Transferring hydrologic data across continents -- leveraging US data to improve hydrologic prediction in other countries

Kai Ma[1,2], Dapeng Feng[2], Kathryn Lawson[2], Wen-Ping Tsai[2], Chuan Liang[1], Xiaorong Huang[1], Ashutosh Sharma[2], Chaopeng Shen[2*]

[1]State Key Laboratory of Hydraulics and Mountain River Engineering, Sichuan University, Sichuan, China

[2]Civil and Environmental Engineering, Pennsylvania State University, University Park, PA, USA

Abstract

There is a drastic geographic imbalance in available global streamflow gauge and catchment property data, with additional large variations in data characteristics, so that models calibrated in one region cannot normally be migrated to another. Currently in these regions, non-transferable machine learning models are habitually trained over small local datasets. Here we show that transfer learning (TL), in the sense of weights initialization and weights freezing, allows long short-term memory (LSTM) streamflow models that were trained over the Conterminous United States (CONUS, the source dataset) to be transferred to catchments on other continents (the target regions), without the need for extensive catchment attributes. We demonstrate this possibility for regions where data are dense (664 basins in the UK), moderately dense (49 basins in central Chile), and where data are scarce and only globally-available attributes are available (5 basins in China). In both China and Chile, the TL models significantly elevated model performance compared to locally-trained models. The benefits of TL increased with the amount of available data in the source dataset, but even 50-100 basins from the CONUS dataset provided significant value for TL. The benefits of TL were greater than pre-training LSTM using the outputs from an uncalibrated hydrologic model. These results suggest hydrologic data around the world have commonalities which could be leveraged by deep learning, and significant synergies can be had with a simple modification of the currently predominant workflows, greatly expanding the reach of existing big data. Finally, this work diversified existing global streamflow benchmarks.

Key points:

1. Basins in the world can be well modeled by transferring a deep network trained in the US and tuning it locally, altering common workflows

2. The benefits of TL increased with the amount of data in the source dataset, and showed better performance than pre-training with a hydrologic model.

3. This work greatly expands the reach of deep learning and adds to the value of existing big data, and calls for synergy of global datasets.

[1] * corresponding author: Chaopeng Shen, cshen@engr.psu.edu

1. Introduction

There is a great deal of geographic imbalance in global hydrologic datasets, especially streamflow and water quality gauges. While the US and Europe are blessed with thousands of gaging stations and open access to data, other parts of the world including Asia, Africa, South America, and Oceania have much sparser gauge networks for logistical, economic, or political reasons (Fekete & Vörösmarty, 2007; Do et al., 2017). Apart from streamflow gauges, these regions also lack data on physiographic attributes such as geology and soil depth. Nevertheless, climate change is stressing these parts of the world, and accurate hydrologic simulations are needed for these regions just as much, or even more than for data-rich regions.

Catchments across the world are often perceived as being unique from each other, requiring customized development for each basin (Teutschbein & Seibert, 2012). As a rule of thumb, when we create process-based hydrologic models, our development effort scales roughly linearly to the modeled area, computational effort scales linearly at best, and accuracy is unrelated to the number of basins modeled. It is typically difficult to apply knowledge gained from one basin to another, as parameters or experiences do not transfer easily. As a result, although there have been calls for hydrologic studies to transcend the uniqueness of places (McDonnell et al., 2007), success at modeling some basins does not in general translate into equivalent success or reduced effort for basins in other continents.

Recently, data-driven hydrologic models, especially those based on the deep learning algorithm of long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997), have shown strong skills in learning streamflow dynamics for forward runs and forecasting (Feng et al., 2020; Kratzert et al., 2020; Li et al., 2020). Such performance has benefited from the availability of big data uniquely available over the conterminous United States (CONUS). In other parts of the world, however, we could not apply these same techniques, due to shortage of streamflow gauge data. Moreover, these

techniques require uniform input variables that not only have the same physical concepts but also roughly consistent characteristics, which makes it difficult to apply outside of a unified dataset like CAMELS (Addor et al., 2017). Across different continents, climate forcing data and static attributes were collected from different sources and have different characteristics, e.g., biases. Therefore, even if we were to, despite all odds, compile a global database just like the CAMELS series, it is uncertain if a uniform global model could be trained.

In data-scarce regions (relatively few streamflow gauges and/or short history of observations), there are often daily streamflow measurements that have been recorded for a few years, but not with the consistency and breadth of the CONUS data. For example, screening through the dataset of the Global Runoff Data Centre (GRDC, available at http://grdc.bafg.de), there are a large portion of basins in the world that have <3 years' worth of daily streamflow observations and related data. In these scenarios, machine learning models have still been employed, but mostly in a small-data setting, i.e., a model is fitted to the data from one basin or a few neighboring basins, e.g., Zhu et al. (2020), Yaseen et al. (2015), Liang et al. (2018), Bowes et al. (2019), and de la Fuente et al. (2019). Shen (2018) provided a summary and an entry point into a vast body of work in this realm, with hundreds of papers attesting to the huge demand for solutions. While still effective and typically demonstrated to outperform traditional methods, there is a risk that these models may have not seen sufficient data to thoroughly inform their behavior, which could lead to unexpected performance in future scenarios.

Transfer learning (TL) (Thrun & Pratt, 1998; Pan & Yang, 2010) is a method to migrate knowledge learned from one task to another. Because some different tasks have similar mathematical principles or require similar responses, their representations in a deep learning network are similar. Therefore, it should be possible to train a model with one task and dataset and transfer it to another task, keeping part of the original model while retraining a different portion of the model. Figuratively

82 speaking, a student who has learned piano could learn to play the violin faster than a student
83 inexperienced with any instrument, and one who has mastered roller skating may learn ice skating
84 faster than one who has never skated. TL allows the skills extracted from a large dataset to be
85 reused for different tasks, improving efficiency while greatly expanding the value of large datasets,
86 and it allows some extent of procedure encapsulation. Hence, it has become a highly popular
87 technique in the artificial intelligence community (George et al., 2017; Shen, 2018). Transfer learning
88 has been widely used in language classification and image classification (Y. Zhu et al., 2011; Yosinski
89 et al., 2014). In geosciences and more specifically remote sensing, it is now a popular practice to
90 transfer models from image-recognition datasets, e.g., ImageNet, for remote sensing tasks such as
91 land use classification (Marmanis et al., 2016; X. X. Zhu et al., 2017). Applications in other fields
92 address specific problems where models typically perform well when sufficient source data is
93 available, such as using migration learning for atmospheric dust aerosol particle classification to
94 enhance global climate models (Y. Ma et al., 2015), predicting crop yields with remote sensing data
95 (Wang et al., 2018), fault diagnosis in fog radio access networks (Wu et al., 2020), and cross-mapping
96 cellular and clinical information between single cell and patient data in the medical field (Johnson et
97 al., 2020). However, transferring knowledge from one region to another had not been attempted in
98 hydrologic time series modeling problems, and it had been unclear whether such a transfer would be
99 fruitful.

