Probing the limit of hydrologic predictability with the Transformer network

Jiangtao Liu[1], Yuchen Bian[2], Kathryn Lawson[1], and Chaopeng Shen*[,1]
[1] Civil and Environmental Engineering, The Pennsylvania State University
[2] Amazon Search Science and AI
* Corresponding author: Chaopeng Shen, cshen@engr.psu.edu

Keywords: Transformer, long short-term memory, streamflow, CAMELS, deep learning

Abstract

For a number of years since their introduction to hydrology, recurrent neural networks like long short-term memory (LSTM) networks have proven remarkably difficult to surpass in terms of daily hydrograph metrics on community-shared benchmarks. Outside of hydrology, Transformers have now become the model of choice for sequential prediction tasks, making it a curious architecture to investigate for application to hydrology. Here, we first show that a vanilla (basic) Transformer architecture is not competitive against LSTM on the widely benchmarked CAMELS streamflow dataset, and lagged especially prominently for the high-flow metrics, perhaps due to the lack of memory mechanisms. However, a recurrence-free variant of the Transformer model can obtain mixed comparisons with LSTM, producing very slightly higher Kling-Gupta efficiency coefficients (KGE), along with other metrics. The lack of advantages for the vanilla Transformer network is linked to the nature of hydrologic processes. Additionally, similar to LSTM, the Transformer can also merge multiple meteorological forcing datasets to improve model performance. Therefore, the modified Transformer represents a rare competitive architecture to LSTM in rigorous benchmarks. Valuable lessons were learned: (1) the basic Transformer architecture is not suitable for hydrologic modeling; (2) the recurrence-free modification is beneficial so future work should continue to test such modifications; and (3) the performance of state-of-the-art models may be close to the prediction limits of the dataset. As a non-recurrent model, the Transformer may bear scale advantages for learning from bigger datasets and storing knowledge. This work lays the groundwork for future explorations into pretraining models, serving as a foundational benchmark that underscores the potential benefits in hydrology.

Introduction

Rainfall-runoff modeling is essential for flood prediction, water resource management, and environmental protection (Hrachowitz & Clark, 2017). Rainfall-runoff modeling is a critical aspect of hydrology, as it models the intricate relationships between precipitation, watershed characteristics, and streamflow. The introduction of long short-term memory (LSTM) networks marked a significant advancement in this field for numerous variables of interest including soil moisture (Fang et al., 2017; J. Liu et al., 2022, 2023), streamflow (Botterill & McMillan, 2023; Feng et al., 2020, 2021; Konapala et al., 2020; Kratzert et al., 2019; Sun et al., 2021; Xiang & Demir, 2020), water temperature (Rahmani, Lawson, et al., 2021; Rahmani, Shen, et al., 2021), and groundwater levels (Afzaal et al., 2020; Wunsch et al., 2022). For these applications, LSTM consistently outperformed traditional models and process-based models (Feng et al., 2020; Papacharalampous et al., 2018). LSTM's ability to learn many-step dependencies and handle variable-length input sequences has proven particularly advantageous in capturing the inherent complexity of hydrological processes (Hochreiter & Schmidhuber, 1997).

As a recurrent neural network (RNN), LSTM processes data sequentially through time steps, updating its internal states at each step based on the current input and the previous states. This iterative process, which involves repeatedly applying its internal neural network mechanisms, leads to some limitations. The recurrent nature means RNNs are prone to an issue called the vanishing gradient (Hochreiter, 1991; Hochreiter et al., 2001), where the gradient of the loss with respect to the network weights becomes very small, making network training extremely slow. This issue limits the length of the training sequence, and reduces the impact of inputs from the longer-term past on present predictions. This could be one of the reasons why baseflow was previously identified as a limitation (Feng et al., 2020). Even though LSTM was developed to mitigate this issue and can suppress it better than the original RNNs, it is not immune to it (Dai et al., 2019;

Zhang et al., 2016). Furthermore, recurrence means these time steps must be taken in sequence --- the time steps cannot be run in parallel. This imposes a restriction on the efficiency of parallel processing, and thus the scale of data on which the model can be trained.

In many applications outside hydrology, the Transformer architecture (Vaswani et al., 2017) has demonstrated superior performance over LSTM networks in various domains, including machine translation, speech recognition (Karita et al., 2019), natural language processing and sentiment analysis (Devlin et al., 2019), question answering (Rajpurkar et al., 2018), computer vision (Carion et al., 2020), protein structure prediction (Rives et al., 2021), and music generation (Huang et al., 2018). The Transformer model uses an attention mechanism, where each word (or “input token”) is transformed into three different kinds of information: a 'query' that asks how relevant other words are to it, a 'key' that responds to others' queries about its relevance, and a 'value' that carries the word's actual meaning. The model calculates the relevancies between the query and keys of all words, then combines the values of the most relevant words to understand the current word better. With LSTM, the most recent input tokens are always more important than further-away ones, whereas a Transformer could learn to put more focus on further-away tokens (Dehghani et al., 2019; Raganato & Tiedemann, 2018), which makes it ideal for language modeling. Moreover, as it does not have recurrence, a Transformer can run the time steps in parallel and can scale up in parallel computation when more data and more GPUs are available. Considering such benefits, there should be a heightened interest in harnessing Transformers for hydrologic applications. Transformers are increasingly being used in hydrologic and water quality modeling (Castangia et al., 2023; Koya & Roy, 2023; Li & Yang, 2019; Xu et al., 2021; H. Yang et al., 2023), especially for near-term forecasting. However, the scale of application tends to be limited and their benchmarking on standardized, well-understood datasets, such as the Catchment Attributes and Meteorology for Large-sample Studies (CAMELS) dataset (Addor et al., 2017; Newman et al., 2014), remains limited in the literature. It is thus intriguing whether the

Transformer's advantages over recurrent networks will apply in the case of natural systems, which can be argued to lack the irregular sequential structure found in human languages.

