflow or other output fluxes directly. A model of this type has to 'learn' how catchment attributes or other ancillary data distinguish different catchment response behaviours. However, hydrological modeling typically provides the most accurate predictions when a model is calibrated to a single specific catchment (Mizukami et al., 2017), whereas data-driven approaches might benefit from large cross-section of diverse training data, because knowledge can be transferred across sites. Among the category of data-driven approaches are neural networks. Besaw et al. (2010) showed that an artificial neural network trained on one catchment (using only meteorological inputs) could be moved to a similar catchment (during a similar time period). However, the accuracy of their network in the *training* catchment was only a NSE of 0.29. Recently, Kratzert et al. (2018b) have shown, that Long Short-Term Memory (LSTM) networks, a special type of recurrent neural networks, are well suited for the task of rainfall-runoff modeling. This study already included

first experiments towards regional modeling, while still using only meteorological inputs and ignoring ancillary catchment attributes. In a preliminary study Kratzert et al. (2018c) demonstrated that their LSTM-based approach outperforms, on average, a well-calibrated Sacramento Soil Moisture Accounting Model (SAC-SMA) in an asymmetrical comparison where the LSTM was used in an *ungauged* setting and SAC-SMA was used in a *gauged* setting - i.e., SAC-SMA was calibrated individually for each basin whereas the LSTM never saw training data from any catchment where it was used for prediction. This was done by providing the LSTM-based model with meteorological forcing-data and additional catchment attributes. From these preliminary results we can already assume that this general modeling approach is promising and has the potential for regionalization.

The objectives of this study are:

(i) to demonstrate that we can use large-sample hydrology data (Gupta et al., 2014; Peters-Lidard et al., 2017) to develop a regional rainfall-runoff model that capitalizes on observable ancillary data in the form of catchment attributes to produce accurate streamflow estimates over a large number of basins,

(ii) to benchmark the performance of our neural network model against several existing hydrology models, and

(iii) to show how the model uses information about catchment characteristics to differentiate between different rainfall-runoff behaviors.

To this end we built an LSTM-based model that learns catchment similarities directly from meteorological forcing-data and ancillary data of multiple basins and evaluate its performance in a 'gauged' setting, meaning that we never ask our model to predict in a basin where it did not see training data. Concretely, we propose an adaption of the LSTM where catchment attributes explicitly control which parts of the ~~network~~ LSTM state space are used for a given basin. Because the model is trained using both catchment attributes and meteorological time series data, to predict streamflow, it can learn how to combine different parts of the network to simulate different types of rainfall-runoff behaviors. In principle, the approach explicitly allows for sharing parts of the networks for similarly behaving basins, while using different independent parts for basins with completely different rainfall-runoff behavior. Furthermore, our adaption provides a mapping from catchment attribute space into a learned, high-dimensional space, i.e. a so-called embedding, in which catchments with similar rainfall-runoff behavior can be placed together. This embedding can be used to preform data-driven catchment similarity analysis.

The paper is organized as follows. Section 2 (Methods) describes our LSTM-based model, the data, the benchmark hydrological models, and the experimental design. Section 3 (Results) presents our modelling results, the benchmarking results and the results of our embedding layer analysis. Section 4 (Discussion and Conclusion) reviews certain implications of our model and results, and summarizes the advantages of using data-driven methods for extracting information from catchment observables for regional modeling.

2 Methods

2.1 A Brief Overview of the Long Short-Term Memory network

An LSTM network is a type of recurrent neural network that includes dedicated memory cells that store information over long time periods. A specific configuration of operations in this network, so-called gates, control the information flow within the LSTM ~~(Hochreiter and Schmidhuber, 1997)~~(Hochreiter, 1991; Hochreiter and Schmidhuber, 1997). These memory cells are, in a sense, analogous to a state vector in a traditional dynamical systems model, which makes LSTMs potentially an ideal candidate for modeling dynamical systems like watersheds. Compared to other types of recurrent neural networks, LSTMs do not have a problem with exploding and/or vanishing gradients, which allows them to learn long-term dependencies between input and output features. This is desirable for modeling catchment processes like snow-accumulation and snow-melt that have relatively long time scales compared with the timescales of purely input driven domains (i.e., precipitation events).

An LSTM works as follows (see also Fig. 1a): Given an input sequence $\boldsymbol{x} = [\boldsymbol{x}[1], .., \boldsymbol{x}[T]]$ with T time steps, where each element $\boldsymbol{x}[t]$ is a vector containing input features (model inputs) at time step t ($1 \leq t \leq T$), the following equations describe the forward pass through the LSTM:

$$\boldsymbol{i}[t] = \sigma(\boldsymbol{W_i x[t]} + \boldsymbol{U_i h[t-1]} + \boldsymbol{b_i}) \tag{1}$$

$$\boldsymbol{f}[t] = \sigma(\boldsymbol{W_f x[t]} + \boldsymbol{U_f h[t-1]} + \boldsymbol{b_f}) \tag{2}$$

$$\boldsymbol{g}[t] = \tanh(\boldsymbol{W_g x[t]} + \boldsymbol{U_g h[t-1]} + \boldsymbol{b_g}) \tag{3}$$

$$\boldsymbol{o}[t] = \sigma(\boldsymbol{W_o x[t]} + \boldsymbol{U_o h[t-1]} + \boldsymbol{b_o}) \tag{4}$$

$$\boldsymbol{c}[t] = \boldsymbol{f}[t] \odot \boldsymbol{c}[t-1] + \boldsymbol{i[t]} \odot \boldsymbol{g}[t] \tag{5}$$

$$\boldsymbol{h}[t] = \boldsymbol{o}[t] \odot \tanh(\boldsymbol{c[t]}), \tag{6}$$

where $\boldsymbol{i}[t]$, $\boldsymbol{f}[t]$ and $\boldsymbol{o}[t]$ are the *input gate*, *forget gate*, and *output gate*, respectively, $\boldsymbol{g}[t]$ is the *cell input* and $\boldsymbol{x}[t]$ is the *network input* at time step t ($1 \leq t \leq T$), $\boldsymbol{h}[t-1]$ is the *recurrent input* $\boldsymbol{c}[t-1]$ the *cell state* from the previous time step. At the first time step, the hidden and cell states are initialized as a vector of zeros. $\boldsymbol{W}$, $\boldsymbol{U}$ and $\boldsymbol{b}$ are learnable parameters for each gate, where subscripts indicate which gate the particular weight matrix/vector is used for, $\sigma(\cdot)$ is the sigmoid-function, $\tanh(\cdot)$ the hyperbolic tangent function and $\odot$ is element-wise multiplication. The intuition behind this network is that the cell states ($\boldsymbol{c}[t]$) characterize the memory of the system. The cell states can get modified by the forget gate ($\boldsymbol{f}[t]$), which can delete states, and the input gate ($\boldsymbol{i}[t]$) and cell update ($\boldsymbol{g}[t]$), which can add new information. In the latter case, the cell update is seen as the information that is added and the input gate controls into which cells new information is added. Finally, the output gate ($\boldsymbol{o}[t]$) controls which information, stored in the cell states, is outputted. For a more detailed description, as well as a hydrological interpretation, see Kratzert et al. (2018b).

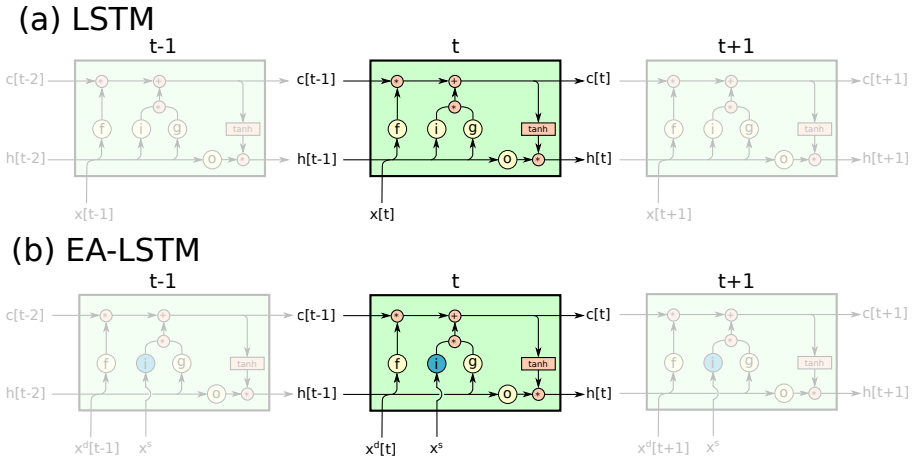


Figure 1. Visualization of (a) the standard LSTM cell as defined by Eq. (1-6) and (b) the proposed Entity-Aware-LSTM (EA-LSTM) cell as defined by Eq. (7-12).

2.2 A New Type of Recurrent Network: The Entity-Aware-LSTM

To reiterate from the introduction, our objective is to build a network that learns ~~catchment similarities directly from~~ to extract information that is relevant to rainfall-runoff ~~data in multiple basins~~behaviors from observable catchment attributes. To achieve this, it is necessary to provide the network with information on the catchment characteristics that contain some amount of information that allows for discriminating between different catchments. Ideally, we want the network to condition the *processing of the dynamic inputs* on a set of static catchment characteristics. That is, we want the network to learn a mapping from meteorological time series into streamflow that itself (i.e., the mapping) depends on a set of static catchment characteristics that could, in principle, be measured anywhere in our modeling domain.

One way to do this would be to add the static features as additional inputs at every time step. That is, we could simply augment the vectors $\boldsymbol{x}[t]$ at every time step with a set of catchment characteristics that do not (necessarily) change over time. However, this approach does not allow us to directly inspect what the LSTM learns from these static catchment attributes.

Our proposal is therefore to use a slight variation on the normal LSTM architecture (an illustration is given in Fig. 1b):

$$\boldsymbol{i} = \sigma(\boldsymbol{W_i x_s} + \boldsymbol{b_i}) \tag{7}$$

$$\boldsymbol{f}[t] = \sigma(\boldsymbol{W_f x_d}[\boldsymbol{t}] + \boldsymbol{U_f h}[\boldsymbol{t-1}] + \boldsymbol{b_f}) \tag{8}$$

$$\boldsymbol{g}[t] = \tanh(\boldsymbol{W_g x_d}[\boldsymbol{t}] + \boldsymbol{U_g h}[\boldsymbol{t-1}] + \boldsymbol{b_g}) \tag{9}$$

$$\boldsymbol{o}[t] = \sigma(\boldsymbol{W_o x_d}[\boldsymbol{t}] + \boldsymbol{U_o h}[\boldsymbol{t-1}] + \boldsymbol{b_o}) \tag{10}$$

$$\boldsymbol{c}[t] = \boldsymbol{f}[t] \odot \boldsymbol{c}[t-1] + \boldsymbol{i} \odot \boldsymbol{g}[t] \tag{11}$$

$$\boldsymbol{h}[t] = \boldsymbol{o}[t] \odot \tanh(\boldsymbol{c}[\boldsymbol{t}]) \tag{12}$$

Here $\boldsymbol{i}$ is an input gate, which now does not change over time. $\boldsymbol{x}_s$ are the static inputs (e.g., catchment attributes) and $\boldsymbol{x}_d[t]$ are the dynamic inputs (e.g., meteorological forcings) at time step t ($1 \leq t \leq T$). The rest of the LSTM remains unchanged. The intuition is as follows: we explicitly process the static inputs $\boldsymbol{x}_s$ and the dynamic inputs $\boldsymbol{x}_d[t]$ separately within the architecture and assign them special tasks. The static features control, through input gate ($\boldsymbol{i}$), which parts of the LSTM are activated for any individual catchment, while the dynamic and recurrent inputs control what information is written into the memory ($\boldsymbol{g}[t]$), what is deleted ($\boldsymbol{f}[t]$), and what of the stored information to output ($\boldsymbol{o}[t]$) at the current time step t.