100 In this work, we applied transfer learning to streamflow modeling to better understand if and when
101 such knowledge transfer could be useful for hydrological time series modeling problems. Our results
102 demonstrate that LSTM models trained over the data-rich CONUS, along with distilled knowledge
103 stored in the form of network weights, can be transferred to data-scarce regions such as Asia and
104 South America to mitigate the limitations of local observations and input attributes. For clarity, the
105 CONUS dataset is called the source dataset, while the basins in the second, transferred location are

106 referred to as the target. We reveal the benefits of TL by comparing models employing TL (hereafter
107 called TL models) with those that are trained using only data from the target region (hereafter called
108 local models) (Section 3.1). We also investigate the impacts of data quantity in the source and target
109 datasets (Section 3.2) and how they compare to alternative initialization methods (Section 3.3).

110 **2. Data and Methods**

111 ## 2.1 Data

112 To examine the effects of TL for different data-density scenarios, we used datasets from four
113 different countries: the original Catchment Attributes and MEteorology for Large-sample Studies
114 (CAMELS) dataset for the contiguous United States (Addor et al., 2017; Newman et al., 2014),
115 CAMELS-GB, a dense dataset for Great Britain based on the CAMELS framework (Coxon et al., 2020),
116 CAMELS-CL, a moderately-dense dataset for Chile based on the CAMELS framework (Alvarez-
117 Garreton et al., 2018), and hydrological and meteorological data for the upper Min River region of
118 China (CHINA-MR) (K. Ma et al., 2020).

119 The datasets all included daily streamflow, precipitation, and temperature data, but specifics varied
120 by dataset (Table S1 in Supporting Information). The CAMELS dataset, containing 671 basins with
121 minimal anthropogenic impacts from across the conterminous United States (CONUS), was used as
122 the source dataset. The daily meteorological data used came from the North American Land Data
123 Assimilation System (NLDAS). CAMELS-CL also includes maximum, minimum, and mean
124 temperatures, and potential evapotranspiration (PET) for 516 basins. For CAMELS-CL, precipitation
125 and PET were available from multiple sources, and we used the Chilean national precipitation data
126 and the PET obtained from MODerate resolution Imaging Spectroradiometer (MODIS). CAMELS-GB
127 provides additional daily hydro-meteorological data including radiation and humidity. The CHINA-MR
128 dataset contains 5 basins larger than 1720 km^2. Atmospheric forcing data from The China

Meteorological Assimilation Driving Datasets (Meng & Wang, 2018) included daily solar radiation and maximum and minimum temperatures while observations included streamflow. Attributes were mainly geographical variables, including area percentage of land use, area on different slopes, and land use landscape metrics, derived from the Harmonized World Soil Database (Fischer et al., 2008) and the Landsat Thematic Mapper and Enhanced Thematic Mapper data (https://landsat.gsfc.nasa.gov/). A complete list of forcings and attributes as inputs to models are presented in Table S1.

We selected all the basins in CAMELS to train the source model (training period from 1985-Oct-01 to 2015-Oct-01). For context, if we trained the model for 10 years (1985-Oct-01 to 1995-Oct-01) and tested it in the next 10 years (1995-Oct-01 to 2005-Oct-01), the median NSE for the test period was 0.72, which was essentially identical to other models reported in the literature relying on NLDAS forcing data (Kratzert et al., 2020). However, when serving as the source dataset, we trained our model with all 30 years' worth of data to maximize extraction of information from the available data. In Chile, we selected 49 basins by screening for basins in CAMELS-CL located in the moderate central Chile, between latitudes 38°S and 42°S, with less than 20% of missing streamflow data from 2000-Jan-01 to 2010-Jan-01. In our preliminary tests, LSTM models were found to give poor results for the extremely dry deserts in the North (which include the driest known place on Earth) and glacier-influenced cold regions in the south, a phenomenon worth future investigation. Given that the scope of this study was to improve LSTM-based modeling, we excluded these regions because the current LSTM models seem to be unsuitable. The 664 basins in the CAMELS-GB dataset, used to represent a data-rich case in the target region, were selected with the same conditions for available streamflow data. As there were only 5 basins in CHINA-MR, all were used.

2.2 Transfer Learning (TL) model based on LSTM

Here we discuss TL in two senses: weights initialization and weights freezing. Deep networks such as LSTM are defined by a basic architecture and a number of weights and nonlinear activation functions across many layers. Upon training, a lot of the information in the training data is stored in the weights, with some parts of the network self-organizing to perform certain functionalities as dictated by the architecture. For example, for a LSTM model trained for streamflow prediction, some of the cell states could be used to track accumulated snow storage. We can migrate all or part of the trained network into another network. Once migrated, we have a choice: to allow all or parts of the weights to further change during training to the target task, or to freeze these weights so that they don't change.

Retraining all the weights allows them all to adapt to the new task, which effectively amounts to *weights initialization* using the source dataset. Compared to training the network from a blank (or cold) initial state, this procedure allows the network to converge faster and requires fewer data points to train. If we freeze the weights, we keep parts of the functionality as is, and force the weights in other layers in the new network to adapt around the frozen part. Typically, freezing weights will allow the network to be trained faster or with less data for the new task compared to weights initialization, because it reduces the number of trainable parameters, though there may be some performance penalties due to reduced flexibility. On the other hand, if the target dataset is small, freezing more weights could reduce the chance of overfitting. A word of caution is that while we can describe what the network as a whole does, it is oftentimes difficult for humans to ascertain exactly what some of the hidden layers do. Even with some specialized visualization techniques, e.g., image reconstruction (Mahendran & Vedaldi, 2015), one could at best obtain approximate answers.