While some past hydrologic studies have claimed superior performance for some other architectures compared to LSTM, many times, a rigorous comparison was not carried out due to the different modeling objectives. The conclusions were often conditional on using a small dataset for benchmarking, e.g., see (Abed et al., 2022; Amanambu et al., 2022; Ghobadi & Kang, 2022), using procedures and configurations (training and test periods, sites, and forcing data) that are different from published benchmarks (Yin et al., 2022, 2023), or on a case study which was not easy to compare to the work of other independent teams (Koya & Roy, 2023; C. Liu et al., 2022). Specifically, Yin et al. (2022) proposed the RR-Former model (a transformer variant) and conducted experiments with 7-day forecasts on the CAMELS dataset. They modeled 673 distinct basins independently and calculated performance metrics for each, and they also assessed a selected set of 448 basins using a single model. In contrast, our research primarily focuses on long-term prediction problems rather than forecasting. Forecasting typically involves predicting results within a relatively short period based on historical data, whereas our study concentrates on the long-term rainfall-runoff relationship to better understand its underlying patterns. Building on the work by Yin et al. (2022), Yin et al. (2023) introduced the RRS-Former model, which conducted a one-day-ahead runoff experiment. A similar study by (Feng et al., 2020) applied a data integration approach to an LSTM model. Although direct comparison is challenging, Feng et al. (2020) reported a median NSE of 0.86, which was superior to the RRS-Former model's performance in Yin et al. (2023). Koya and Roy (2023) evaluated the Temporal Fusion Transformer (TFT) model on the Caravan dataset (Kratzert et al., 2023) and reported median Kling Gupta efficiency (KGE) of 0.705. However, Feng et al., (2023) benchmarked LSTM on a similarly large dataset and showed median KGEs of 0.74 for 3753 global basins and 0.78 for 1675 basins with long-term records. Furthermore, while more benchmarking is welcomed, the model in

Koya and Roy (2023) is not purely attention-based, as it incorporates some LSTM layers in its encoder, making it difficult to determine whether the performance improvements are due to the attention mechanism or the LSTM layers. This approach also brought back time recurrence and did not leverage the time parallelism of the transformer network as advocated in the original transformer model. In the interest of reproducibility and comparability, which underpin scientific progress, it is useful to benchmark under similar conditions, on the same (reasonably large) dataset. Data-driven deep learning models enjoy the feature of "data synergy", where larger and more diverse data leads to stronger and more robust models (Fang et al., 2022; Kratzert et al., 2021; Pasquiou et al., 2022; E. Yang et al., 2023). Thus small-data comparison results may not be valid for a case with more data. Thus far, on the CAMELS dataset (Addor et al., 2017; Newman et al., 2014), both Kratzert et al. (2019) and Feng et al. (2021) reported very similar metric Nash-Sutcliffe model efficiency coefficient (NSE) (Nash & Sutcliffe, 1970) for LSTM --- 0.72 for 571 basins with the NLDAS forcing alone, making this a reliable benchmark that has thus far not been exceeded by other models. Sun et al. (2021) reported comparable results using GraphWaveNet, although with different training periods and ensemble setups. Furthermore, Kratzert simultaneously employed multiple forcing dataset (NLDAS, Maurer, and Daymet) for LSTM and obtained a Kling-Gupta model efficiency coefficient (KGE) (Gupta et al., 2009) of 0.80, which is the record on this dataset that no other model has matched.

In this study, we investigate the performance of the Transformer architecture in rainfall-runoff modeling compared to LSTM using the CAMELS dataset. We analyze the performance of single models and ensembles for both architectures, and examine the models' ability to handle multiple forcings and mixed forcing cases. This approach aims to establish a reference point for future studies to compare, enhancing our understanding of these models in complex scenarios. Our findings contribute to the understanding of the strengths and limitations of both LSTM and

Transformer models in hydrological predictions, and highlight the potential of the Transformer as an alternative and scalable solution for hydrologic modeling.

Data and Methods

Datasets

In this paper, we utilized the Catchment Attributes and Meteorology for Large-sample Studies (CAMELS) dataset (Addor et al., 2017; Newman et al., 2014), which includes basin-averaged daily data from 671 catchments across the conterminous United States (CONUS) characterized by minimal anthropogenic disturbances. The catchment attributes encompass an array of characteristics such as topography, soil properties, and geological aspects. Furthermore, CAMELS provides daily meteorological forcing inputs derived from three distinct gridded data products, namely Daymet (Thornton et al., 1997), Maurer (Maurer et al., 2002), and the North American Land Data Assimilation System (NLDAS) (Xia et al., 2012).

Vanilla (basic) Transformer models

The Transformer model, as first introduced in the paper “Attention is all you need” by Vaswani et al. (2017), is a neural network architecture for sequential data processing. The Transformer model consists of an encoder and a decoder. The encoder has a number (n_{layer}) of stacked encoding layers (“stacked” means the output of one layer becomes the input to the next one), each of which consists of a self-attention layer and a position-wise fully connected layer, while the decoder has only a simple position-wise linear layer. The critical mechanism within the encoder is self-attention, which computes the weighted sum of all input features. The equations for one of the stacked encoding layers are shown below and explain the calculations one by one.

$$Q = x * D(W_q) \quad (1)$$

$$K = x * D(W_k) \quad (2)$$

$$V = x * D(W_v) \quad (3)$$

$$a = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (4)$$

$$c = a * V \quad (5)$$

$$u' = c * W_o \quad (6)$$

$$u = BatchNorm(x + D(u')) \quad (7)$$

$$z' = W_2 * D(GELU(W_1 * u)) \quad (8)$$

$$z = BatchNorm(u + z') \quad (9)$$

The inputs *x* to the attention layer have two dimensions --- the sequence length (*n*) and a hidden-size dimension (d_k). In Equations (1-3), the layer computes three sets of linear transformations, called Query, Key, and Value vectors (Q, K, V), and W_q, W_k, and W_v, all with the dimensions (d_k, d_k), represent the respective learnable weights. These position-wise transformations (or matrix multiplications) mix information along the hidden-size dimension, not along the sequence length dimension, while applying the dropout operator *D()*. To mitigate overfitting, a dropout mask with a ratio of 0.5 is applied to W_q, W_k, and W_v. Equation 4 computes the dot product of Query and Key, and obtains a matrix of the size (n, n) which tabulates the similarity between each Query-Key pair. It then scales the calculations by $\sqrt{d_k}$, before applying the softmax operation along the sequence dimension. The output a is the above-mentioned attention weight while c is the attention-weighted values, called "contexts". The model is called "multiheaded" in that multiple sets of Q, K, V are computed and their results c are concatenated as *c* before applying a linear layer in Equation 6. Equations 7-9 apply additional linear layers with activation functions and residual connections to enhance training. z' is a feed-forward neural network (FFN) consisting of two linear transformations with a Gaussian Error Linear Unit (GELU) activation function in between. z includes a residual connection and batch normalization, where the elements along the

batch dimension is normalized. As described earlier, Equations 1-9 are repeated n_{layer} times and the outputs of one layer serves as the inputs to the subsequent layer. The dimensional descriptions here all ignore the batch dimension (a collection of instances to compute a loss value and update the weights) which is in practice computed in parallel. The sequence length (n), the number of heads (h) and the hidden size (d_k) are hyperparameters to be tuned using the validation dataset.