We are calling this an **Entity-Aware-LSTM (EA-LSTM)** because it explicitly differentiates between similar types of dynamical behaviors (here rainfall-runoff processes) that differ between individual entities (here different watersheds). After training, the static input gate of the **EA-LSTM** contains a series of real values in the range (0,1) that allow certain parts of the input gate to be active through the simulation of any individual catchment. In principle, different groups of catchments can share different parts of the full trained network.

This is an embedding layer, which allows for a non-naive information sharing between the catchments. For example, we could potentially discover, after training, that two particular catchments share certain parts of the activated network based on geological similarities while other parts of the network remain distinct due to ecological dissimilarities. This embedding layer allows for complex interactions between catchment characteristics, and - importantly - makes it possible for those interactions to be directly informed by the rainfall-runoff data from all catchments used for training.

2.3 Objective Function: A Smooth Joint NSE

An objective function is required for training the network. For regression tasks such as runoff prediction, the mean-squared-error (MSE) is commonly used. Hydrologists also sometimes use the NSE because it has an interpretable range of (-∞, 1). Both the MSE and NSE are squared error loss functions, with the difference being that the latter is normalized by the total variance of the observations. For single-basin optimization, the MSE and NSE will typically yield the same optimum parameter values, discounting any effects in the numerical optimizer that depend on the absolute magnitude of the loss value.

The linear relation between these two metrics (MSE and NSE) is lost, however, when calculated over data from multiple basins. In this case, the means and variances of the observation data are no longer constant because they differ between basins. We will exploit this fact. In our case, the MSE from a basin with low average discharge (e.g. smaller, arid basins) is generally smaller than the MSE from a basin with high average discharge (e.g. larger, humid basins). We need an objective function that does not depend on basin-specific mean discharge so that we do not overweight large humid basins (and thus perform poorly on small, arid basins). Our loss function is therefore the average of the NSE values calculated at each basin that supplies training data – referred to as basin averaged Nash-Sutcliffe Efficiency (NSE*). Additionally, we add a constant term to the denominator ($\epsilon = 0.1$), the variance of the observations, so that our loss function does not explode (to negative infinity) for catchments with very low flow-variance. Our loss function is therefore:

$$\text{NSE*} = \frac{1}{B} \sum_{b=1}^{B} \sum_{n=1}^{N} \frac{(\widehat{y}_n - y_n)^2}{(s(b) + \epsilon)^2}, \tag{13}$$

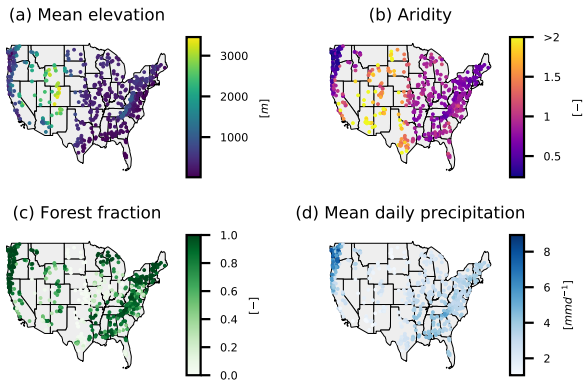


Figure 2. Overview of the basin location and corresponding catchment attributes. (a) The mean catchment elevation, (b) the catchment aridity (PET/P), (c) the fraction of the catchment covered by forest and (d) the daily average precipitation

where B is the number of basins, N is the number of samples (days) per basin B, $\hat{y}_n$ is the prediction of sample n ($1 \leq n \leq N$), y_n the observation and $s(b)$ is the standard deviation of the discharge in basin b ($1 \leq b \leq B$), calculated from the training period. In general, an entity-aware deep learning model will need a loss function that does not underweight entities with lower (relative to other entities in the training data set) absolute values in the target data.

2.4 The NCAR CAMELS Dataset

To benchmark our proposed EA-LSTM model, and to assess its ability to learn meaningful catchment similarities, we will use the Catchment Attributes and Meteorological (CAMELS) data set (Newman et al., 2014; Addor et al., 2017b). CAMELS is a set of data concerning 671 basins that is curated by the US National Center for Atmospheric Research (NCAR). The CAMELS basins range in size between 4 and 25 000 km^2, and were chosen because they have relatively low anthropogenic impacts. These catchments span a range of geologies and ecoclimatologies, as described in Newman et al. (2015) and Addor et al. (2017a).

We used the same subselection of 531 basins from the CAMELS data set ~~as Newman et al. (2017). The~~ that was used by Newman et al. (2017). These basins are mapped in Fig. 2. ~~These basins were chosen~~, and were chosen (by Newman et al. (2017)) out of the full set because some of the basins have a large (> 10 %) discrepancy between different strategies for calculating the basin area, and incorrect basin area would introduce significant uncertainty into a modeling study. ~~The basin selection and subset is described by Newman et al. (2017).~~ Furthermore, only basins with a catchment area smaller than 2000 km^2 were kept.

For time-dependent meteorological inputs ($\boldsymbol{x}_d[t]$), we used the daily, basin-averaged Maurer forcings (Wood et al., 2002) supplied with CAMELS. Our input data includes: (i) daily cumulative precipitation, (ii) daily minimum air temperature, (iii) daily maximum air temperature, (iv) average short-wave radiation and (v) vapor pressure. Furthermore, 27 CAMELS catchment characteristics were used as static input features ($\boldsymbol{x}_s$); these were chosen as a subset of the full set of characteristics explored

by Addor et al. (2017b) that are derivable from remote sensing or CONUS-wide available data products. These catchment attributes include climatic and vegetation indices, as well as soil and topographical properties (see Tab. A1 for an exhaustive list).

2.5 Benchmark models

The first part of this study benchmarks our proposed model against several high-quality benchmarks. The purpose of this exercise is to show that the EA-LSTM provides reasonable hydrological simulations.

To do this, we collected a set of existing hydrological models[1] that were configures, calibrated, and run by several previous studies over the CAMELS catchments. These models are: (i) SAC-SMA (Burnash et al., 1973; Burnash, 1995) coupled with the Snow-17 snow routine (Anderson, 1973), hereafter referred to as SAC-SMA, (ii) VIC (Liang et al., 1994), (iii) FUSE (Clark et al., 2008; Henn et al., 2008) (three different model structures, 900, 902, 904), (iv) HBV (Seibert and Vis, 2012) and (v) mHM (Samaniego et al., 2010; Kumar et al., 2013). In some cases, these models were calibrated to individual basins, and in other cases they were not. All of these benchmark models were run by other groups - we did not run any of our own benchmarks. We chose to use existing model runs so to not bias the calibration of the benchmarks to possibly favor ~~of~~ our own model. Each set of simulations that we used for benchmarking is documented elsewhere in the Hydrology literature (references below). Each of these benchmark models use the same daily Maurer forcings that we used with our EA-LSTM, and all were calibrated and validated on the same time period(s). These benchmark models can be distinguished into two different groups:

1. **Models calibrated for each basin individually**: These are SAC-SMA (Newman et al., 2017), VIC (Newman et al., 2017), FUSE[2], mHM (Mizukami et al., 2019) and HBV (Seibert et al., 2018). The HBV model supplied both a lower and upper benchmark, where the lower benchmark is an ensemble mean of 1000 uncalibrated HBV models and the upper benchmark is an ensemble of 100 calibrated HBV models.

2. **Models that were regionally calibrated**: These share one parameter set for all basins in the data set. Here we have calibrations of the VIC model (Mizukami et al., 2017) and mHM (Rakovec et al., 2019).

2.6 Experimental Setup

All model calibration and training was performed using data from the time period 1 October 1999 through 30 September 2008. All model and benchmark evaluation was done using data from the time period 1 Oct 1989 through 30 September 1999. We trained a single LSTM or EA-LSTM model using calibration period data from all basins, and evaluated this model using validation period data from all basins. This implies that a single parameter set (i.e. W, U, b from Eq. 1-4 and Eq. 7-10) was trained to work across all basins.

We trained and tested the following three model configurations:

[1] Will be released on HydroShare. DOI will be added for publication.

[2] The FUSE runs were generated by Nans Addor (n.addor@uea.ac.uk) and given to us by personal communication. These runs are part of current development by N. Addor on the FUSE model itself and might not reflect the final performance of the FUSE model.

– **LSTM without static inputs**: A single LSTM trained on the combined calibration data from all basins, using only the meteorological forcing data and ignoring static catchment attributes.

– **LSTM with static inputs**: A single LSTM trained on the combined calibration data of all basins, using the meteorological features as well as the static catchment attributes. These catchment descriptors were concatenated to the meteorological inputs at each time step.

– **EA-LSTM with static inputs**: A single EA-LSTM trained on the combined calibration data of all basins, using the meteorological features as well as the static catchment attributes. The catchment attributes were input to the static input gate in Eq. 7, while the meteorological inputs were used at all remaining parts of the network (Eq. 8-10).

All three model configurations were trained using the squared-error performance metrics discussed in Sect. 2.3 (MSE and NSE*). This resulted in six different model/training configurations.

To account for stochasticity in the network initialization and in the optimization procedure (we used stochastic gradient descent), all networks were trained with $n = 8$ different random seeds. Predictions from the different seeds were combined into an ensemble by taking the mean prediction at each timestep of all n different models under each configuration. In total, we trained and tested six different settings and eight different models per setting for a total of 48 different trained LSTM-type models. For all LSTMs we used the same architecture (apart from the inclusion of a static input gate in the EA-LSTM), which we found through hyperparameter optimization (see Appendix B for more details about the hyperparameter search). The LSTMs had 256 memory cells and a single fully connected layer with a dropout rate (Srivastava et al., 2014) of 0.4. The LSTMs were run in sequence-to-value mode (as opposed to sequence-to-sequence mode), so that to predict a single (daily) discharge value required meteorological forcings from 269 preceding days, as well as the forcing data of the target day, making the input sequences 270 time steps long.

2.6.1 Assessing Model Performance

Because no one evaluation metric can fully capture the consistency, reliability, accuracy, and precision of a streamflow model, it was necessary to use a variety of performance metrics for model benchmarking (Gupta et al., 1998). Evaluation metrics used to compare models are listed in Tab. 1. These metrics focus specifically on assessing the ability of the model to capture high-flows and low-flows, as well as assessing overall performance using a decomposition of the standard squared error metrics that is less sensitive to bias (Gupta et al., 2009).