For this work, the TL models were based on an established LSTM architecture which was already successfully tested for predictions of streamflow (Feng et al., 2020) and soil moisture (Fang et al., 2017, 2019; Fang & Shen, 2020). LSTM is a type of Recurrent Neural Network (RNN) that learns from sequential data. The difference from a simple RNN is that LSTM has "memory states" and "gates", which allow it to learn how long to retain the state information, what to forget, and what to output. The forward pass of the LSTM model is described by the following equations:

Input transformation: $$x^t = ReLU(W_I I^t + b_I) \quad (1)$$

Input node: $$g^t = tanh(D(W_{gx} x^t) + b_{gx} + D(W_{gh} h^{t-1}) + b_{gh}) \quad (2)$$

Input gate: $$i^t = \sigma(D(W_{ix} x^t) + b_{ix} + D(W_{ih} h^{t-1}) + b_{ih}) \quad (3)$$

Forget gate: $$f^t = \sigma(D(W_{fx} x^t) + b_{fx} + D(W_{fh} h^{t-1}) + b_{fh}) \quad (4)$$

Output gate: $$o^t = \sigma(D(W_{ox} x^t) + b_{ox} + D(W_{oh} h^{t-1}) + b_{oh}) \quad (5)$$

Cell state: $$s^t = g^t \odot i^t + s^{t-1} \odot f^t$$

Hidden state: $$h^t = tanh(s^t) \odot o^t \quad (7)$$

Output: $$y^t = W_{hy} h^t + b_y \quad (8)$$

where I^t represents the raw inputs for the time step , $ReLU$ is the rectified linear unit, □^{t} is the vector to the LSTM cell, □ is the dropout operator, W's are network weights, b's are bias parameters, □ is the sigmoidal function, ⊙s the element-wise multiplication operator, □^{t} is the output of the input node, i^t, f^t, o^t are the input, forget, and output gates, respectively, h^t represents the hidden states, s^t represents the memory cell states, and □□ is the predicted output. More detailed description can be found in Feng et al. (2020).

There are different forcing and attribute variables in different datasets, and even for the same variable name, variable characteristics (such as biases) are often substantially different across datasets. Therefore, the dimensions of the input transformations will necessarily be different. To accommodate this, we always allowed weights retraining of the linear layer for the input transformation (Equation 1) before the LSTM cell. Keeping this linear layer unfrozen enabled the model to function despite different inputs between the local and source data. Weights retraining was also always allowed for the linear transformation from model hidden states to the target variable.

Beyond weights initialization, we tested three different combinations of freezing some weights while allowing others to be updated in the new local training task (Figure 1). For option TL-a, we only allowed the input and output linear transformation layers to be updated. By unfreezing these layers but keeping all the LSTM unit memory mechanisms (length of memory and the hidden features that are remembered or forgotten) of the transferred model, TL-a amounts to finding a linear combination that turns the inputs in the target region into the known inputs for the source model, and applying a linear adjustment to the outputs. In the other two options (TL-b, TL-c), these LSTM units were also allowed to update in the new training step. TL-b allows some of the weights on the recurrent hidden states to be adjusted. TL-c essentially uses the source dataset only to provide weight initialization, and allows all the weights on the recurrent hidden states to be adjusted. The specific parameters unfrozen for each option are shown in Figure 1.

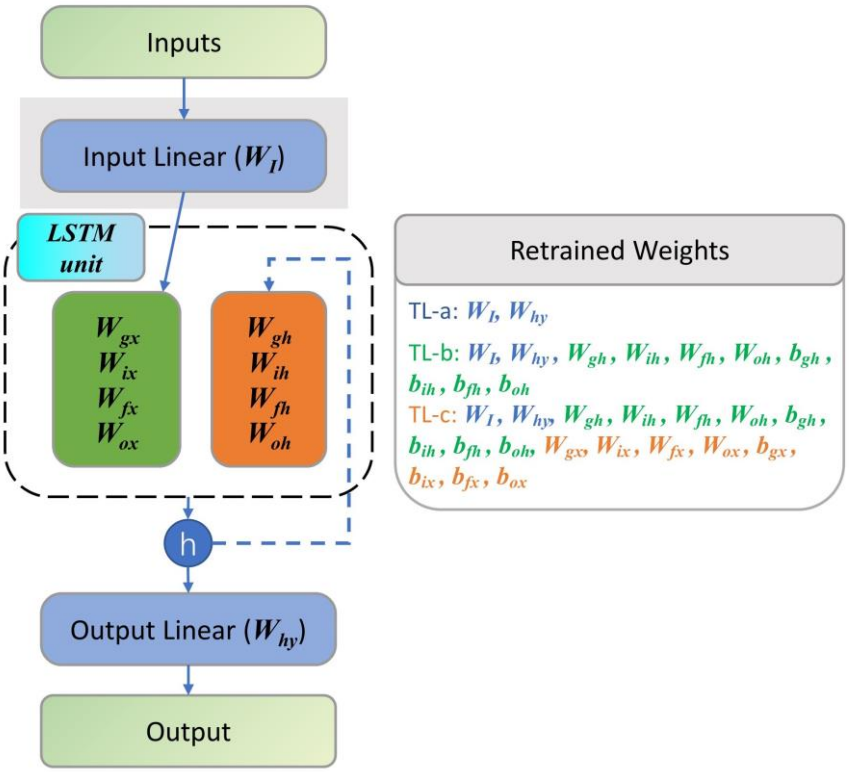


Figure 1. The architecture of LSTM with transfer learning (TL) options. TL-a, TL-b and TL-c add more weights to be tuned, progressively.