Equations (4-5) can be interpreted as weighing every token in the sequence to make a combined prediction at a given location. We observe that, unlike RNNs which would naturally put more weight to adjacent tokens, the sense of adjacency is lost for the attention layer --- for prediction location *i*, all input tokens are treated equally, regardless whether they are close or far from *i*. The larger focus to adjacent tokens, if it exists in the training dataset, is completely obtained from data. Furthermore, any relational structure in the sequence dimension is not modeled --- the softmax operator in Equation 4 is the only operator that mixes information over the sequence length, as all the other operators are calculated in parallel for each token in the sequence. This setup is reasonable in language modeling where inversion structures are common, but may not be optimal if the proximity is important as in natural physical processes. However, stacking many layers of attention sequentially as done in the Transformer could enable the modeling of some sequential structure.

The initial input to the model, X, is of dimension (n, n_X), which is transformed by an embedding function. It includes three parts: a linear layer transformation of the inputs, a “positional embedding” (Equation 10-11), and a “temporal embedding” (Equation 12-13). These three components are directly summed to obtain the input *x* in Equations 1-3 which is then fed into the attention layers described above. The embeddings are added because the Transformer does not inherently account for the positional information. The positional encoding uses sine and cosine functions to

create a unique encoding for each position, allowing the model's self-attention mechanism to maintain the sequence order in context (Vaswani et al., 2017):

$$P_{k,2i} = sin\left(\frac{k}{10000^{2i/d}}\right) \tag{10}$$

$$P_{k,2i+1} = cos\left(\frac{k}{10000^{2i/d}}\right) \tag{11}$$

where *k* is the position in the sequence, *i* is the dimension, and *d* is the number of columns in the embedding matrix.

Furthermore, in the time series data, positional embedding alone can hardly reflect the seasonality information. Hence, hierarchical global timestamp information (weekly, monthly, yearly) is used to encode seasonality and long-term ordinal information (Zhou et al., 2021). This temporal embedding calculates and normalizes the day of the week, day of the month, and day of the year for each time period to a range of -0.5 to 0.5:

$$d_i(k) = t_i(k)/ N_i - 0.5 \tag{12}$$

$$T_e(k) = \oplus_{\{i \in time\_features\}} d_i(k) \tag{13}$$

where t_i is the value of time feature *i* at position *k* in the sequence; for example, day of the week, day of the month, or day of the year. N_i is the total number of values for the time feature *i*; for example, for the day of the year *i*, N_i would be 365. d_i is the normalized value for each time feature. $T_e(k)$ indicates the temporal embedding at position *k*. The '$\oplus$' symbol denotes concatenation, meaning it concatenates the time features into a single vector at the last dimension.

Positional and temporal embeddings are added to the input embeddings to form the input to the transformer layer.

$$x = E_X(X) + E_P(P) + E_T(\mathrm{T}) \tag{14}$$

where E_X , E_P, E_P are the learned linear embedding layers projecting the inputs to the model hidden dimension, respectively.

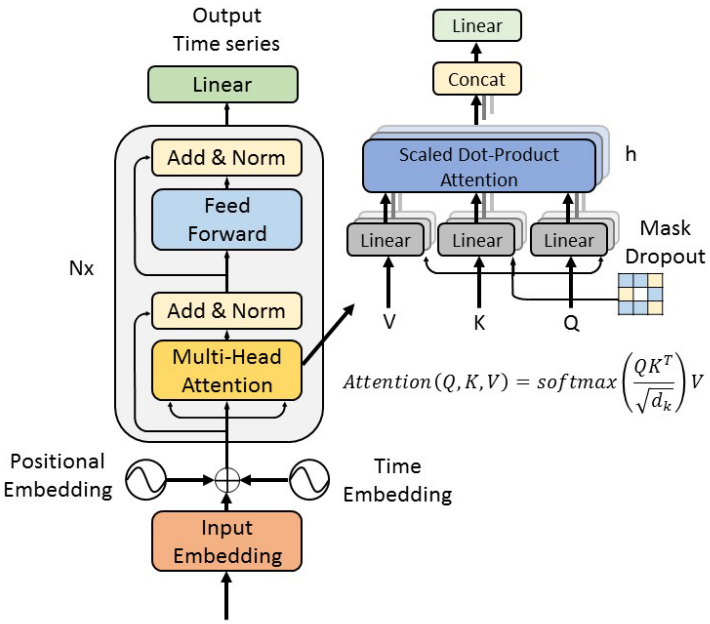

Figure 1. The base Transformer model structure (adapted from Figure 1 in Vaswani et al., 2017) used in this paper.

The Modified Transformer Model with Convolutional Embeddings

As a variant of the Transformer model, we added a one-dimensional convolutional embedding layer just before the attention layer to produce relational features in the time dimension. In this embedding layer, two stacked convolution sub-layers were introduced, with residual connections between them, and their outputs are fed into a linear layer. In each convolutional sub-layer, the time sequence length dimension gets convolved and, as such convolutions are non-recurrent, the model does not need to go through time steps in order to represent the

temporal relational structures. The convolutional sub-layers have a dilation of 1, a stride of 1, ReLU as the activation functions, and a backward-focusing kernel to ensure that inputs from the future do not get used to make a prediction of the current time step. The kernel width, hidden sizes and the number of convolutional layers were set as hyperparameters that were tuned along with the hyperparameters of the attention layers. The outputs of the whole convolutional embedding layer are, along with the time positional and temporal embeddings, added to the input embeddings just as in Equation (14).

LSTM Models and SAC-SMA Models

In order to impartially evaluate the Transformer model's performance, we compared its results with those of LSTM and the Sacramento Soil Moisture Accounting (SAC-SMA) conceptual model (Anderson & McDonnell, 2005; Burnash et al., 1973), and used the latter two as benchmarks. We downloaded the SCA-SMA dataset from HydroShare (Kratzert et al., 2019), and set the same test time for all models to ensure a balanced comparison. This approach helps provide a thorough and fair assessment of each model's performance capabilities. The LSTM model's configurations were based on Kratzert et al. (2021), with the models' hyperparameters set to 30 epochs, a sequence length of 365, a hidden size of 256, and a dropout rate of 0.4.

The LSTM model from Kratzert et al., 2019 was originally evaluated on 531 basins. To broaden our insights into the impacts of a single forcing dataset on the entire CAMELS dataset and ensure a fair comparison, we retrained their model on the full set of 671 basins with the single NLDAS forcing dataset. We further attempted to incorporate time stamp information as inputs into the LSTM model (data not shown). However, this did not lead to any improved performance, suggesting it does not introduce new information to LSTM.