2.6.2 Robustness and Feature Ranking

All catchment attributes used in this study are derived from gridded data products (Addor et al., 2017a). Taking the catchment's mean elevation as an example, we would get different mean elevations depending on the resolution of the gridded digital elevation model. More generally, there is uncertainty in all CAMELS catchment attributes. Thus, it is important that we evaluate the robustness of our model and of our embedding layer (particular values of the 256 static input gates) to changes in the exact

Table 1. Overview of used evaluation metrics. The notation of the original publications is kept.

Metric	Reference	Equation
Nash-Sutcliff-Efficiency (NSE)	Nash and Sutcliffe (1970)	$1 - \frac{\sum_{t=1}^{T}(Q_m[t] - Q_o[t])^2}{\sum_{t=1}^{T}(Q_o[t] - \bar{Q}_o)^2}$
α-NSE Decomposition	Gupta et al. (2009)	σ_s / σ_o
β-NSE Decomposition	Gupta et al. (2009)	$(\mu_s - \mu_o)/\sigma_o$
Top 2% peak flow bias (FHV)	Yilmaz et al. (2008)	$\frac{\sum_{h=1}^{H}(QS_h - QO_h)}{\sum_{h=1}^{H} QO_h} \times 100$
Bias of FDC midsegment slope (FMS)	Yilmaz et al. (2008)	$\frac{(\log(QS_{m1}) - \log(QS_{m2})) - (\log(QO_{m1}) - \log(QO_{m2}))}{(\log(QO_{m1}) - \log(QO_{m2}))} \times 100$
30% low flow bias (FLV)	Yilmaz et al. (2008)	$\frac{\sum_{l=1}^{L}(\log(QS_l) - \log(QS_L)) - \sum_{l=1}^{L}(\log(QO_l) - \log(QO_L))}{\sum_{l=1}^{L}(\log(QO_l) - \log(QO_L))} \times 100$

values of the catchment attributes. Additionally, we want some idea about the relative importance of different catchment attributes.

To estimate the robustness of the trained model to uncertainty in the catchment attributes, we added Gaussian noise $\mathcal{N}(0, \sigma)$ with increasing standard deviation to the individual attribute values and assessed resulting changes in model performance for each noise level. Concretely, additive noise was drawn from normal distributions with 10 different standard deviations: $\sigma = [0.1, 0.2, \ldots, 0.9, 1.0]$. All input features (both static and dynamic) were standardized (zero mean, unit variance) before training, so these perturbation sigmas did not depend on the units or relative magnitudes of the individual catchment attributes. For each basin and each standard deviation we drew 50 random noise vectors, resulting in $531 * 10 * 50 = 265500$ evaluations of each trained EA-LSTM.

To provide a simple estimate of the most important static features of the trained model, we used the method of Morris (Morris, 1991). Albeit the Morris method is relatively simple, it has been shown to provide meaningful estimations of the global sensitivity and is widely used (e.g., Herman et al., 2013; Wang and Solomatine, 2019). The method of Morris uses an approximation of local derivatives, which can be extracted directly from neural networks without additional computations, which makes this a highly efficient method of sensitivity analysis.

The method of Morris typically estimates feature sensitivities (EE_i) from local (numerical) derivatives.

$$\text{EE}_i = \frac{f(x_1, ..., x_i + \triangle_i, ..., x_p) - f(x)}{\triangle_i}, \tag{14}$$

Neural networks are completely differentiable (to allow for back-propagation) and thus it is possible to calculate the exact gradient with respect to the static input features. Thus, for neural networks the method of Morris can be applied analytically.

$$\text{EE}_i = \lim_{\triangle_i \to 0} \frac{f(x_1, ..., x_i + \triangle_i, ..., x_p) - f(x)}{\triangle_i} = \frac{\partial f(x)}{\partial x_i}, \tag{15}$$

This makes it unnecessary to run computationally expensive sampling methods to approximate the local gradient. Further, since we predict one time step of discharge at the time, we obtain this sensitivity measure for each static input for each day in the validation period. A global sensitivity measure for each basin and each feature is then derived from taking the average absolute gradient (Saltelli et al., 2004).

2.6.3 Analysis of Catchment Similarity from the Embedding Layer

Once the model is trained, the input gate vector ($\boldsymbol{i}$, see Eq. 7) for each catchment is fixed for the simulation period. This results in a vector that represents an embedding of the static catchment features (here in $\mathbb{R}^{27}$) into the high-dimensional space of the LSTM (here in $\mathbb{R}^{256}$). The result is a set of real-valued numbers that map the catchment characteristics onto a strength, or weight, associated with each particular cell state in the EA-LSTM. This weight controls how much of the cell input ($\boldsymbol{g}[t]$, see Eq. 9) is written into the corresponding cell state ($\boldsymbol{c}[t]$, see Eq. 11).

Per design, our hypothesis is that the EA-LSTM will learn to group similar basins together into the high-dimensional space, so that hydrologically-similar basins use similar parts of the LSTM cell states. This is dependent, of course, on the information content of the catchment attributes used as inputs, but the model should at least not degrade the quality of this information, and should learn hydrologic similarity in a way that is useful for rainfall-runoff prediction. We tested this hypothesis by analyzing the learned catchment embedding from a hydrological perspective. We analyzed geographical similarity by using k-means clustering on the $\mathbb{R}^{256}$ feature space of the input gate embedding to delineate basin groupings, and then plotted the clustering results geographically. The number of clusters was determined using a mean silhouette score.

In addition to visually analyzing the k-means clustering results by plotting them spatially (to ensure that the input embedding preserved expected geographical similarity), we measured the ability of these cluster groupings to explain variance in certain hydrological signatures in the CAMELS basins. For this, we used thirteen of the hydrologic signatures that were used by Addor et al. (2018): (i) mean annual discharge (q-mean), (ii) runoff ratio, (iii) slope of the flow duration curve (slope-fdc), (iv) baseflow index, (v) streamflow-precipitation elasticity (stream-elas), (vi) 5th percentile flow (q5), (vii) 95th percentile flow (q95), (viii) frequency of high flow days (high-q-freq), (ix) mean duration of high flow events (high-q-dur), (x) frequency of low flow days (low-q-freq), (xi) mean duration of low flow events (low-q-dur), (xii) zero flow frequency (zero-q-freq), and (xiii) average day of year when half of cumulative annual flow occurs (mean-hfd).

Finally, we reduced the dimension of the input gate embedding layer (from $\mathbb{R}^{256}$ to $\mathbb{R}^{2}$) so as to be able to visualize dominant features in the input embedding. To do this we ~~used UMAP (McInnes et al., 2018) , which is a nonparametric dimensionality reduction technique~~use a dimension reduction algorithm, called UMAP (McInnes et al., 2018) for "Uniform Manifold Approximation and Projection for Dimension Reduction". UMAP is based on neighbour graphs (while e.g., principle component analysis is based on matrix factorization), and it uses ideas from topological data analysis and manifold learning techniques to guarantee that information from the high dimensional space is preserved in the reduced space. For further details we refer the reader to the original publication by McInnes et al. (2018).

3 Results

This section is organized as follows:

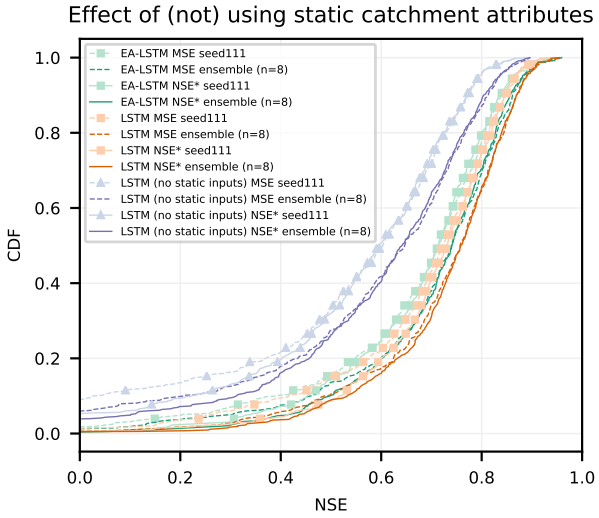


Figure 3. Cumulative density functions of the NSE for all LSTM-type model configurations described in Sect, 2.6.1. For each model type the ensemble mean and one of the $n = 8$ repetitions are shown. LSTM configurations are shown in orange (with catchment attributes) and purple (without catchment attributes), and the EA-LSTM configurations (always with catchment attributes) are shown in green.

- The first subsection (Sect. 3.1) presents a comparison between the three different LSTM-type model configurations discussed in Sect. 2.6.1. The emphasis in this comparison is to examine the effect of adding catchment attributes as additional inputs to the LSTM using the standard vs. adapted (EA-LSTM) architectures.
- The second subsection (Sect. 3.2) presents results from our benchmarking analysis – that is, the direct comparison between the performances of our EA-LSTM model with the full set of benchmark models outlined in Sect. 2.5.
- The third subsection (Sect. 3.3) present results of the sensitivity analysis outlined in Sect. 2.6.2.
- The final subsection (Sect. 3.4) presents an analysis of the EA-LSTM embedding layer to demonstrate that the model learned how to differentiate between different rainfall-runoff behaviors across different catchments.

3.1 Comparison between LSTM Modeling Approaches

The key results from a comparison between the LSTM approaches are in Fig. 3, which shows the cumulative density functions (CDF) of the basin-specific NSE values for all six LSTM models (three model configurations, and two loss functions) over the 531 basins.

Table 2 contains average key overall performance statistics. Statistical significance was evaluated using the paired Wilcoxon test (Wilcoxon, 1945), and the effect size was evaluated using Cohen's *d* (Cohen, 2013). The comparison contains four key results:

(i) Using catchment attributes as static input features improves overall model performance as compared with not providing the model with catchment attributes. This is expected, but worth confirming.

(ii) Training against the basin-average NSE* loss function improves overall model performance as compared with training against an MSE loss function, especially in the low NSE spectra.

(iii) There is statistically significant difference between the performance of the standard LSTM with static input features and the EA-LSTM however, with a small effect size.

(iv) Some of the error in the LSTM-type models is due to randomness in the training procedure and can be mitigated by running model ensembles.

Related to result (i), there was a significant difference between LSTMs with standard architecture trained with vs. without static features (square vs. triangle markers in Fig. 3). The mean (over basins) NSE improved in comparison with the LSTM that did not take catchment characteristics as inputs by 0.44 (range (0.38, 0.56)) when optimized using the MSE and 0.30 (range(0.22, 0.43)) when optimized using the basin-average NSE*. To assess statistical significance for single models, we first calculated the mean basin performance~~(e.g., i.e. the~~ mean NSE per basin ~~and across all seeds) between two different model settings was compared between different model configurations~~across the 8 repetitions. The so derived mean basin performance was then used for the test of significance. To assess statistical significance for ensemble means, the ~~mean basin performance of the ensemble mean was compared~~ ensemble prediction (i.e. the mean discharge prediction of the 8 model repetitions) was used to compare between different model ~~configurations~~approaches. For models trained using the standard MSE loss function, the p-value for the single model was $p = 1.2 * 10^{-75}$ and the p-value between the ensemble means was $p = 4 * 10^{-68}$. When optimized using the basin-average NSE*, the p-value for the single model was $p = 8.8 * 10^{-81}$) and the p-value between the ensemble means was $p = 3.3 * 10^{-75}$.