2.3 Experiments

2.3.1 The effect of source and target data quantity on TLw

We hypothesized that the TL model would benefit from the wealth of knowledge accumulated in the model weights as they were pre-trained by the source dataset. To test this hypothesis and to understand how the quantity of data in the source dataset influences the effects of TL, we ran experiments where we varied the training data for the pre-trained model from CAMELS. In theory, we should see the benefits of TL increase with increasing amounts of data in the source dataset but give diminishing returns, such that eventually a point will be reached where more data isn't useful. We set the number of basins used in the source data to 10, 50, 100, 300, 500, and all 671, and randomly sampled the CONUS CAMELS basins. We defined the groups with fewer basins as subsets

of the larger ones, so results shouldn't fluctuate between the basin number groups based on which basins were included.

Apart from sparse gauges, the target data-scarce regions may only have observations from a limited period of time, and thus one might be concerned about the quality of the model. To validate the effects of TL when we have different lengths of training data in the target region, we ran a multi-year training scenario and a 1-year training scenario, to emulate regions with a shorter or longer history of observations. For the multi-year training scenario, CAMELS-CL and CAMELS-GB were trained for five years (2000-Jan-01 to 2005-Jan-01) and tested for five years (2005-Jan-01 to 2010-Jan-01), while due to dataset limitations, CHINA-MR models were trained for four years (2009-Jan-01 to 2013-Jan-01), and tested for three years (2013-Jan-01 to 2016-Jan-01). For the 1-year training scenario, China-MR, CAMELS-CL, and CAMELS-GB basins were trained in 2009-Jan-01 to 2010-Jan-01, 2004-Jan-01 to 2005-Jan-01, and 2004-Jan-01 to 2005-Jan-01, respectively. Their testing periods were the same as for the multi-year training scenario.

2.3.2 Comparison of TL to pre-training by a process-based model

It has been suggested that pre-training a machine learning model (determining an improved initialization of the network weights) using outputs from a process-based model could improve the model (Jia et al., 2019; Read et al., 2019). The idea is that the process-based model outputs, even if imperfect or downright flawed, could teach the basic hydrologic inputs and responses and reduce the data demand. Would TL essentially serve the same purpose as a process-based model? To test this, we included such an experiment for comparison.

We created a Soil Water Assessment Tool (SWAT)(Arnold et al., 1998) model for the Min River in China, without any calibration. Note that for the purpose of testing the benefit of physics encoded in SWAT, the SWAT model could not be calibrated, as it would otherwise contain information from the

local observations, which would defeat the purpose of the test. The model was fed with data from 2008-Jan-01 to 2013-Jan-01 with a warm-up period of one year, and the SWAT simulation provided streamflow for the basins from 2009-Jan-01 to 2013-Jan-01 (Arnold et al., 2013; K. Ma et al., 2020), which was then used to train the LSTM model. We then further trained outputs of the warmed-up model on the observed streamflow data from the Min River, which is referred to as SWAT-MR.

Parallel to pretraining the model with process-based model outputs, some have shown that using the output of a process-based model as one of the inputs to LSTM could improve the robustness of the model. We concur, and have found such effects before (Fang et al., 2017), although somewhat minor. However, training a model like that would entail creating the process-based model for all the basins in the source dataset. Hence, we did not explore this option.

2.4 Evaluation metrics

The main metric we used for model evaluation is the Nash–Sutcliffe model efficiency coefficient (NSE) (Nash & Sutcliffe, 1970). Performance evaluations for test periods are reported from an ensemble of five simulations, each with a different random seed.

2.5. Hyperparameters

We manually adjusted the hyperparameters by sensitivity analysis, and they were selected such that each model had optimal performance, for more fair comparison. We tried many combinations, with table S2 listing all the tested hyperparameters and the final values that were chosen. We used a training-instance length of 365 days for all models. The local model's hyperparameters include hidden size and batch size, which are set by sensitivity analysis of the test period performance for different datasets. The batch size chosen for the TL model was the same as for the local model, and the hidden size in TL was consistent with that for the corresponding source model.

3. Results and Discussion

3.1 Performance of TL models in each region

Our results suggest TL is an effective strategy to significantly improve streamflow predictions (Table 1 & Figure 2). For each region, the optimal TL model had better metrics than the local model, and the advantages tended to be larger for smaller target datasets. The 1-year training models of CAMELS-CL showed the highest benefits, with the optimal TL model improving the mean and median NSEs by 0.118 (0.587 to 0.705) and 0.128 (0.597 to 0.725) compared to the local model, respectively. For CHINA-MR, the mean NSE was elevated by 0.039 (0.564 to 0.603) for 1-year training models. With the multi-year training scenario, CHINA-MR also showed the highest TL benefit, where the TL-a improved the mean NSE by 0.068 (0.666 to 0.734) (Because CHINA-MR only had 5 basins, we put more focus on the mean NSE). The TL benefits for CAMELS-GB were smaller, as was expected for a larger target dataset, but were nonetheless non-trivial. The optimal TL models for CAMELS-GB improved the median by 0.033 (0.794 to 0.827) and merely 0.008 (0.853 to 0.861) for 1-year and multi-year training models, respectively.

For both CAMELS-CL and CHINA-MR, the benefits from TL should be substantial enough to be of interest to most modelers. The benefit was less pronounced for CAMELS-GB, but might still be attractive to those who want to have the best possible performance. These results agree with our hypothesis that TL more prominently benefits target regions with smaller datasets: re-training on the local target region fine-tuned the network weights and adapted them to local conditions, but when there was a small target dataset, the model needed to heavily rely on knowledge obtained from the source dataset. The more data were available for the target region, the more adjustments were applied to the network weights, until the amount of data was comparable to the source dataset and the benefit of TL was almost negligible, as with CAMELS-GB.

All the multi-year training models performed better than the corresponding 1-year training models, and the TL benefits for the 1-year training scenario tended to be greater than the multi-year training scenario, although there were exceptions. Across all the regions, multi-year training of the local models improved the median NSE by an average of 0.151 (ranging between 0.059 and 0.233) compared to 1-year training models, while multi-year training of the TL models improved the median NSE by an average of 0.100 (ranging between 0.024 and 0.183). The exception to this trend was that the multi-year training TL models benefited even more than 1-year TL models for CAMELS-GB: the mean NSE of CAMELS-GB models were enhanced by 0.024 (0.769 to 0.794) and 0.043 (0.726 to 0.770) for multi-year and 1-year models, respectively.