Experiments and Model Evaluation

To enable a comprehensive comparison with various benchmarks, we utilized data from all 671 CAMELS basins. To be consistent with previous benchmark experiments, we employ both single and multi-forcing datasets, in the same manner as the benchmark (Kratzert et al., 2021). Initially, we applied a single forcing dataset derived from NLDAS, referred to as the 'single-forcing' experiment. Subsequently, we conducted a multi-forcing analysis using forcing data from Daymet, Maurer, and NLDAS. For this analysis, our scope was narrowed to 531 basins. It should be noted that the variable selection and settings for the model input data were chosen to be consistently aligned with those employed by Kratzert et al. (2021).

For all models, the data used for the training period was from 1 October 1999 to 30 September 2008, while the data used for the testing period was from 1 October 1989 to 30 September 1999. During the training period, the weights were optimized using the Adam optimizer with a learning rate of 0.0001.

To accurately compare different model architectures and hyperparameters, we used one specific seed in Figure 2. Thus any differences in model performance can be fully attributed to the specific architectural or hyperparameter variations. To increase the robustness of the analysis, we employed an ensemble approach, using ten simulations with different random seeds for each of the model architectures. The ensemble-averaged discharge for each model architecture is what is presented in the results here, as it not only helps to capture the variation in results due to randomness, but also provides more stable performance estimates.

We evaluated model performance using several metrics, including the Nash-Sutcliffe model efficiency coefficient (NSE) (Nash & Sutcliffe, 1970) and the Kling-Gupta model efficiency

coefficient (KGE) (Gupta et al., 2009). We also considered the percent bias of the top 2% peak flow range (FHV) and the bottom 30% low flow range (FLV), which respectively highlight model performance for peak flows and baseflow (Yilmaz et al., 2008).

Results and Discussion

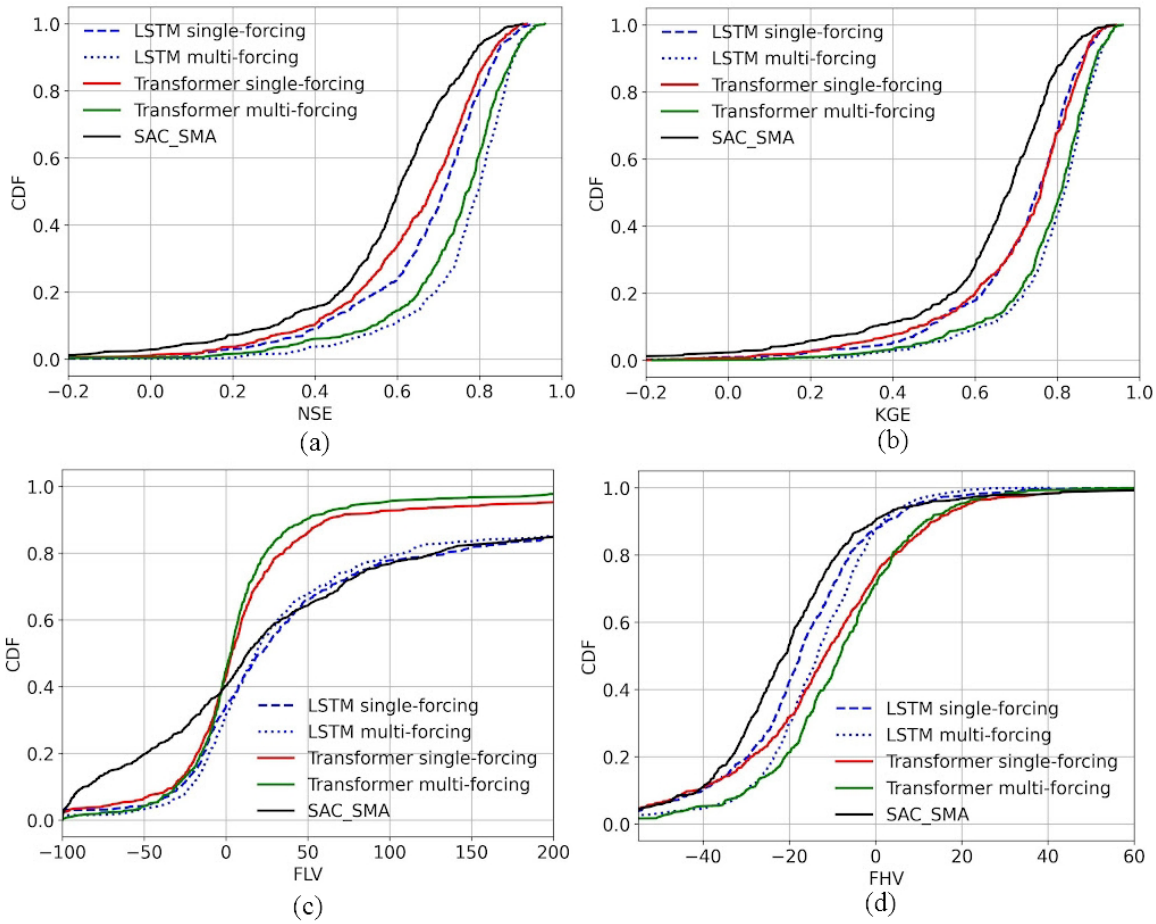

Figure 2. Comparative analysis of Cumulative Density Function (CDF) across various models including Long Short-Term Memory (LSTM) and modified Transformer deep learning models, and the conceptual Sacramento Soil Moisture Accounting (SAC-SMA), with units in mm/day and one specific seed (rather than a random seed). The model encompasses single and multi-forcing data for models. The figure depicts the following comparisons: (a) Nash-Sutcliffe Efficiency (NSE) vs CDF, (b) Kling-Gupta Efficiency (KGE) vs CDF, (c) Low flow percent bias (FLV) vs CDF, and (d) High flow percent bias (FHV) vs CDF. Single-forcing models were implemented on a set of 671 basins in the CAMELS dataset, whereas multi-forcing models were applied to a subset of 531 basins from that dataset.

Table 1. Comparative performance metrics for single and multi-forcing experiments with LSTM, vanilla Transformer, and modified Transformer models. We conducted an evaluation of single forcing on 671 basins and multi-forcing on 531 basins, employing the LSTM model results from Kratzert et al., 2019, originally evaluated on 531 basins. To broaden our insights into the impacts of single-forcing on the entire CAMELS dataset and make a fair comparison, we retrained their model on an expanded set of 671 basins with single NLDAS dataset. These numbers are only very slightly different from Kratzert et al., 2019. The means for Kling-Gupta Efficiency (KGE), high flow percent bias (FHV), and low flow percent bias (FLV) are averages from the 10 different ensemble members, each with a different random seed, while the standard deviations (std) for KGE, FHV and FLV are calculated for the ensemble members.