It is worth emphasizing that the improvement in overall model performance due to including catchment attributes implies that these attributes contain information that helps to distinguish different catchment-specific rainfall-runoff behaviors. This is especially interesting since these attributes are derived from remote sensing and other everywhere-available data products, as described by Addor et al. (2017b). Our benchmarking analysis presented in the next subsection (Sect. 3.2), shows that this information content is sufficient to perform high quality regional modeling (i.e., competitive with lumped models calibrated separately for each basin).

Related to result (ii), using the basin-average NSE* loss function instead of a standard MSE loss function improved performance for single models (different individual seeds) as well as for the ensemble means across all model configurations (see Tab. 2). The differences are most pronounced for the EA-LSTM and for the LSTM without static features. For the EA-LSTM, the mean NSE for the single model raised from 0.63 when optimized with MSE to 0.67 when optimized with the basin average NSE*. For the LSTM trained without catchment characteristics the mean NSE went from 0.23 when optimized with MSE to 0.39 when optimized with NSE*. Further, the median NSE did not change significantly depending on loss function due to the fact that the improvements from using the NSE* are mostly to performance in basins at the lower-end of the NSE spectra (see

Table 2. Evaluation results of the single models and ensemble means.

Model	NSE[i]		No. of basins
	mean	median	with NSE ≤ 0
LSTM w/o static inputs			
using MSE:			
Single model:	0.24 ($\pm$ 0.049)	0.60 ($\pm$ 0.005)	44 ($\pm$ 4)
Ensemble mean (n=8):	0.36	0.65	31
using NSE*:			
Single model:	0.39 ($\pm$ 0.059)	0.59 ($\pm$ 0.008)	28 ($\pm$ 3)
Ensemble mean (n=8):	0.49	0.64	20
LSTM with static inputs			
using MSE:			
Single model:	0.66 ($\pm$ 0.012)	0.73 ($\pm$ 0.003)	6 ($\pm$ 2)
Ensemble mean (n=8):	0.71	0.76	3
using NSE*:			
Single model:	0.69 ($\pm$ 0.013)	0.73 ($\pm$ 0.002)	2 ($\pm$ 1)
Ensemble mean (n=8):	0.72	0.76	2
EA-LSTM			
using MSE:			
Single model:	0.63 ($\pm$ 0.018)	0.71 ($\pm$ 0.005)	9 ($\pm$ 1)
Ensemble mean (n=8):	0.68	0.74	6
using NSE*:			
Single model:	0.67 ($\pm$ 0.006)	0.71 ($\pm$ 0.005)	3 ($\pm$ 1)
Ensemble mean (n=8):	0.70	0.74	2

[i]: *Nash-Sutcliffe efficiency:* $(-\infty, 1]$*, values closer to one are desirable.*

also Figure 1 dashed vs. solid lines). This is as expected as catchments with relatively low average flows have a small influence on LSTM training with an MSE loss function, which results in poor performance in these basins. Using the NSE* loss function helps to mitigate this problem. It is important to note that this is not the only reason why certain catchments have low skill scores, which can happen for a variety of reasons with any type of hydrological model (e.g., bad input data, unique catchment behaviors, etc.). This improvement at the low-performance end of the spectrum can also been seen by looking at the number of 'catastrophic failures', i.e., basins with an NSE value of less than zero. Across all models we see a reduction in this number when optimized with the basin average NSE*, compared to optimizing with MSE.

Related to result (iii), Fig. 3 shows a small difference in the empirical CDFs between the standard LSTM with static input features and the EA-LSTM under both functions (compare green vs orange lines) . The difference is significant (p-value for

single model $p = 1 * 10^{-28}$, p-value for the ensemble mean $p = 2.1 * 10^{-26}$, paired Wilcoxon test), however the effect size is small $d = 0.055$. This is important because the embedding layer in the EA-LSTM adds a layer of interpretability to the LSTM, which we argue is desirable for scientific modeling in general, and is useful in our case for understanding catchment similarity. This is only useful, however, if the EA-LSTM does not sacrifice performance compared to the less interpretable traditional LSTM. There is some small performance sacrifice in this case, likely due to an increase in the number of tunable parameters in the network, but the benefit of this small reduction in performance is explainability.

Related to results (iv), in all cases there were several basins with very low NSE values (this is also true for the benchmark models, which we will discuss in Sect. 3.2). Using catchment characteristics as static input features with the EA-LSTM architecture reduced the number of such basins from 44 (31) to 9 (6) for the average single model (ensemble mean) when optimized with the MSE, and from 28 (20) to 3 (2) for the average single model (ensemble mean) if optimized using the basin-average NSE*. This result is worth emphasizing: each LSTM or EA-LSTM trained over all basins results in a certain number of basins that perform poorly (NSE ≤ 0), but the basins where this happens are not always the same. The model outputs, and therefore the number of catastrophic failures, differ depending on the randomness in the weight initialization and optimization procedure and thus, running an ensemble of LSTMs substantively reduces this effect. This is good news for deep learning - it means that at least a portion of uncertainty can be mitigated using model ensembles. We leave as an open question for future research how many ensemble members, as well as how these are initialized, should be used to minimize uncertainty for a given data set.

3.2 Model Benchmarking: ~~Comparison with Traditional~~ EA-LSTM vs. Calibrated Hydrology Models

The results in this section are calculated from 447 basins that were modeled by all benchmark models, as well as our ~~LSTMs~~ EA-LSTMs. In this section, we concentrate on benchmarking the EA-LSTM, however for the sake of completeness, we added the results of the LSTM with static inputs to all figures and tables.

First we compared the EA-LSTM against the two hydrological models that were regionally calibrated (VIC and mHM). Specifically, what was calibrated for each model was a single set of transfer functions that map from static catchment characteristics to model parameters. The procedure for parameterizing these models for regional simulations is described in detail by the original authors: Mizukami et al. (2017) for VIC and Rakovec et al. (2019) for mHM. Figure 4 shows that the EA-LSTM outperformed both regionally-calibrated benchmark models by a large margin. Even the LSTM trained without static catchment attributes (only trained on meteorological forcing data) outperformed both regionally calibrated models consistently as a single model, and even more so as an ensemble.

The mean and median NSE scores across the basins of the individual EA-LSTM models ($N_{ensemble} = 8$) were 0.67 $\pm$ 0.006 (0.71) and 0.71 $\pm$ 0.004 (0.74) respectively. In contrast, VIC had a mean NSE of 0.17 and a median NSE of 0.31 and the mHM had a mean NSE of 0.44 and median NSE of 0.53. Overall, VIC scored higher than the EA-LSTM ensemble in 2 out of 447 basins (0.4%) and mHM scored higher than the EA-LSTM ensemble in 16 basins (3.58%). Investigating the number of catastrophic failures (the number of basins where NSE ≤ 0), the average single EA-LSTM failed in approximately 2 basins out of 447 basins (0.4 $\pm$ 0.2%) and the ensemble mean of the EA-LSTM failed in only a single basin (i.e., 0.2%). In comparison mHM failed in 29 basins (6.49%) and VIC failed in 41 basins (9.17%).

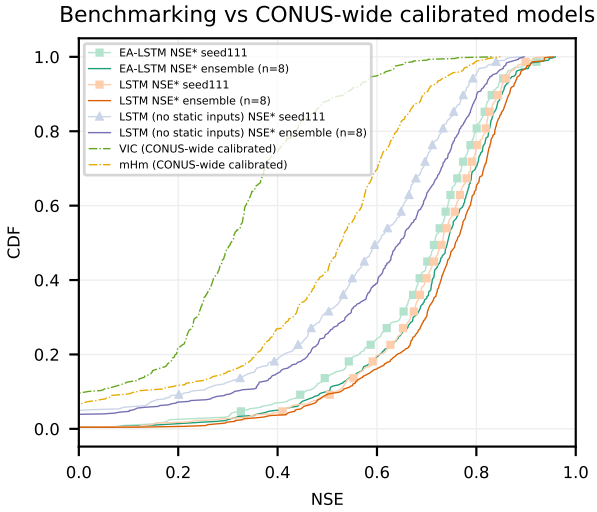


Figure 4. Cumulative density functions of the NSE of two regionally-calibrated benchmark models (VIC and mHM), compared to the EA-LSTM and the LSTM trained with and without static input features.

Second, we compared our multi-basin calibrated EA-LSTMs to individual-basin calibrated hydrological models. This is a more rigorous benchmark than the regionally calibrated models, since hydrological models usually perform better when trained for specific basins. Figure 5 compares CDFs of the basin-specific NSE values for all benchmark models over the 447 basins. Table 3 contains the performance statistics for these benchmark models as well as for the re-calculated EA-LSTM.

The main benchmarking result is that the EA-LSTM significantly outperforms all benchmark models in the overall NSE. The two best performing hydrological models were the ensemble ($n = 100$) of basin-calibrated HBV models and a single basin-calibrated mHM model. The EA-LSTM out-performed both of these models at any reasonable alpha level. The p-value for the single model, compared to the HBV upper bound was $p = 1.9 * 10^{-4}$ and for the ensemble mean $p = 6.2 * 10^{-11}$ with a medium effect size (Cohen's d for single model $d = 0.22$ and for the ensemble mean $d = 0.40$). The p-value for the single model, compared to the basin-wise calibrated mHM was $p = 4.3 * 10^{-6}$ and for the ensemble mean $p = 1.0 * 10^{-13}$ with a medium effect size (Cohen's d for single model $d = 0.26$ and for the ensemble mean $d = 0.45$).

Regarding all other metrics except the Kling-Gupta decomposition of the NSE, there was no statistically significant difference between the EA-LSTM and the two best performing hydrological models. The β-decomposition of the NSE measures a scaled difference in simulated vs. observed mean streamflow values, and in this case the HBV benchmark performed better that the EA-LSTM, with an average scaled absolute bias (normalized by the root-variance of observations) of -0.01, where as the EA-LSTM had an average scaled bias of -0.03 for the individual model as well as for the ensemble ($p = 3.5 * 10^{-4}$).

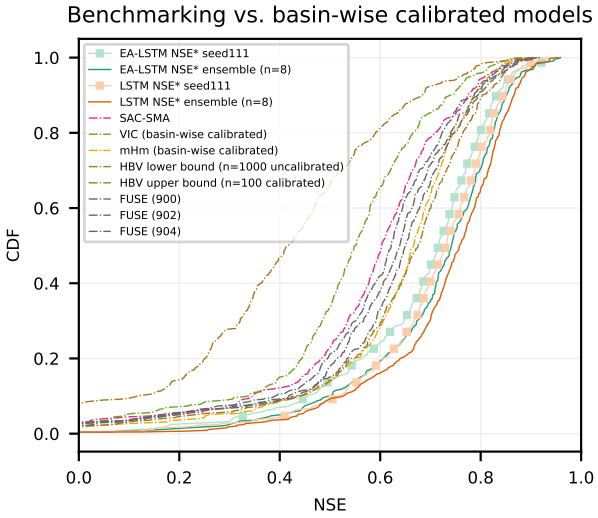


Figure 5. Cumulative density function of the NSE for all basin-wise calibrated benchmark models compared to the EA-LSTM and the LSTM with static input features.