The optimal TL options differed for each dataset, but they seemed to be the same for each region regardless of training length. Table 1a shows that the optimal TL options for were TL-c (all weights unfrozen) for CHINA-MR, TL-a (just input and output transformation layers unfrozen) for CAMELS-CL, and TL-b (many, but not all weights unfrozen) for,CAMELS-GB. The difference between different options was substantial for CHINA-MR and CAMELS-CL, but relatively minor for CAMELS-GB. These findings suggest it will be difficult to find the best option *a priori*.

It turned out to be difficult to anticipate the best TL option for different datasets. TL-b was the best option for CAMELS-GB by a very small margin over TL-c, which was largely consistent with our intuition that more local data can be better exploited by models with larger complexity. TL-a was found to be better for CAMELS-CL, which suggests central Chile may be climatologically and hydrologically similar to some basins in CONUS and was sufficient to only perform linear transformations of inputs and outputs. However, TL-c, which was the equivalent of only a weight initialization, was found to be the best option for all CHINA-MR experiments, which countered our intuition that a smaller target dataset would benefit from a partially-frozen model. One potential explanation is that CHINA-MR contains larger basins than the CONUS source basins, which could

have made the routing process comparably more important, and thus obtaining optimal results required the retraining of all the LSTM weights to sufficiently alter the model's memory dynamics. On a side note, the good performance with CHINA-MR and CAMELS-CL suggests that the forcing information for CHINA-MR is potentially of a quality similar to NLDAS.

Table 1. The NSE values of the 5-member ensemble-mean discharge for different training scenarios. Local models were trained only with data from the target region. TL options include TL-a (just input and output transformation layers retrained), TL-b (many, but not all weights retrained), and TL-c (all weights retrained). Bold numbers indicate the best performing model for each category. (b) Comparison between the TL model originally trained over the CONUS (option TL-c), the TL model initialized with SWAT model outputs, and the locally-trained models for CHINA-MR. Because CHINA-MR has only 5 basins, we focus more on the mean.

(a)

Model		NSE_{mean}			NSE_{median}		
		CHINA-MR	CAMELS-CL	CAMELS-GB	CHINA-MR	CAMELS-CL	CAMELS-GB
1-year training	Local	0.564	0.587	0.726	0.571	0.597	0.794
	TL-a	0.597	**0.705**	0.765	0.609	**0.725**	0.824
	TL-b	0.593	0.650	**0.770**	0.620	0.657	**0.827**
	TL-c	**0.603**	0.636	0.767	**0.624**	0.645	0.822
Multi-year training	Local	0.666	0.810	0.769	0.733	0.830	0.853
	TL-a	0.706	**0.845**	0.789	0.708	**0.868**	0.847
	TL-b	0.718	0.820	0.794	0.698	0.840	**0.861**
	TL-c	**0.734**	0.801	**0.796**	**0.749**	0.823	0.859

(b)

Model		NSE_{mean}	NSE_{median}
1-year training	Local	0.564	0.571
	TL (SWAT-MR)	0.580	0.603
	TL-c (CONUS)	**0.603**	**0.624**
Multi-year training	Local	0.666	0.733
	TL (SWAT-MR)	0.693	0.748
	TL-c (CONUS)	**0.734**	**0.749**