	Forcing: NLDAS			Forcing: Multi-forcing		
	LSTM	Vanilla Transformer	Modified Transformer	LSTM	Vanilla Transformer	Modified Transformer
KGE (mean±std)	0.73 ±0.003	0.71 ±0.007	0.74 ±0.007	0.80 ±0.004	0.77 ±0.016	0.80 ±0.007
FHV (mean±std)	-17.49 ±0.58	-26.66 ±2.83	-18.00 ±2.94	-11.91 ±0.549	-21.54 ±2.64	-9.19 ±4.01
FLV (mean±std)	-2.82 ±8.15	3.31 ±2.34	2.28 ±4.24	2.57 ±4.072	0.77 ±1.65	2.72 ±2.41

The SCA-SMA model had the lowest performance across all experiments, and aligns consistently with the results of Feng et al. (2020) and Kratzert et al. (2021). For the single-forcing CAMELS benchmark (671 basins), the vanilla Transformer was outperformed by LSTM (Table 1; Figure 2). Overall, the vanilla Transformer fell behind LSTM in all metrics, although not by much. Looking at Kling-Gupta Efficiency, the vanilla Transformer achieved a value of 0.71, compared to 0.73 for the LSTM. These results suggest that, without modification, the vanilla Transformer is missing some critical ability to simulate hydrologic processes.

The vanilla Transformer's under-performance is a curious case as it has been widely recognized that "attention is all you need" (Vaswani et al., 2017) in sequential modeling, and we have several

interpretations of the results presented here. First, it is possible that the dataset size is too small here and advantages for the Transformer could emerge for larger quantities of data. Second, the natural hydrologic process is a “Markovian” system (Grey Nearing, personal communication) where the states at the current time step, rather than more remotely-in-the-past steps, completely determine the system’s trajectory for future time steps (along with the forcings). To be more concrete, the soil moisture today, rather than that from any previous days, would have far more of an impact on tomorrow’s streamflow discharge. This is in strong contrast to human languages where the order of the words can often be inverted without changing the context, which would favor the attention-based Transformer architecture. Third, the accumulation of water and its nonlinear interactions makes memory effects important, but the Transformer does not have memory and is not necessarily strong at capturing the effects of memory. Regardless of the reason, the results mean that the vanilla Transformer is not optimal for streamflow predictions at the very least, and further changes are likely needed in order to use it for modeling natural systems.

On the other hand, the modified Transformer demonstrates performance metrics that are comparable to or slightly surpass those of the LSTM. However, it exhibits greater variability among ensemble members, indicated by the standard deviation of the KGE metric: ±0.003 for the LSTM and ±0.007 for the modified Transformer. Its KGE (0.74) is slightly higher than LSTM (0.73), and the differences in FLV and FHV from LSTM’s values are too small to call an advantage considering their variability. As to be discussed below, while these differences are small, we simply should not expect larger differences as the possible room of improvement may be very small at this stage. The ensemble standard deviation of KGE is 0.003 with LSTM and 0.007 with the modified Transformer. The LSTM has a smaller ensemble standard deviation for FHV than the modified Transformer, while the opposite is true for FLV. The ensemble standard deviation of median FHV is 0.58 for LSTM and 2.94 for the modified Transformer, while this value for the FLV

is 8.15 for the LSTM and 4.24 for the modified Transformer. This suggests that while we obtain very similar overall metrics, the LSTM and the modified Transformer preferentially address different parts of the hydrograph in this experiment. LSTM more reliably focuses on the high-flow regime (quantified by the smaller ensemble standard deviation of FHV) than the modified Transformer, but the latter can better capture the long-term dependence (quantified by ensemble standard deviation of FLV representing groundwater baseflow). It seems there is some tradeoff for the different flow regimes.

The multi-forcing experiment generally shows similar patterns: the vanilla Transformer falls behind the other two models, which have very similar ensemble-mean performance metrics but different ensemble standard deviations. The high KGE (0.80) and slightly better-than-LSTM FHV (9.19) for the modified Transformer demonstrates that it, too, is able to fuse different forcing datasets as can LSTM, which no other model architecture has shown. Just as in the single-forcing NLDAS experiment, the modified Transformer has a larger stochastic variability (quantified by ensemble standard deviation) for FHV but smaller variability for FLV. Because both FHV and FLV have improved compared to the single-forcing experiment, the modified Transformer was able to utilize the short-term and long-term dependencies of multiple forcing datasets. For one particular ensemble member (based on different random seeds), the cumulative density plot shows very similar curves between the modified Transformer and LSTM models.

The high agreement between the two model architectures, both of which are state of the art, suggests that we are likely at or very close to the predictive limit of the CAMELS dataset for this test (temporal test, training in one time period and testing in another). We suspect that unless we bring in new information, it is highly unlikely for any other models to produce noticeable advantages beyond these two models on this dataset, for the tests presented here. Errors with forcing, basin shapes, attribute, and discharge data are likely the remaining factors preventing

higher performance. It should be mentioned that for another test, e.g., prediction in ungauged regions or spatial extrapolation, physics-informed hybrid models (called differentiable models (Shen et al., 2023)) can actually outperform LSTM (Aboelyazeed et al., 2023; Feng et al., 2022; Feng, Beck, Lawson, et al., 2023; Tsai et al., 2021). Moreover, several issues surrounding the CAMELS dataset include using basin-average attributes that cannot resolve subbasin-level spatial heterogeneity, using daily precipitation that does not represent hourly rainfall intensity, a fraction of basins having major reservoirs, and the inclusion of some overly large basins.

Nevertheless, exactly because the Transformer algorithm does not have time integration, it can be trained in a highly parallel fashion and is suited to learning from large amounts of data. As the amount of data and the amount of neurons increase, it is possible to observe emergent behaviors of intelligence (Bubeck et al., 2023). This is a property that is worth further exploring in future studies in hydrology and geosciences. We leave to future work the question of how to incorporate more data with the modified Transformer, and testing this architecture on spatial extrapolation (for data-scarce scenarios) (Feng et al., 2021) and temporal extrapolation (for multidecadal projection of trends).