3.3 Robustness and Feature Ranking

In Sect. 3.1, we found that adding static features provided a large boost in performance. We would like to check that the model is not simply 'remembering' each basin instead of learning a general relation between static features and catchment-specific hydrologic behavior. To this end, we examined model robustness with respect to noisy perturbations of the catchment attributes. Figure 6 shows the results of this experiment by comparing the model performance when forced (not trained) with perturbed static features in each catchment against model performance using the same static feature values that were used for training. As expected, the model performance degrades with increasing noise in the static inputs. However, the degradation does not happen abruptly but smoothly with increasing levels of noise, which is an indication that the LSTM is not over-fitting on the static catchment attributes. That is, it is not remembering each basin with its set of attributes exactly, but rather learns a smooth mapping between attributes and model output. To reiterate from Sect 2.6.2, the perturbation noise is always relative to the overall standard deviation of the static features across all catchments, which is always $\sigma = 1$ (i.e., all static input features were normalized prior to training). When noise with small standard deviation was added (e.g. $\sigma = 0.1$ and $\sigma = 0.2$) the mean and median NSEs were relatively stable. The median NSE decreased from 0.71 without noise to 0.48 with an added noise equal to the total variance of the input features ($\sigma = 1$). This is roughly similar to the performance of the LSTM without static input features (Tab. 2). In contrast, the lower percentiles of the NSE distributions were more strongly affected by input noise. For example, the 1st (5th) percentile of the NSE values decreased from an NSE of 0.13 (0.34) to -5.87 (-0.94) when going from zero noise (the catchment attributes data from CAMELS) to additive noise with variance equal to the total variance of the inputs

Table 3. Comparison of the EA-LSTM and LSTM (with static inputs) average single model and ensemble mean to the full set of benchmark models. VIC (basin) and mHM (basin) denote the basin-wise calibrated models, while VIC (CONUS) and mHM (CONUS) denote the CONUS-wide calibrated models. HBV (lower) denotes the ensemble mean of $n = 1000$ uncalibrated HBVs, while HBV (upper) denotes the ensemble mean of $n = 100$ calibrated HBVs (for details see Seibert et al. (2018)). For the FUSE model, the numbers behind the name denote different FUSE model structures. All statistics were calculated from the validation period of all 447 commonly modeled basins.

Model	NSE[i]		No. of basins	α-NSE[ii]	β-NSE[iii]	FHV[iv]	FMS[v]	FLV[vi]
	mean	median	with NSE ≤ 0	median	median	median	median	median
EA-LSTM Single	0.674	0.714	2	0.82	-0.03	-16.9	-10.0	2.0
	($\pm$ 0.006)	($\pm$ 0.004)	($\pm$ 1)	($\pm$ 0.013)	($\pm$ 0.009)	($\pm$ 1.1)	($\pm$ 1.7)	($\pm$ 7.6)
EA-LSTM Ensemble	0.705	0.742	1	0.81	-0.03	-18.1	-11.3	31.9
LSTM Single	0.685	0.731	1	0.85	-0.03	-14.8	-8.3	26.5
	($\pm$ 0.015)	($\pm$ 0.002)	($\pm$ 0)	($\pm$ 0.011)	($\pm$ 0.007)	($\pm$ 1.1)	($\pm$ 1.2)	($\pm$ 7.6)
LSTM Ensemble	0.718	0.758	1	0.84	-0.03	-15.7	-8.8	55.1
SAC-SMA	0.564	0.603	13	0.78	-0.07	-20.4	-14.3	37.3
VIC (basin)	0.518	0.551	10	0.72	-0.02	-28.1	-6.6	-70.0
VIC (CONUS)	0.167	0.307	41	0.46	-0.07	-56.5	-28.0	17.4
mHM (basin)	0.627	0.666	7	0.81	-0.04	-18.6	-7.2	11.4
mHM (CONUS)	0.442	0.527	29	0.59	-0.04	-40.2	-30.4	36.4
HBV (lower)	0.237	0.416	35	0.58	-0.02	-41.9	-15.9	23.9
HBV (upper)	0.631	0.676	9	0.79	-0.01	-18.5	-24.9	18.3
FUSE (900)	0.587	0.639	12	0.80	-0.03	-18.9	-5.1	-11.4
FUSE (902)	0.611	0.650	10	0.80	-0.05	-19.4	9.6	-33.2
FUSE (904)	0.582	0.622	9	0.78	-0.07	-21.4	15.5	-66.7

[i]: *Nash-Sutcliffe efficiency:* $(-\infty, 1]$*, values closer to one are desirable.*
[ii]: *α-NSE decomposition:* $(0, \infty)$*, values close to one are desirable.*
[iii]: *β-NSE decomposition:* $(-\infty, \infty)$*, values close to zero are desirable.*
[iv]: *Top 2 % peak flow bias:* $(-\infty, \infty)$*, values close to zero are desirable.*
[v]: *Bias of FDC midsegment slope:* $(-\infty, \infty)$*, values close to zero are desirable.*
[vi]: *30 % low flow bias:* $(-\infty, \infty)$*, values close to zero are desirable.*

(i.e., $\sigma = 1$). This reinforces that static features are especially helpful for increasing performance in basins at the lower end of the NSE spectrum - that is, differentiating hydrological behaviors that are under-represented in the training data set.

Figure 7 plots a spatial map where each basin is labeled corresponding to the most sensitive catchment attribute derived from the explicit Morris method for neural networks (Sect. 2.6.2). In the Appilachain Mountains, sensitivity in most catchments is dominated by topological features (e.g., mean catchment elevation and catchment area), and in the Eastern US more generally, sensitivity is dominated by climate indices (e.g., mean precipitation, high precipitation duration). Meteorological patterns like aridity and mean precipitation become more important as we move away from the Appalachians and towards the Great Plains, likely because elevation and slope begin to play less of a role. The aridity index dominates sensitivity in the Central Great

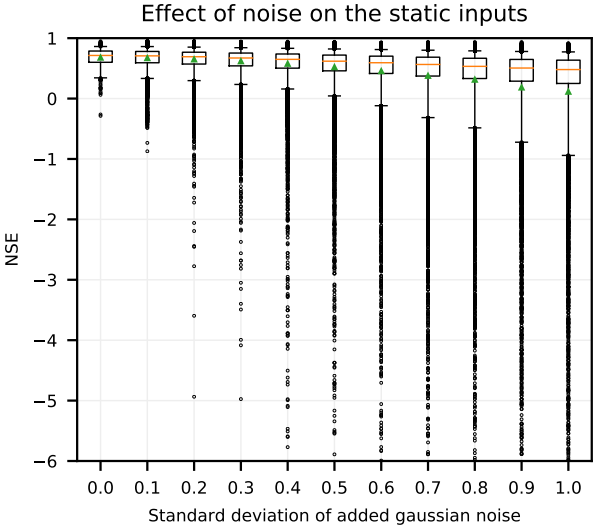


Figure 6. Boxplot showing degradation of model performance with increasing noise level added to the catchment attributes. Orange lines denote the median across catchments, green markers represent means across catchments, box denote the 25 and 75-percentiles, whiskers denote the 5 and 95-percentiles, and circles are catchments that fall outside the 5-95-percentile range.

Plains. In the Rocky Mountains most basins are sensitive climate indices (mean precipitation and high precipitation duration), with some sensitivity to vegetation in the Four-Corners region (northern New Mexico). In the West Coast there is a wider variety of dominant sensitivities reflecting a diversity of catchments.

Table 4 provides an overall ranking of dominant sensitivities for one of the 8 model repetitions of the EA-LSTM. These were derived by normalizing the sensitivity measures per basin to the range (0,1) and then calculating the overall mean across all features. As might be inferred from Fig. 7 the most sensitive catchment attributes are topological features (mean elevation and catchment area) and climate indices (mean precipitation, aridity, duration of high precipitation events and the fraction of precipitation falling as snow). Certain groups of catchment attributes did not typically provide much additional information. These include vegetation indices like maximum leaf area index or maximum green vegetation fraction, as well as the annual vegetation differences. Most soil features were at the lower end of the feature ranking. This sensitivity ranking is interesting in that most of the top-ranked features are relatively easy to measure or estimate globally from readily-available gridded data products. Soil maps are one of the hardest features to obtain accurately at a regional scale because they require extensive in situ mapping and interpolation. Note that the results between the 8 model repetitions (not shown here) vary slightly in terms of sensitivity values and ranks. However, the quantitative ranking is robust between all 8 repetitions, meaning that climate indices (e.g. aridity and mean precipitation) and topological features (e.g. catchment area and mean catchment elevation) are always ranked highest, while soil and vegetation features are of less importance and are ranked lower. It is worth noting that our rankings qualitatively agree with much of the analysis by Addor et al. (2018).

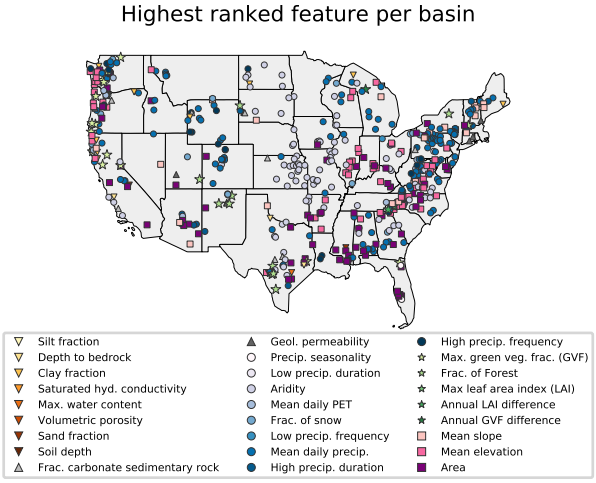


Figure 7. Spatial map of all basins in the data set. Markers denote the individual catchment characteristic with the highest sensitivity value for each particular basin.

3.4 Analysis of Catchment Similarity from the Embedding Layer

Kratzert et al. (2018a, 2019) showed that these LSTM networks are able to learn to model snow, and store this information in specific memory cells, without ever directly training on any type of snow-related observation data other than total precipitation and temperature. Multiple types of catchments will use snow-related states in mixture with other states that represent other processes or combinations of processes. The memory cells allow an interpretation along the time axis for each specific basin, and are part of both the standard LSTM and the EA-LSTM. A more detailed analysis of the specific functionality of individual cell states is out-of-scope for this manuscript and will be part of future work. Here, we focus on analysis of the embedding layer, which is a unique feature of the EA-LSTM.

From each of the trained EA-LSTM models, we calculated the input gate vector (Eq. 7) for each basin. The raw EA-LSTM embedding from one of the models trained over all catchments is shown in Fig. 8. Yellow colors indicate that a particular one of the 256 cell states is activated and contributes to the simulation of a particular catchment. Blue colors indicate that a particular cell state is not used for a particular catchment. These (real-valued) activations are a function of the 27 catchment characteristics input into the static feature layer of the EA-LSTM.

The embedding layer is necessarily high-dimensional – in this case $\mathbb{R}^{256}$ – due to the fact that the LSTM layer of the model requires sufficient cell states to simulate a wide variety of catchments. Ideally, hydrologically-similar catchments should utilize overlapping parts of the LSTM network - this would mean that the network is both learning and using catchment similarity to train a regionalizable simulation model.