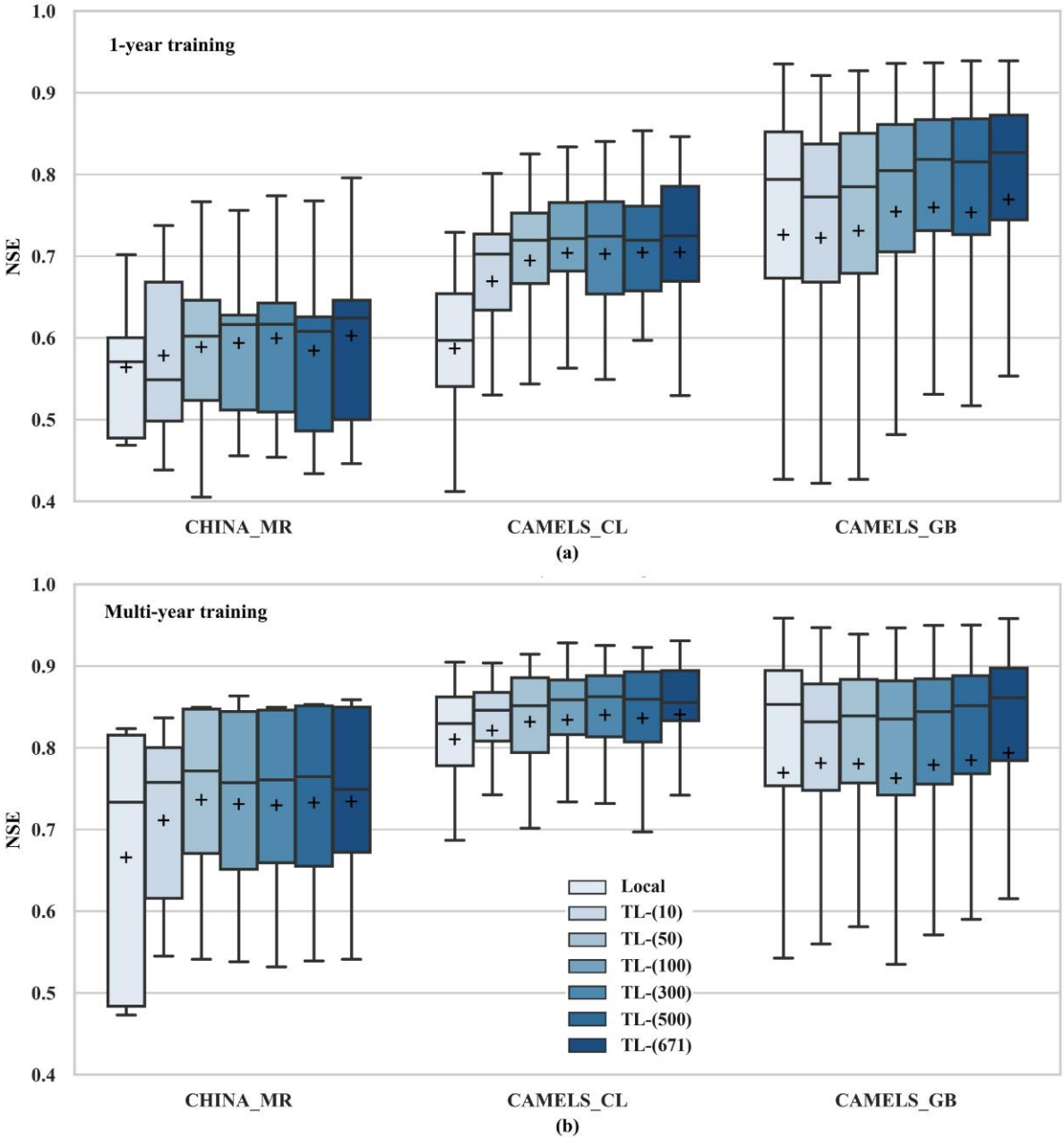


Figure 2. Performance of local and optimal TL models (selected based on Table 1) pre-trained with different numbers of CONUS (source) basins for (a) 1-year training and (b) multi-year training. All the metrics were calculated for the five-member ensemble mean discharge during the test period. Plus symbols indicate mean values. For CHINA-MR, there are only five basins, so the two "whiskers", the two edges of the boxes, and the median line each represent performance for one basin.

3.2 The effect of source data amount on TL

In general, the performance of TL was enhanced as the number of basins in the source dataset increased, though some inherent randomness did exist. We evaluated both 1-year training and multi-year training models (Figure 2). For one year of training, both the mean and median NSE of the TL model trained only on 10 CONUS basins were still higher than those of the local models for CAMELS-CL and CAMELS-GB, suggesting that a relatively small source dataset can already be beneficial. In the multi-year training models, all TL models showed optimal performance when the number of basins was maximized, and the model performance progressively improved with the number of basins. This supports our hypothesis: the increase in the number of training basins enriches the knowledge extracted into the source model, which translates into better model performance in the target region.

As expected, there were diminishing returns, and the gain resulting from increasing the source dataset size became smaller and smaller as the dataset became larger: the initial 50 basins showed the most notable benefit, raising median NSE from 0.629 to 0.720. As the source dataset continued to increase, the benefit per added basin became smaller and smaller, albeit still non-zero if we discount some stochastic results. This is consistent with other results from big data machine learning: the marginal benefit of a larger dataset gradually decreases toward very large sample size, but may be non-zero even for a very large dataset (Sun et al., 2017).

3.3 Comparison with model initialization using a process-based hydrologic model

Our experiments showed that the benefits of TL were larger than what would have been contributed by weights initialization using outputs from SWAT. The initialization by SWAT outputs raised the mean NSE from 0.564 to 0.580 in one year training and from 0.666 to 0.693 for multi-year training, suggesting this approach is useful (Table 1b). Nevertheless, the optimal TL model had higher mean

NSE values of 0.603 for 1-year training and 0.734 for multi-year training. A possible explanation is that the source LSTM model is a more accurate hydrologic model than many process-based models, which has been illustrated in papers cited earlier.

Because initialization can only be done once, the two different approaches cannot accommodate each other. We must also consider the cost of process-based model initialization, which is very high in this case, because we need to create process-based models for each target basin of interest. We cannot possibly implement this method for CAMELS-CL and CAMELS-GB within the scope of this work. Hence, when possible, the process of transferring a model trained in a data-rich region and partially retraining to better fit the local region seems to be a more valuable and efficient approach than initialization by the tested hydrologic model.

4. Conclusion

We introduced a transfer learning scheme to leverage information from data-rich regions to mitigate the limitations of small data sets and incomplete input attributes in data-scarce regions on different continents. Trained on the CAMELS dataset over the CONUS, our LSTM model was transferred to data-scarce regions in Asia, Europe, and South America to provide high-accuracy streamflow predictions. There is tremendous value in the transfer learning procedure, as a huge number of basins around the world with only a few years' worth of local observations are now amenable to accurate modeling with deep learning.

These results suggest that hydrologic dynamics around the world, while often perceived as unique, have commonalities that could be leveraged by modelers across different continents. It also means that enticing rewards in terms of model performance are "right at the fingertips" of the steadily-rising amount of streamflow forecasters in data-scarce regions who employ LSTM on small datasets. Multiple transfer learning options are possible, and the choices need to be evaluated for each target

region's use cases. This work suggests modelers across the world can and should look beyond their watersheds or even their continents for useful data. Efforts such as the Global Runoff Data Center and the CAMELS dataset series are highly meritorious, and could be leveraged for these efforts. A global synergy, which was not envisioned before, is now possible with deep learning frameworks.

Acknowledgments

KM was supported by the China Scholarship Council for 1 year study at the Pennsylvania State University. AS and CS were supported by US National Science Foundation Award OAC #1940190. The LSTM code used in this work can be accessed at http://doi.org/10.5281/zenodo.3993880. Data for CAMELS can be downloaded at https://ral.ucar.edu/solutions/products/camels. Data for CAMELS-CL can be downloaded at http://www.cr2.cl/camels-cl/. Data for CAMELS-GB can be downloaded at https://doi.org/10.5285/8344e4f3-d2ea-44f5-8afa-86d2987543a9. Atmospheric data for CHINA-MR can be downloaded at http://www.cmads.org/.