While some studies claim that Transformer models surpass LSTM models in performance, their evaluations are often limited to small datasets (Abed et al., 2022; Amanambu et al., 2022; Ghobadi & Kang, 2022), forecasting experiments (Yin et al., 2022), or still incorporate a mix of recurrent neural networks and attention mechanisms (Koya & Roy, 2023) (hereafter called KR23). KR23 evaluated the global Caravan dataset using the TFT model, yet several aspects of their approach hinder direct comparison. Firstly, FR23's comparison on a global dataset is valuable and welcomed, as they reported median KGE of 0.705 on 2610 basins for LSTM (basin-by-basin training) across the entire Caravan dataset, but did not provide results for the CAMELS dataset. When LSTM was benchmarked on a similar global dataset in another paper (Feng, Beck, de

Bruijn, et al., 2023) but was trained and tested for all basins simultaneously, it obtained a median KGE of 0.78 for 1675 basins with long-term records, far higher than the KGE of 0.647 reported in KR23, presumably due to using the practice of training on all basins simultaneously (Fang et al., 2022; Kratzert et al., 2024). Secondly, with the addition of LSTM units in KR23's TFT, it is unclear how much the attention mechanisms helped performance in contrast to the LSTM units. Moreover, their model, being a hybrid of RNN and attention, still faces challenges with parallelization issues. Finally, they employed cubic spline interpolation on the streamflow data, which smooths the target data and undermines the comparability of model results. To validate our perspective, we employed the TFT model from PyTorch Forecasting (Beitner, 2020) on the CAMELS dataset and observed that the model's training speed was significantly slower, with each epoch taking 30 times longer than LSTM. We welcome the community to benchmark on shared datasets with transparent basin list and input list.

Conclusions

In this work, we compared a vanilla Transformer encoder and a modified Transformer to the current state-of-the-art model, LSTM, on the CAMELS benchmark dataset. The vanilla Transformer seems to miss some critical functionality so that it is not optimal for simulating streamflow. The modified Transformer with no recurrent connection obtains slightly more favorable results (albeit only with a scale advantage) than LSTM. These results already represent rare competitive results to the LSTM in rigorous community-shared benchmarks. This means we can technically continue to search for better architecture to further improve its performance and suitability for natural physical systems, as the current setup may not yet be optimal. Nevertheless, the differences are overall small between the models, and we may already be very close to the optimum for this dataset with this test. On the one hand, we do not expect any architectural change to result in any significant improvement (to be more precise, on the order of 0.02 for KGE).

An expansion of the dataset, e.g., better descriptors or forcing dataset will be required to obtain substantial prediction improvements. On the other hand, the modified Transformer architecture is a viable alternative to LSTM and may find advantages for larger datasets in the future. The transformer architecture's advantages may not reside with sequential information extraction but with serving as a foundational model to capture the joint distribution, accumulate knowledge and extract deep, abstract and complex concepts. These advantages should be explored in future hydrologic and geoscientific research.

Funding

JL is supported by National Science Foundation EAR-2221880 and the U.S. Department of Energy under award DE-SC0016605. KL and CS are supported by Cooperative Institute for Research to Operations in Hydrology (CIROH) through the NOAA Cooperative Agreement with The University of Alabama (NA22NWS4320003) and subaward A22-0307-S003.

Data availability and sharing

All data used for the analysis in this work is publicly available and is cited respectively. An updated zenodo code release will be uploaded upon manuscript acceptance.

Competing Interests

CS and KL have financial interests in HydroSapient, Inc., a company which could potentially benefit from the results of this research. This interest has been reviewed by the University in accordance with its Individual Conflict of Interest policy, for the purpose of maintaining the objectivity and the integrity of research at The Pennsylvania State University.

Bibliography

Abed, M., Imteaz, M. A., Ahmed, A. N., & Huang, Y. F. (2022). A novel application of transformer neural network (TNN) for estimating pan evaporation rate. *Applied Water Science*, *13*(2), 31. https://doi.org/10.1007/s13201-022-01834-w

Aboelyazeed, D., Xu, C., Hoffman, F. M., Liu, J., Jones, A. W., Rackauckas, C., et al. (2023). A differentiable, physics-informed ecosystem modeling and learning framework for large-scale inverse problems: demonstration with photosynthesis simulations. *Biogeosciences*, *20*(13), 2671–2692. https://doi.org/10.5194/bg-20-2671-2023

Addor, N., Newman, A. J., Mizukami, N., & Clark, M. P. (2017). Catchment Attributes and MEteorology for Large-Sample studies (CAMELS) version 2.0 [Data set]. Dataset, Boulder, CO: UCAR/NCAR. https://doi.org/10.5065/D6G73C3Q

Afzaal, H., Farooque, A. A., Abbas, F., Acharya, B., & Esau, T. (2020). Groundwater estimation from major physical hydrology components using artificial neural networks and deep learning. *Water*, *12*(1), 5. https://doi.org/10.3390/w12010005

Amanambu, A. C., Mossa, J., & Chen, Y.-H. (2022). Hydrological Drought Forecasting Using a Deep Transformer Model. *Water*, *14*(22), 3611. https://doi.org/10.3390/w14223611

Anderson, M. G., & McDonnell, J. J. (2005). Sacramento Soil Moisture Accounting Model (SAC-SMA). In *Encyclopedia of Hydrological Sciences*. Chichester, UK: John Wiley & Sons, Ltd. https://doi.org/10.1002/0470848944.hsa279

Beitner, J. (2020, October 12). PyTorch Forecasting. Retrieved from https://github.com/jdb78/pytorch-forecasting

Botterill, T. E., & McMillan, H. K. (2023). Using machine learning to identify hydrologic signatures with an encoder–decoder framework. *Water Resources Research*, *59*(3), e2022WR033091. https://doi.org/10.1029/2022WR033091

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., et al. (2023, March 27). Sparks of Artificial General Intelligence: Early experiments with GPT-4. arXiv. https://doi.org/10.48550/arXiv.2303.12712

Burnash, R. J. C., Ferral, R. L., & McGuire, R. A. (1973). *A generalized streamflow simulation system: conceptual modeling for digital computers*. Sacramento, Calif.: U. S. Dept. of Commerce, National Weather Service.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020, May 28). End-to-End Object Detection with Transformers. arXiv. https://doi.org/10.48550/arXiv.2005.12872

Castangia, M., Grajales, L. M. M., Aliberti, A., Rossi, C., Macii, A., Macii, E., & Patti, E. (2023). Transformer neural networks for interpretable flood forecasting. *Environmental Modelling & Software*, *160*, 105581. https://doi.org/10.1016/j.envsoft.2022.105581

Dai, S., Li, L., & Li, Z. (2019). Modeling vehicle interactions via modified LSTM models for trajectory prediction. *IEEE Access*, *7*, 38287–38296. https://doi.org/10.1109/ACCESS.2019.2907000

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., & Kaiser, Ł. (2019, March 5). Universal Transformers. arXiv. https://doi.org/10.48550/arXiv.1807.03819

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, May 24). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv. https://doi.org/10.48550/arXiv.1810.04805