To assess whether this happened, we first performed a clustering analysis on the $\mathbb{R}^{256}$ embedding space using k-means with an Euclidean distance criterion. We compared this with a k-means clustering analysis using directly the 27 catchment charac-

Table 4. Feature ranking derived from the explicit Morris method for one of the EA-LSTM model repetitions.

Rank	Catchment characteristic	Sensitivity
1.	Mean precipitation	0.68
2.	Aridity	0.56
3.	Area	0.50
4.	Mean elevation	0.46
5.	High precip. duration	0.41
6.	Fraction of snow	0.41
7.	High precip. frequency	0.38
8.	Mean slope	0.37
9.	Geological permeability	0.35
10.	Frac. carbonate sedimentary rock	0.34
11.	Clay fraction	0.33
12.	Mean PET	0.31
13.	Low precip. frequency	0.30
14.	Soil depth to bedrock	0.27
15.	Precip. seasonality	0.27
16.	Frac. of Forest	0.27
17.	Sand fraction	0.26
18.	Saturated hyd. conductivity	0.24
19.	Low precip. duration	0.22
20.	Max. green veg. frac. (GVF)	0.21
21.	Annual GVF diff.	0.21
22.	Annual leaf area index (LAI) diff.	0.21
23.	Volumetric porosity	0.19
24.	Soil depth	0.19
25.	Max. LAI	0.19
26.	Silt fraction	0.18
27.	Max. water content	0.16

teristics, to see if there was a difference in clusters before vs. after the transformation into the embedding layer - remember that this transform was informed by rainfall-runoff training data. To choose an appropriate cluster size, we looked at the mean (and minimum) silhouette value. Silhouette values measure within-cluster similarity and range between [-1,1], with positive values indicating a high degree of separation between clusters, and negative values indicating a low degree of separation between clusters. The mean and minimum silhouette values for different cluster sizes are shown in Fig. 9. In all cases with cluster

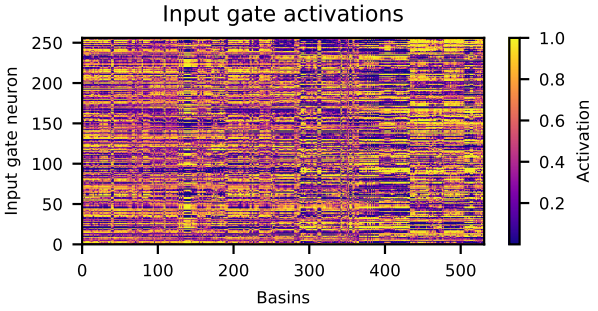


Figure 8. Input gate activations (y-axis) for all 531 basins (x-axis). The basins are ordered from left to right according to the ascending 8-digit USGS gauge ID. Yellow colors denote open input gate cells, blue colors denote closed input gate cells for a particular basin.

sizes less than 15, we see that clustering by the values of the embedding layer provides more distinct catchment clusters than when clustering by the raw catchment attributes. This indicates that the EA-LSTM is able to use catchment attribute data to effectively cluster basins into distinct groups.

The highest mean silhouette value from clustering with the raw catchment attributes was $k = 6$ and the highest mean silhouette value from clustering with the embedding layer was $k = 5$. Ideally, these clusters would be related to hydrologic behavior. To test this, Fig. 10 shows the fractional reduction in variance of 13 hydrologic signatures due to clustering by both raw catchment attributes vs. by the EA-LSTM embedding layer. Ideally, the within-cluster variance of any particular hydrological signature should be as small as possible, so that the fractional reduction in variance is as large (close to one) as possible. In both the $k = 5$ and $k = 6$ cluster examples, clustering by the EA-LSTM embedding layer reduced variance in the hydrological signatures by more or approximately the same amount as by clustering on the raw catchment attributes. The exception to this was the hfd-mean date, which represents an annual timing process (i.e., the day of year when the catchment releases half of its annual flow). This indicates that the EA-LSTM embedding layer is largely preserving the information content about hydrological behaviors, while overall increasing distinctions between groups of similar catchments. The EA-LSTM was able to learn about hydrologic similarity between catchments by directly training on both catchment attributes and rainfall-runoff time series data. Remember that the EA-LSTMs were trained on the time series of streamflow data that these signatures were calculated from, but were not trained directly on these hydrologic signatures.

Clustering maps for $k = 5$ and $k = 6$ are shown in Fig. 11. Although latitude and longitude were not part of the catchment attributes vector that was used as input into the embedding layer, both the raw catchment attributes and the embedding layer clearly delineated catchments that correspond to different geographical regions within the CONUS.

To visualize the high-dimensional embedding learned by the EA-LSTM, we used UMAP (McInnes et al., 2018) to project the full $\mathbb{R}^{256}$ embedding onto $\mathbb{R}^2$. Figure 12 shows results of the UMAP transformation for one of the eight EA-LSTMs. In each subplot in Fig. 12, each point corresponds to one basin. The absolute values of the transformed embedding are not of particular interest, but we are interested in the relative arrangement of the basins in this 2-dimensional space. Because this is

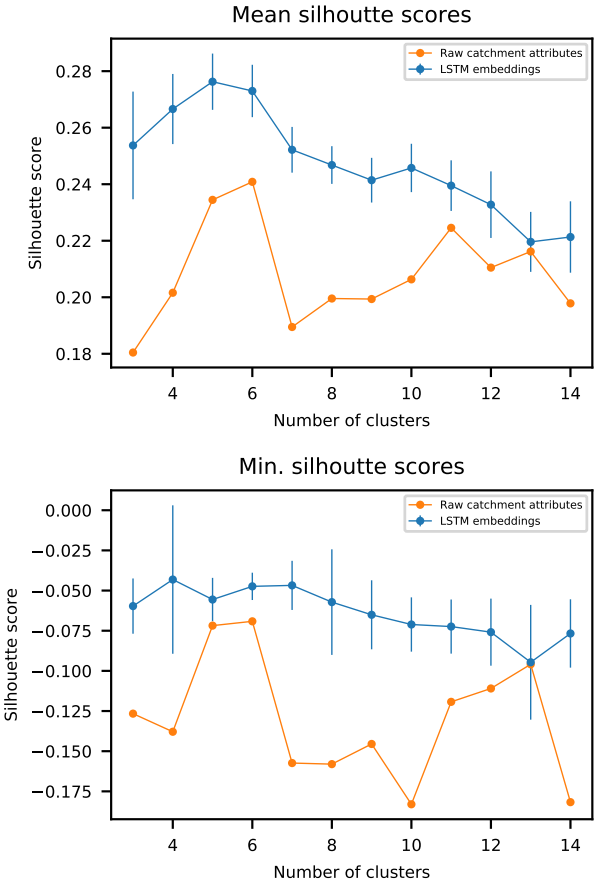


Figure 9. Mean and minimum silhouette scores over varying cluster sizes. For the LSTM embeddings, the line denotes the mean of the $n = 8$ repetitions and the vertical lines the standard deviation over 10 random restarts of the k-means clustering algorithm.

a reduced-dimension transformation, the fact that there are ~~three~~ four clear clusters of basins does not necessarily indicate that these are the only distinct basin clusters in our 256-dimensional embedding layer (as we saw above). Figure 12 shows that there is strong interaction between the different catchment characteristics in this embedding layer. For example, high-elevation dry catchments with low forest cover are in the same cluster as low-elevation wet catchments with high forest cover (see cluster B in Fig. 12). These two groups of catchments share parts of their network functionality in the LSTM, whereas highly seasonal catchments activate a different part of the network. Additionally, there are two groups of basins with a high forest fractions (cluster A and B), however if we also consider the mean annual green vegetation difference, both of these clusters are quite distinct. The cluster A in the upper left of each subplot in Fig. 12 contains forest type basins with a high annual variation in the green vegetation fraction (possibly deciduous forests) and the cluster B at the right has almost no annual variation (possibly coniferous forests). One feature that does not appear to affect catchment groupings (i.e., apparently acts independent

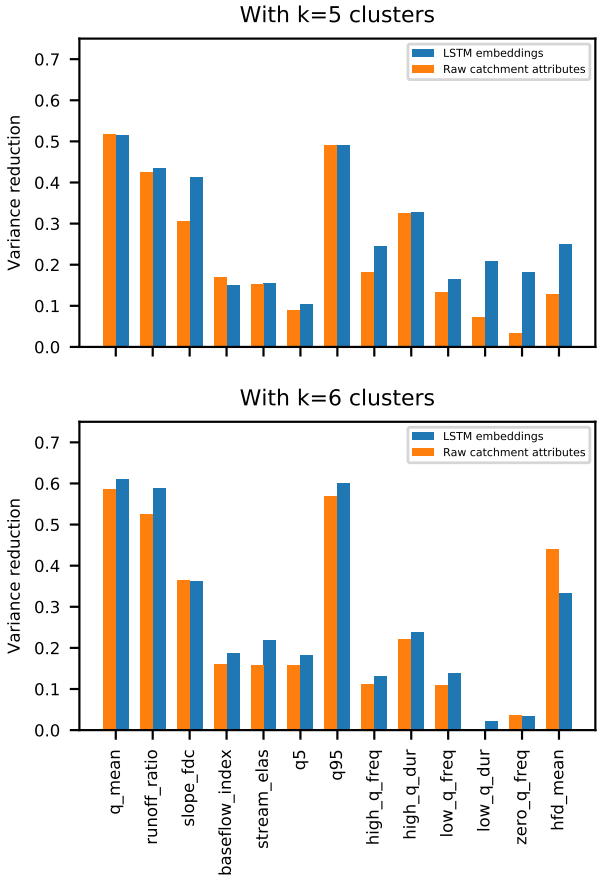


Figure 10. Fractional reduction in variance about different hydrological signatures due to k-means clustering on catchment attributes vs. the EA-LSTM embedding layer.

of other catchment characteristics) is basin size – large and small basins are distributed throughout the three UMAP clusters. To summarize, this analysis demonstrates that the EA-LSTM is able to learn complex interactions between catchment attributes that allows for grouping different basins (i.e., choosing which cell states in the LSTM any particular basin or group of basins will use) in ways that account for interaction between different catchment attributes.

4 Discussion and Conclusion

The EA-LSTM is an example of what Razavi and Coulibaly (2013) called a *model-independent* method for regional modeling. We cited Besaw et al. (2010) as an earlier example this type of approach, since they ~~also~~ used classical feed-forward neural networks. In our case, the EA-LSTM achieved state-of-the-art results, outperforming multiple locally- and regionally-calibrated benchmark models. These benchmarking results are arguably ~~the critical result~~ a pivotal part of this paper.