References

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

Alvarez-Garreton, C., Mendoza, P. A., Boisier, J. P., Addor, N., Galleguillos, M., Zambrano-Bigiarini, M., et al. (2018). The CAMELS-CL dataset: catchment attributes and meteorology for large sample studies – Chile dataset. *Hydrology and Earth System Sciences*, *22*(11), 5817–5846. https://doi.org/10/gfnxgz

Arnold, J. G., Srinivasan, R., Muttiah, R. S., & Williams, J. R. (1998). Large area hydrologic modeling and assessment PART I: Model Development. *Journal of the American Water Resources*

Association, *34*(1), 73–89. https://doi.org/10/cfhxn7

Arnold, J. G., Kiniry, J. R., Srinivasan, R., Williams, J. R., Haney, E. B., & Neitsch, S. L. (2013). *SWAT 2012 Input/Output Documentation*. Texas Water Resources Institute. Retrieved from https://hdl.handle.net/1969.1/149194.

Bowes, B. D., Sadler, J. M., Morsy, M. M., Behl, M., & Goodall, J. L. (2019). Forecasting Groundwater Table in a Flood Prone Coastal City with Long Short-term Memory and Recurrent Neural Networks. *Water*, *11*(5), 1098. https://doi.org/10/ggkdcg

Coxon, G., Addor, N., Bloomfield, J. P., Freer, J., Fry, M., Hannaford, J., et al. (2020). *CAMELS-GB: Hydrometeorological time series and landscape attributes for 671 catchments in Great Britain* (preprint). Hydrology and Soil Science – Hydrology. https://doi.org/10.5194/essd-2020-49

Do, H. X., Westra, S., & Leonard, M. (2017). A global-scale investigation of trends in annual maximum streamflow. *Journal of Hydrology*, *552*, 28–43. https://doi.org/10/gbz9h9

Fang, K., & Shen, C. (2020). Near-real-time forecast of satellite-based soil moisture using long short-term memory with an adaptive data integration kernel. *Journal of Hydrometeorology*, *21*(3), 399–413. https://doi.org/10/ggj669

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/gcr7mq

Fang, K., Pan, M., & Shen, C. (2019). The value of SMAP for long-term soil moisture estimation with the help of deep learning. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(4), 2221–2233. https://doi.org/10/gghp3v

Fekete, B. M., & Vörösmarty, C. J. (2007). The current status of global river discharge monitoring and potential new technologies complementing traditional discharge measurements, 8.

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*, 2019WR026793. https://doi.org/10/gg3kgw

Fischer, G., F. Nachtergaele, S. Prieler, H.T. van Velthuizen, L. Verelst, & D. Wiberg. (2008). Global Agro-ecological Zones Assessment for Agriculture (GAEZ 2008). *IIASA, Laxenburg, Austria and FAO, Rome, Italy.*

de la Fuente, A., Meruane, V., & Meruane, C. (2019). Hydrological Early Warning System Based on a Deep Learning Runoff Model Coupled with a Meteorological Forecast. *Water*, *11*(9), 1808. https://doi.org/10/ggkdch

George, D., Shen, H., & Huerta, E. A. (2017). Deep Transfer Learning: A new deep learning glitch classification method for advanced LIGO. *Arxiv Preprint 1706.07446*. Retrieved from http://arxiv.org/abs/1706.07446

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10/bxd65w

Jia, X., Willard, J., Karpatne, A., Read, J., Zwart, J., Steinbach, M., & Kumar, V. (2019). Physics Guided RNNs for Modeling Dynamical Systems: A Case Study in Simulating Lake Temperature Profiles. In Proceedings of the 2019 SIAM International Conference on Data Mining (pp. 558–566). Calgary, Alberta, Canada: Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611975673.63

Johnson, T. S., Yu, C. Y., Huang, Z., Xu, S., Wang, T., Dong, C., et al. (2020). *Diagnostic Evidence GAuge of Single cells (DEGAS): A transfer learning framework to infer impressions of cellular and patient phenotypes between patients and single cells* (preprint). Bioinformatics. https://doi.org/10.1101/2020.06.16.142984

Kratzert, F., Klotz, D., Hochreiter, S., & Nearing, G. S. (2020). *A note on leveraging synergy in multiple meteorological datasets with deep learning for rainfall-runoff modeling* (preprint). Global hydrology/Modelling approaches. https://doi.org/10.5194/hess-2020-221

457 Li, W., Kiaghadi, A., & Dawson, C. (2020). High temporal resolution rainfall–runoff modeling using
458 long-short-term-memory (LSTM) networks. *Neural Computing and Applications*.
459 https://doi.org/10/gg7mn7
460 Liang, C., Li, H., Lei, M., Du, & Qingyun. (2018). Dongting Lake Water Level Forecast and Its
461 Relationship with the Three Gorges Dam Based on a Long Short-Term Memory Network.
462 *Water*, *10*(10), 1389. https://doi.org/10/gfr7pz
463 Ma, K., Huang, X., Liang, C., Zhao, H., Zhou, X., & Wei, X. (2020). Effect of land use/cover changes on
464 runoff in the Min River watershed. *River Research and Applications*, *36*(5), 749–759.
465 https://doi.org/10/gg685z
466 Ma, Y., Gong, W., & Mao, F. (2015). Transfer learning used to analyze the dynamic evolution of the
467 dust aerosol. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *153*, 119–130.
468 https://doi.org/10/gg7mpk
469 Mahendran, A., & Vedaldi, A. (2015). Understanding deep image representations by inverting them.
470 In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5188–
471 5196). Boston, MA, USA: IEEE. https://doi.org/10/gc7rg9
472 Marmanis, D., Datcu, M., Esch, T., & Stilla, U. (2016). Deep Learning Earth Observation Classification
473 Using ImageNet Pretrained Networks. *IEEE Geoscience and Remote Sensing Letters*, *13*(1),
474 105–109. https://doi.org/10/f77jc4
475 McDonnell, J. J., Sivapalan, M., Vaché, K., Dunn, S., Grant, G., Haggerty, R., et al. (2007). Moving
476 beyond heterogeneity and process complexity: A new vision for watershed hydrology. *Water*
477 *Resources Research*, *43*(7), W07301. https://doi.org/10/bqmcpn
478 Meng, X., & Wang, H. (2018). China meteorological assimilation driving datasets for the SWAT model
479 Version 1.1 (2008-2016). *National Tibetan Plateau Data Center*. https://doi.org/10/gg7mqf
480 Nash, J. E., & Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I — A
481 discussion of principles. *Journal of Hydrology*, *10*(3), 282–290. https://doi.org/10/fbg9tm