Fang, K., Shen, C., Kifer, D., & Yang, X. (2017). Prolongation of SMAP to spatiotemporally seamless coverage of continental U.S. using a deep learning neural network. *Geophysical Research Letters*, *44*(21), 11,030-11,039. https://doi.org/10.1002/2017gl075619

Fang, K., Kifer, D., Lawson, K., Feng, D., & Shen, C. (2022). The data synergy effects of time-series deep learning models in hydrology. *Water Resources Research*, *58*(4), e2021WR029583. https://doi.org/10.1029/2021WR029583

Feng, D., Fang, K., & Shen, C. (2020). Enhancing streamflow forecast and extracting insights using long-short term memory networks with data integration at continental scales. *Water Resources Research*, *56*(9), e2019WR026793. https://doi.org/10.1029/2019WR026793

Feng, D., Lawson, K., & Shen, C. (2021). Mitigating prediction error of deep learning streamflow models in large data-sparse regions with ensemble modeling and soft data. *Geophysical Research Letters*, *48*(14), e2021GL092999. https://doi.org/10.1029/2021GL092999

Feng, D., Liu, J., Lawson, K., & Shen, C. (2022). Differentiable, learnable, regionalized process-based models with multiphysical outputs can approach state-of-the-art hydrologic prediction accuracy. *Water Resources Research*, *58*(10), e2022WR032404. https://doi.org/10.1029/2022WR032404

Feng, D., Beck, H., Lawson, K., & Shen, C. (2023). The suitability of differentiable, physics-informed machine learning hydrologic models for ungauged regions and climate change impact assessment. *Hydrology and Earth System Sciences*, *27*(12), 2357–2373. https://doi.org/10.5194/hess-27-2357-2023

Feng, D., Beck, H., de Bruijn, J., Sahu, R. K., Satoh, Y., Wada, Y., et al. (2023, October 5). Deep dive into global hydrologic simulations: Harnessing the power of deep learning and physics-informed differentiable models (δHBV-globe1.0-hydroDL). Geoscientific Model Development Discussions. https://doi.org/10.5194/gmd-2023-190

Ghobadi, F., & Kang, D. (2022). Improving long-term streamflow prediction in a poorly gauged basin using geo-spatiotemporal mesoscale data and attention-based deep learning: A comparative study. *Journal of Hydrology*, *615*, 128608. https://doi.org/10.1016/j.jhydrol.2022.128608

Gupta, H. V., Kling, H., Yilmaz, K. K., & Martinez, G. F. (2009). Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *Journal of Hydrology*, *377*(1), 80–91. https://doi.org/10.1016/j.jhydrol.2009.08.003

Hochreiter, S. (1991). *Untersuchungen zu dynamischen neuronalen Netzen*. Institut fur Informatik, Technische Universitat, Munchen, 1-150. Retrieved from https://www.semanticscholar.org/paper/Untersuchungen-zu-dynamischen-neuronalen-Netzen-Hochreiter/3f3d13e95c25a8f6a753e38dfce88885097cbd43

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hochreiter, S., Bengio, Y., Frasconi, P., & Jürgen Schmidhuber. (2001). Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies. In S. C. Kremer & J. F. Kolen (Eds.), *A Field Guide to Dynamical Recurrent Neural Networks* (pp. 237–244). Piscataway, NJ, USA: IEEE Press.

Hrachowitz, M., & Clark, M. P. (2017). HESS Opinions: The complementary merits of competing modelling philosophies in hydrology. *Hydrology and Earth System Sciences*, *21*(8), 3953–3973. https://doi.org/10.5194/hess-21-3953-2017

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., et al. (2018, December 12). Music Transformer. arXiv. https://doi.org/10.48550/arXiv.1809.04281

Karita, S., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., et al. (2019). A Comparative Study on Transformer vs RNN in Speech Applications. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (pp. 449–456). https://doi.org/10.1109/ASRU46091.2019.9003750

Konapala, G., Kao, S.-C., Painter, S. L., & Lu, D. (2020). Machine learning assisted hybrid models can improve streamflow simulation in diverse catchments across the

conterminous US. *Environmental Research Letters*, *15*(10), 104022. https://doi.org/10.1088/1748-9326/aba927

Koya, S. R., & Roy, T. (2023, May 20). Temporal Fusion Transformers for Streamflow Prediction: Value of Combining Attention with Recurrence. arXiv. https://doi.org/10.48550/arXiv.2305.12335

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., & Nearing, G. (2019). Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences*, *23*(12), 5089–5110. https://doi.org/10.5194/hess-23-5089-2019

Kratzert, F., Klotz, D., Hochreiter, S., & Nearing, G. S. (2021). A note on leveraging synergy in multiple meteorological data sets with deep learning for rainfall–runoff modeling. *Hydrology and Earth System Sciences*, *25*(5), 2685–2703. https://doi.org/10.5194/hess-25-2685-2021

Kratzert, F., Nearing, G., Addor, N., Erickson, T., Gauch, M., Gilon, O., et al. (2023). Caravan - A global community dataset for large-sample hydrology. *Scientific Data*, *10*(1), 61. https://doi.org/10.1038/s41597-023-01975-w

Kratzert, F., Gauch, M., Klotz, D., & Nearing, G. (2024). HESS Opinions: Never train an LSTM on a single basin. *Hydrology and Earth System Sciences Discussions*, 1–19. https://doi.org/10.5194/hess-2023-275

Li, Y., & Yang, J. (2019). Hydrological time series prediction model based on attention-LSTM neural network. In *Proceedings of the 2019 2nd International Conference on Machine Learning and Machine Intelligence* (pp. 21–25). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3366750.3366756

Liu, C., Liu, D., & Mu, L. (2022). Improved Transformer Model for Enhanced Monthly Streamflow Predictions of the Yangtze River. *IEEE Access*, *10*, 58240–58253. https://doi.org/10.1109/ACCESS.2022.3178521