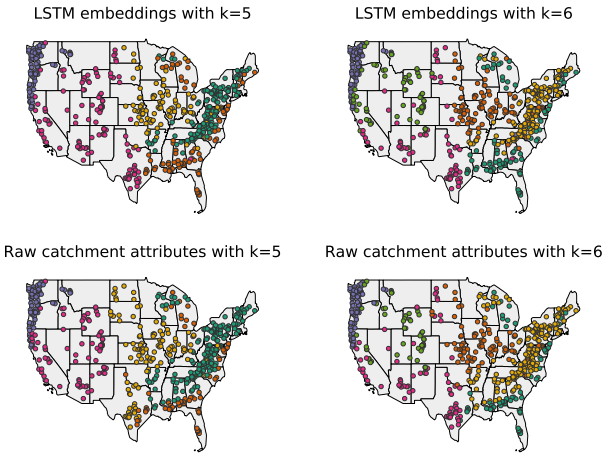


Figure 11. Clustering maps for the LSTM embeddings (top row) and the raw catchment attributes (bottom row) using $k = 5$ clusters (left column, optimal choice for LSTM embeddings) and $k = 6$ clusters (right column, optimal choice for the raw catchment attributes)

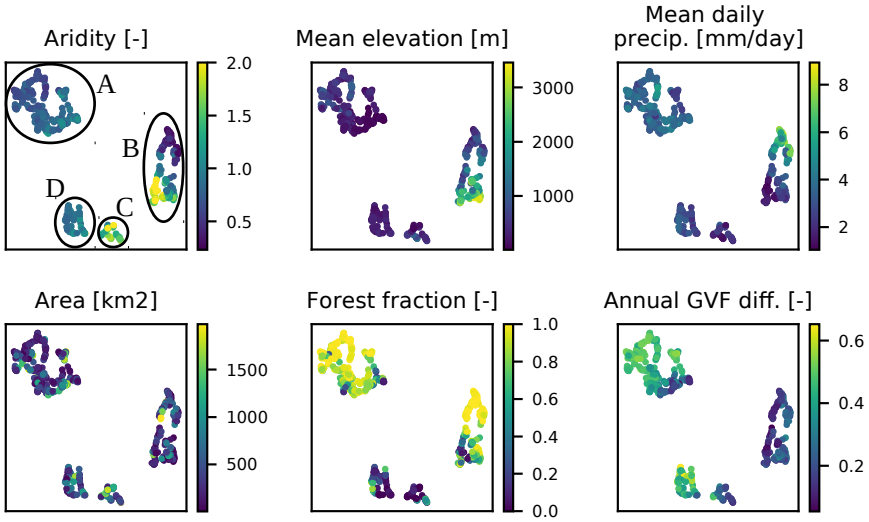


Figure 12. UMAP transformation of the $\mathbb{R}^{256}$ EA-LSTM catchment embedding onto $\mathbb{R}^2$. Each dot in each subplot corresponds to one basin. The colors denote specific catchment attributes (notated in subplot titles) for each particular basin. In the upper left plot, clusters are encircled and named to facilitate the description in the text.

The results of the experiments described above demonstrate that a single 'universal' deep learning model can learn both regionally-consistent and location-specific hydrologic behaviors. The innovation in this study – besides benchmarking the LSTM family of rainfall-runoff models – was to add a static embedding layer in the form of our EA-LSTM. This model

offered similar performance as compared with a conventional LSTM (Sect. 3.1) but offers a level of interpretabiltiy about how the model learns to differentiate aspects of complex catchment-specific behaviors (Sect. 3.3 and Sect. 3.4). In a certain sense, this is similar to the aforementioned MPR approach, which links its model parameters to the given spatial characteristics (in a non-linear way, by using transfer functions), but has a fixed model structure to work with. In comparison, our EA-LSTM links catchment characteristics to the dynamics of specific sites and learns the overall model from the combined data of all catchments. Again, the critical take-away, in our opinion, is that the EA-LSTM learns a single model from large catchment data sets in a way that explicitly incorporates local (catchment) similarities and differences.

Neural networks generally require a lot of training data (our unpublished results indicate that it is often difficult to reliably train an LSTM at a single catchment, even with multi-decade data records), and adding the ability for the LSTM architecture to transfer information from similar catchments is critical for this to be a viable approach for regional modeling. This is in contrast with traditional hydrological modeling and model calibration, which typically has the best results when models are calibrated independently for each basin. This property of classical models is somewhat problematic, since it has been observed that the spatial patterns of model parameter obtained by ad-hoc extrapolations based on calibrated parameters from reference catchments can lead to unrealistic parameter fields and spatial discontinuities of the hydrological states (Mizukami et al., 2017). As shown in Sect. 3.4 this does not occur with our proposed approach. Thus, by leveraging the ability of deep learning to simultaneously learn time series relationships and also spatial relationships in the same predictive framework, we sidestep many problems that are currently associated with the estimation and transfer of hydrologic model parameters.

Moving forward, ~~however,~~ it is worth mentioning that treating catchment attributes as static is a strong assumption (especially over long time periods), which is not necessarily reflected in the real world. In reality, catchment attributes may continually change at various timescales (e.g., vegetation, topography, pedology, climate). In future studies it will be important to develop strategies to derive analogues to our embedding layer that allow for dynamic or evolving catchment attributes or features - perhaps that act on raw remote sensing data inputs rather than aggregated indexes derived from time series of remote sensing products. In principle, our embedding layer could learn directly from raw brightness temperatures, since there is no requirement that the inputs be hydrologically relevant - only that these inputs are related to hydrological behavior. A dynamic input gate is, at least in principle, possible without significant modification to the proposed EA-LSTM approach. For example, by using a separate sequence-to-sequence LSTM that encodes time-dependent catchment observables (e.g., from climate models or remote sensing) and feeds an embedding layer that is updated at each timestep. This would allow the model to 'learn' a dynamic embedding that turns off and on different parts of the rainfall-runoff portion of the LSTM over the course of a simulation.

A notable corollary of our main result is that the catchment attributes collected by Addor et al. (2017b) appear to contain sufficient information to distinguish between diverse rainfall-runoff behaviors, at least to a meaningful degree. It is arguable whether this was known previously, since regional modeling studies have largely struggled to fully extract this information (Mizukami et al., 2017) - i.e., existing regional models do not perform with accuracy similarly to models calibrated in a specific catchment. In contrast, our regional EA-LSTM actually performs better than models calibrated separately for individual catchments. This result ~~goes somewhat against the commonly-held belief~~ challenges the idea that runoff time series alone only ~~bear~~ contain enough information to restrict a handful of parameters (Naef, 1981; Jakeman and Hornberger, 1993; Perrin et al.,

2001; Kirchner, 2006), and implies that structural improvements are still possible for most large-scale hydrology models, given the size of today's data sets.

Code and data availability. The CAMELS input data is freely available at the homepage of NCAR. The validation period of all benchmark models used in this study is available at DOI. The code to reproduce the results of this manuscript can be found at https://github.com/kratzert/ealstm_regional_modeling

Author contributions. FK had the idea for for the regional modeling approach. SH proposed the adapted LSTM architecture. FK, DK and GN designed all experiments. FK conducted all experiments and analysed the results, together with DK, GS and GN. GN supervised the manuscript from the hydrological perspective and GK and SH from the machine learning perspective. GN and SH share the responsibility for the last-authorship in the respective fields. All authors worked on the manuscript.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. The project relies heavily on open source software. All programming was done in Python version 3.7 (van Rossum, 1995) and associated libraries including: Numpy (Van Der Walt et al., 2011), Pandas (McKinney, 2010), PyTorch (Paszke et al., 2017) and Matplotlib (Hunter, 2007)

This work was supported by Bosch, ZF, and Google. We thank the NVIDIA Corporation for the GPU donations, LIT with grant LIT-2017-3-YOU-003 and FWF grant P 28660-N31.

Appendix A: Full list of the used CAMELS Catchment Characteristics

Table A1. Table of catchment attributes used in this experiments. Description taken from the data set Addor et al. (2017a)

p_mean	Mean daily precipitation.
pet_mean	Mean daily potential evapotranspiration.
aridity	Ratio of mean PET to mean precipitation.
p_seasonality	Seasonality and timing of precipitation. Estimated by representing annual precipitation and temperature as sin waves. Positive (negative) values indicate precipitation peaks during the summer (winter). Values of approx. 0 indicate uniform precipitation throughout the year.
frac_snow_daily	Fraction of precipitation falling on days with temperatures below $0°C$.
high_prec_freq	Frequency of high precipitation days (>= 5 times mean daily precipitation).
high_prec_dur	Average duration of high precipitation events (number of consecutive days with >= 5 times mean daily precipitation).
low_prec_freq	Frequency of dry days (< 1 mm/day).
low_prec_dur	Average duration of dry periods (number of consecutive days with precipitation < 1 mm/day).
elev_mean	Catchment mean elevation.
slope_mean	Catchment mean slope.
area_gages2	Catchment area.
forest_frac	Forest fraction.
lai_max	Maximum monthly mean of leaf area index.
lai_diff	Difference between the max. and min. mean of the leaf area index.
gvf_max	Maximum monthly mean of green vegetation fraction.
gvf_diff	Difference between the maximum and minimum monthly mean of the green vegetation fraction.
soil_depth_pelletier	Depth to bedrock (maximum 50m).
soil_depth_statsgo	Soil depth (maximum 1.5m).
soil_porosity	Volumetric porosity.
soil_conductivity	Saturated hydraulic conductivity.
max_water_content	Maximum water content of the soil.
sand_frac	Fraction of sand in the soil.
silt_frac	Fraction of silt in the soil.
clay_frac	Fraction of clay in the soil.
carb_rocks_frac	Fraction of the catchment area characterized as "Carbonate sedimentary rocks".
geol_permeability	Surface permeability (log10).

Appendix B: Hyperparameter tuning

The hyperparameters, i.e., the number of hidden/cell states, dropout rate, length of the input sequence and the number of stacked LSTM layers for our model were found by running grid search over a range of parameter values. Concretely we considered the following possible parameter values:

1. Hidden states: 64, 96, 128, 156, 196, 224, 256
2. Dropout rate: 0.0, 0.25, 0.4, 0.5
3. Length of input sequence: 90, 180, 270, 365
4. Number of stacked LSTM layer: 1, 2

We used k-fold cross validation ($k = 4$) to split the basins into an a training set and an independent test set. We trained one model for each split for each parameter combination on the combined calibration period of all basins in the specific training set and evaluated the model performance on the calibration data of the test basins. The final configuration was chosen by taking the parameter set that resulted in the highest median NSE over all possible parameter configurations. The parameters are:

1. Hidden states: 256
2. Dropout rate: 0.4
3. Length of input sequence length: 270
4. Number of stacked LSTM layer: 1

References

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P.: The CAMELS data set: Catchment attributes and meteorology for large-sample studies, Hydrology and Earth System Sciences, 21, 5293–5313, 2017a.

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P.: Catchment attributes for large-sample studies, Boulder, CO: UCAR/NCAR, https://doi.org/https://doi.org/10.5065/D6G73C3Q, 2017b.

Addor, N., Nearing, G., Prieto, C., Newman, A. J., Le Vine, N., and Clark, M. P.: Selection of hydrological signatures for large-sample hydrology, 2018.

Anderson, E. A.: National Weather Service river forecast system: Snow accumulation and ablation model, NOAA Tech. Memo. NWS HYDRO-17, 87 pp., 1973.

Beck, H. E., van Dijk, A. I. J. M., de Roo, A., Miralles, D. G., McVicar, T. R., Schellekens, J., and Bruijnzeel, L. A.: Global-scale regionalization of hydrologic model parameters, Water Resources Research, 52, 3599–3622, https://doi.org/10.1002/2015WR018247, 2016.