585 Liu, J., Rahmani, F., Lawson, K., & Shen, C. (2022). A multiscale deep learning model for soil
586 moisture integrating satellite and in situ data. *Geophysical Research Letters*, *49*(7),
587 e2021GL096847. https://doi.org/10.1029/2021GL096847
588 Liu, J., Hughes, D., Rahmani, F., Lawson, K., & Shen, C. (2023). Evaluating a global soil
589 moisture dataset from a multitask model (GSM3 v1.0) with potential applications for crop
590 threats. *Geoscientific Model Development*, *16*(5), 1553–1567.
591 https://doi.org/10.5194/gmd-16-1553-2023
592 Maurer, E. P., Wood, A. W., Adam, J. C., Lettenmaier, D. P., & Nijssen, B. (2002). A long-term
593 hydrologically based dataset of land surface fluxes and states for the conterminous
594 United States. *Journal of Climate*, *15*(22), 3237–3251. https://doi.org/10.1175/1520-
595 0442(2002)015<3237:ALTHBD>2.0.CO;2
596 Nash, J. E., & Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I —
597 A discussion of principles. *Journal of Hydrology*, *10*(3), 282–290.
598 https://doi.org/10.1016/0022-1694(70)90255-6
599 Newman, A. J., Sampson, K., Clark, M. P., Bock, A., Viger, R. J., & Blodgett, D. (2014). *A large-*
600 *sample watershed-scale hydrometeorological dataset for the contiguous USA. Boulder*.
601 https://doi.org/10.5065/D6MW2F4D
602 Papacharalampous, G., Tyralis, H., & Koutsoyiannis, D. (2018). One-step ahead forecasting of
603 geophysical processes within a purely statistical framework. *Geoscience Letters*, *5*(1),
604 12. https://doi.org/10.1186/s40562-018-0111-1
605 Pasquiou, A., Lakretz, Y., Hale, J., Thirion, B., & Pallier, C. (2022, July 7). Neural Language
606 Models are not Born Equal to Fit Brain Data, but Training Helps. arXiv.
607 https://doi.org/10.48550/arXiv.2207.03380
608 Raganato, A., & Tiedemann, J. (2018). An analysis of encoder representations in transformer-
609 based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP:*

Analyzing and Interpreting Neural Networks for NLP (pp. 287–297). Brussels, Belgium: Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-5431

Rahmani, F., Shen, C., Oliver, S., Lawson, K., & Appling, A. (2021). Deep learning approaches for improving prediction of daily stream temperature in data-scarce, unmonitored, and dammed basins. *Hydrological Processes*, *35*(11), e14400. https://doi.org/10.1002/hyp.14400

Rahmani, F., Lawson, K., Ouyang, W., Appling, A., Oliver, S., & Shen, C. (2021). Exploring the exceptional performance of a deep learning stream temperature model and the value of streamflow data. *Environmental Research Letters*, *16*(2), 024025. https://doi.org/10.1088/1748-9326/abd501

Rajpurkar, P., Jia, R., & Liang, P. (2018). Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 784–789). Melbourne, Australia: Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-2124

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., et al. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, *118*(15), e2016239118. https://doi.org/10.1073/pnas.2016239118

Shen, C., Appling, A. P., Gentine, P., Bandai, T., Gupta, H., Tartakovsky, A., et al. (2023). Differentiable modelling to unify machine learning and physical models for geosciences. *Nature Reviews Earth & Environment*, *4*(8), 552–567. https://doi.org/10.1038/s43017-023-00450-9

Sun, A. Y., Jiang, P., Mudunuru, M. K., & Chen, X. (2021). Explore spatio-temporal learning of large sample hydrology using graph neural networks. *Water Resources Research*, *57*(12), e2021WR030394. https://doi.org/10.1029/2021WR030394

Thornton, P. E., Running, S. W., & White, M. A. (1997). Generating surfaces of daily meteorological variables over large regions of complex terrain. *Journal of Hydrology*, *190*(3), 214–251. https://doi.org/10.1016/S0022-1694(96)03128-9

Tsai, W.-P., Feng, D., Pan, M., Beck, H., Lawson, K., Yang, Y., et al. (2021). From calibration to parameter learning: Harnessing the scaling effects of big data in geoscientific modeling. *Nature Communications*, *12*(1), 5988. https://doi.org/10.1038/s41467-021-26107-z

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017, December 5). Attention Is All You Need. arXiv. https://doi.org/10.48550/arXiv.1706.03762

Wunsch, A., Liesch, T., & Broda, S. (2022). Deep learning shows declining groundwater levels in Germany until 2100 due to climate change. *Nature Communications*, *13*(1), 1221. https://doi.org/10.1038/s41467-022-28770-2

Xia, Y., Mitchell, K., Ek, M., Sheffield, J., Cosgrove, B., Wood, E., et al. (2012). Continental-scale water and energy flux analysis and validation for the North American Land Data Assimilation System project phase 2 (NLDAS-2): 1. Intercomparison and application of model products. *Journal of Geophysical Research: Atmospheres*, *117*(D3). https://doi.org/10.1029/2011JD016048

Xiang, Z., & Demir, I. (2020). Distributed long-term hourly streamflow predictions using deep learning – A case study for State of Iowa. *Environmental Modelling & Software*, *131*, 104761. https://doi.org/10.1016/j.envsoft.2020.104761

Xu, Z., Wang, S., Stanislawski, L. V., Jiang, Z., Jaroenchai, N., Sainju, A. M., et al. (2021). An attention U-Net model for detection of fine-scale hydrologic streamlines. *Environmental Modelling & Software*, *140*, 104992. https://doi.org/10.1016/j.envsoft.2021.104992

Yang, E., Li, M. D., Raghavan, S., Deng, F., Lang, M., Succi, M. D., et al. (2023). Transformer versus traditional natural language processing: how much data is enough for automated

radiology report classification? *The British Journal of Radiology*, 20220769. https://doi.org/10.1259/bjr.20220769

Yang, H., Zhang, Z., Liu, X., & Jing, P. (2023). Monthly-scale hydro-climatic forecasting and climate change impact evaluation based on a novel DCNN-Transformer network. *Environmental Research*, *236*, 116821. https://doi.org/10.1016/j.envres.2023.116821

Yilmaz, K. K., Gupta, H. V., & Wagener, T. (2008). A process-based diagnostic approach to model evaluation: Application to the NWS distributed hydrologic model. *Water Resources Research*, *44*(9). https://doi.org/10/fpvsgb

Yin, H., Guo, Z., Zhang, X., Chen, J., & Zhang, Y. (2022). RR-Former: Rainfall-runoff modeling based on Transformer. *Journal of Hydrology*, *609*, 127781. https://doi.org/10.1016/j.jhydrol.2022.127781

Yin, H., Zhu, W., Zhang, X., Xing, Y., Xia, R., Liu, J., & Zhang, Y. (2023). Runoff predictions in new-gauged basins using two transformer-based models. *Journal of Hydrology*, *622*, 129684. https://doi.org/10.1016/j.jhydrol.2023.129684

Zhang, Y., Chen, G., Yu, D., Yao, K., Khudanpur, S., & Glass, J. (2016). Highway long short-term memory RNNs for distant speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5755–5759). IEEE. https://doi.org/10.1109/ICASSP.2016.7472780

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 35, pp. 11106–11115). Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/17325