Besaw, L. E., Rizzo, D. M., Bierman, P. R., and Hackett, W. R.: Advances in ungauged streamflow prediction using artificial neural networks, Journal of Hydrology, 386, 27–37, 2010.

Beven, K. and Freer, J.: Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methodology, Journal of Hydrology, 249, 11 – 29, 2001.

Blöschl, G. and Sivapalan, M.: Scale issues in hydrological modelling: a review, Hydrological processes, 9, 251–290, 1995.

Blöschl, G., Sivapalan, M., Savenije, H., Wagener, T., and Viglione, A.: Runoff prediction in ungauged basins: synthesis across processes, places and scales, Cambridge University Press, 2013.

Burnash, R.: The NWS river forecast system-catchment modeling, Computer models of watershed hydrology, 188, 311–366, 1995.

Burnash, R. J., Ferral, R. L., and McGuire, R. A.: A generalized streamflow simulation system, conceptual modeling for digital computers, Joint Federal and State River Forecast Center,U.S. National Weather Service, and California Departmentof Water Resources Tech. Rep., 204 pp., 1973.

Clark, M. P., Slater, A. G., Rupp, D. E., Woods, R. A., Vrugt, J. A., Gupta, H. V., Wagener, T., and Hay, L. E.: Framework for Understanding Structural Errors (FUSE): A modular framework to diagnose differences between hydrological models, Water Resour. Res., 44, 2008.

Cohen, J.: Statistical power analysis for the behavioral sciences, Routledge, 2013.

Gupta, H. V., Sorooshian, S., and Yapo, P. O.: Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information, Water Resources Research, 34, 751–763, 1998.

Gupta, H. V., Wagener, T., and Liu, Y.: Reconciling theory with observations: elements of a diagnostic approach to model evaluation, Hydrological Processes: An International Journal, 22, 3802–3813, 2008.

Gupta, H. V., Kling, H., Yilmaz, K. K., and Martinez, G. F.: Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling, Journal of hydrology, 377, 80–91, 2009.

Gupta, H. V., Perrin, C., Bloschl, G., Montanari, A., Kumar, R., Clark, M., and Andréassian, V.: Large-sample hydrology: a need to balance depth with breadth, Hydrology and Earth System Sciences, 18, p–463, 2014.

Henn, B., Clark, M. P., Kavetski, D., and Lundquist, J. D.: Estimating mountain basin-mean precipitation fromstreamflow using Bayesian inference, Water Resour. Res., 51, 2008.

Herman, J. D., Kollat, J. B., Reed, P. M., and Wagener, T.: Technical Note: Method of Morris effectively reduces the computational demands of global sensitivity analysis for distributed watershed models, Hydrology and Earth System Sciences, 17, 2893–2903, 2013.

Hochreiter, S.: Untersuchungen zu dynamischen neuronalen Netzen, Diploma, Technische Universität München, 91, 1991.

Hochreiter, S. and Schmidhuber, J.: Long short-term memory, Neural computation, 9, 1735–1780, 1997.

Hrachowitz, M., Savenije, H., Blöschl, G., McDonnell, J., Sivapalan, M., Pomeroy, J., Arheimer, B., Blume, T., Clark, M., Ehret, U., et al.: A decade of Predictions in Ungauged Basins (PUB)—a review, Hydrological sciences journal, 58, 1198–1255, 2013.

Hunter, J. D.: Matplotlib: A 2D graphics environment, Computing In Science & Engineering, 9, 90–95, 2007.

Jakeman, A. J. and Hornberger, G. M.: How much complexity is warranted in a rainfall-runoff model?, Water Resources Research, 29, 2637–2649, 1993.

Kirchner, J. W.: Getting the right answers for the right reasons: Linking measurements, analyses, and models to advance the science of hydrology, Water Resources Research, 42, 2006.

Kratzert, F., Herrnegger, M., Klotz, D., Hochreiter, S., and Klambauer, G.: Do internals of neural networks make sense in the context of hydrology?, in: AGU Fall Meeting Abstracts, 2018a.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M.: Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks, Hydrology and Earth System Sciences, 22, 6005–6022, 2018b.

Kratzert, F., Klotz, D., Herrnegger, M., and Hochreiter, S.: A glimpse into the Unobserved: Runoff simulation for ungauged catchments with LSTMs, in: Workshop on Modeling and Decision- Making in the Spatiotemporal Domain, 32nd Conference on Neural Information Processing Systems (NeuRIPS 2018), 2018c.

Kratzert, F., Herrnegger, M., Klotz, D., Hochreiter, S., and Klambauer, G.: NeuralHydrology-Interpreting LSTMs in Hydrology, arXiv preprint arXiv:1903.07903, 2019.

Kumar, R., Samaniego, L., and Attinger, S.: Implications of distributed hydrologic model parameterization on water fluxes at multiple scales and locations, Water Resources Research, 49, 360–379, 2013.

Liang, X., Lettenmaier, D. P., Wood, E. F., and Burges, S. J.: A simple hydrologically based model of land surface water and energy fluxes for general circulation models, Journal of Geophysical Research: Atmospheres, 99, 14 415–14 428, 1994.

McInnes, L., Healy, J., and Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction, arXiv preprint arXiv:1802.03426, 2018.

McKinney, W.: Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 1697900, 51–56, 2010.

Mizukami, N., Clark, M. P., Newman, A. J., Wood, A. W., Gutmann, E. D., Nijssen, B., Rakovec, O., and Samaniego, L.: Towards seamless large-domain parameter estimation for hydrologic models, Water Resources Research, 53, 8020–8040, 2017.

Mizukami, N., Rakovec, O., Newman, A. J., Clark, M. P., Wood, A. W., Gupta, H. V., and Kumar, R.: On the choice of calibration metrics for "high-flow" estimation using hydrologic models, Hydrology and Earth System Sciences, 23, 2601–2614, 2019.

Morris, M. D.: Factorial sampling plans for preliminary computational experiments, Technometrics, 33, 161–174, 1991.

Naef, F.: Can we model the rainfall-runoff process today? / Peut-on actuellement mettre en modèle le processus pluie-écoulement?, Hydrological Sciences Bulletin, 26, 281–289, 1981.

Nash, J. E. and Sutcliffe, J. V.: River flow forecasting through conceptual models part I—A discussion of principles, Journal of hydrology, 10, 282–290, 1970.

Newman, A., Sampson, K., Clark, M., Bock, A., Viger, R., and Blodgett, D.: A large-sample watershed-scale hydrometeorological dataset for the contiguous USA, Boulder, CO: UCAR/NCAR, https://doi.org/https://dx.doi.org/10.5065/D6MW2F4D, 2014.

Newman, A. J., Clark, M. P., Sampson, K., Wood, A., Hay, L. E., Bock, A., Viger, R. J., Blodgett, D., Brekke, L., Arnold, J. R., Hopson, T., and Duan, Q.: Development of a large-sample watershed-scale hydrometeorological data set for the contiguous USA: Data set characteristics and assessment of regional variability in hydrologic model performance, Hydrology and Earth System Sciences, 19, 209–223, 2015.

Newman, A. J., Mizukami, N., Clark, M. P., Wood, A. W., Nijssen, B., and Nearing, G.: Benchmarking of a physically based hydrologic model, Journal of Hydrometeorology, 18, 2215–2225, 2017.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A.: Automatic differentiation in PyTorch, 2017.

Perrin, C., Michel, C., and Andréassian, V.: Does a large number of parameters enhance model performance? Comparative assessment of common catchment model structures on 429 catchments, Journal of Hydrology, 242, 275 – 301, 2001.

Peters-Lidard, C. D., Clark, M., Samaniego, L., Verhoest, N. E., Van Emmerik, T., Uijlenhoet, R., Achieng, K., Franz, T. E., and Woods, R.: Scaling, similarity, and the fourth paradigm for hydrology, Hydrology and earth system sciences, 21, 3701, 2017.

Prieto, C., Le Vine, N., Kavetski, D., García, E., and Medina, R.: Flow Prediction in Ungauged Catchments Using Probabilistic Random Forests Regionalization and New Statistical Adequacy Tests, Water Resources Research, 55, 4364–4392, 2019.

Rakovec, O., Mizukami, N., Kumar, R., Newman, A. J., Thober, S., Wood, A. W., Clark, M. P., and Samaniego, L.: Diagnostic Evaluation of Large-domain Hydrologic Models calibrated across the Contiguous United States, J. Geophysical Research – Atmospheres., in review, 2019.

Razavi, T. and Coulibaly, P.: Streamflow Prediction in Ungauged Basins: Review of Regionalization Methods, Journal of Hydrologic Engineering, 18, 958–975, 2013.

Saltelli, A., Tarantola, S., Campolongo, F., and Ratto, M.: Sensitivity analysis in practice: a guide to assessing scientific models, pp. 94–100, Wiley Online Library, 2004.

Samaniego, L., Kumar, R., and Attinger, S.: Multiscale parameter regionalization of a grid-based hydrologic model at the mesoscale, Water Resources Research, 46, 2010.

Seibert, J.: Regionalisation of parameters for a conceptual rainfall-runoff model, Agricultural and Forest Meteorology, 98-99, 279 – 293, 1999.

Seibert, J. and Vis, M. J. P.: Teaching hydrological modeling with a user-friendly catchment-runoff-model software package, Hydrology and Earth System Sciences, 16, 3315–3325, 2012.

Seibert, J., Vis, M. J. P., Lewis, E., and van Meerveld, H. J.: Upper and lower benchmarks in hydrological modelling, Hydrological Processes, 32, 1120–1125, 2018.

Sivapalan, M., Takeuchi, K., Franks, S., Gupta, V., Karambiri, H., Lakshmi, V., Liang, X., McDonnell, J., Mendiondo, E., O'connell, P., et al.: IAHS Decade on Predictions in Ungauged Basins (PUB), 2003–2012: Shaping an exciting future for the hydrological sciences, Hydrological sciences journal, 48, 857–880, 2003.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research, 15, 1929–1958, 2014.

Van Der Walt, S., Colbert, S. C., and Varoquaux, G.: The NumPy array: A structure for efficient numerical computation, Computing in Science and Engineering, 13, 22–30, 2011.

van Rossum, G.: Python tutorial, Technical Report CS-R9526, Tech. rep., Centrum voor Wiskunde en Informatica (CWI), Amsterdam, 1995.

Wang, A. and Solomatine, D. P.: Practical Experience of Sensitivity Analysis: Comparing Six Methods, on Three Hydrological Models, with Three Performance Criteria, Water, 11, 1062, 2019.

Wilcoxon, F.: Individual comparisons by ranking methods, Biometrics bulletin, 1, 80–83, 1945.

Wood, A. W., Maurer, E. P., Kumar, A., and Lettenmaier, D. P.: Long-range experimental hydrologic forecasting for the eastern United States, Journal of Geophysical Research: Atmospheres, 107, ACL–6, 2002.

Yilmaz, K. K., Gupta, H. V., and Wagener, T.: A process-based diagnostic approach to model evaluation: Application to the NWS distributed hydrologic model, Water Resources Research, 44, 1–18, 2008.