Newman, A. J., Sampson, K., Clark, M. P., Bock, A., Viger, R. J., & Blodgett, D. (2014). *A large-sample watershed-scale hydrometeorological dataset for the contiguous USA. Boulder*. https://doi.org/10.5065/D6MW2F4D

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. https://doi.org/10/bc4vws

Read, J. S., Jia, X., Willard, J., Appling, A. P., Zwart, J. A., Oliver, S. K., et al. (2019). Process-Guided Deep Learning Predictions of Lake Water Temperature. *Water Resources Research*, *55*(11), 9173–9190. https://doi.org/10/ggkdcn

Shen, C. (2018). A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resources Research*, *54*(11), 8558–8593. https://doi.org/10/gd8cqb

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In *ICCV 2017*. Retrieved from http://arxiv.org/abs/1707.02968

Teutschbein, C., & Seibert, J. (2012). Bias correction of regional climate model simulations for hydrological climate-change impact studies: Review and evaluation of different methods. *Journal of Hydrology*, *456–457*, 12–29. https://doi.org/10/f2zndf

Thrun, S., & Pratt, Lorien. (1998). *Learning to learn*. Norwell, MA: Kluwer Academic Publishers. Retrieved from http://dl.acm.org/citation.cfm?id=296635

Wang, A. X., Tran, C., Desai, N., Lobell, D., & Ermon, S. (2018). Deep Transfer Learning for Crop Yield Prediction with Remote Sensing Data. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS) - COMPASS '18* (pp. 1–5). Menlo Park and San Jose, CA, USA: ACM Press. https://doi.org/10/gg689c

Wu, W., Peng, M., Chen, W., & Yan, S. (2020). Unsupervised Deep Transfer Learning for Fault Diagnosis in Fog Radio Access Networks. *IEEE Internet of Things Journal*, 1–1. https://doi.org/10/gg7mps

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In *NIPS 2014*. Retrieved from http://arxiv.org/abs/1411.1792

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geoscience and Remote Sensing Magazine*, *5*(4), 8 – 36. https://doi.org/10/gfvxjk

Zhu, Y., Chen, Y., Lu, Z., Pan, S. J., Xue, G., Yu, Y., & Yang, Q. (2011). Heterogeneous Transfer Learning for Image Classification. *In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI'11), AAAI Press*, 1304–1309.

Supporting Information

Table S1 Summary of the forcing and attribute variables from CAMELS, CAMELS-GB, CAMELS-CL, and CHINA-MR datasets. Because of the long-list of attributes from CAMELS-GB and CAMELS-CL, we refer the readers to their respective publications for explanations of variable names.

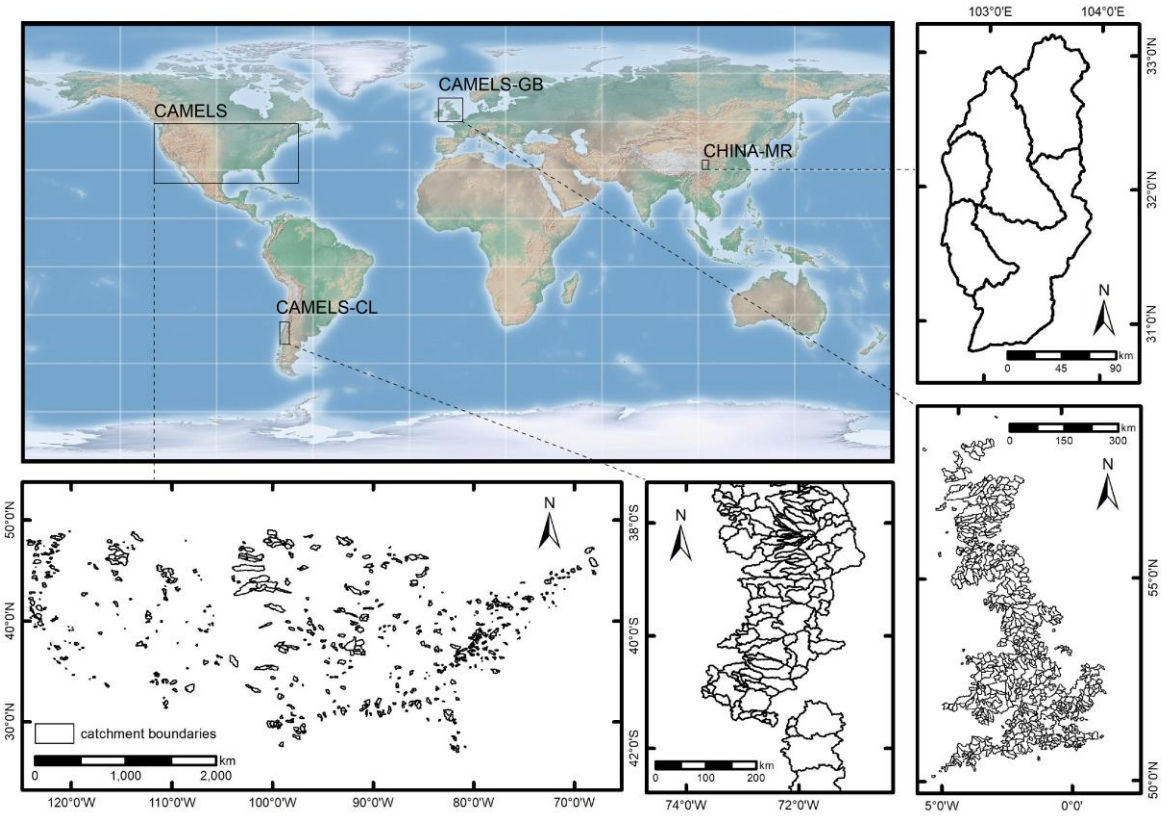
Dataset	Forcing	Attributes
CAMELS	PRCP, SRAD, Tmax, Tmin, Vp, Dayl	elev_mean, slope_mean, area_gages2, frac_forest, lai_max, lai_diff, dom_land_cover_frac, dom_land_cover, root_depth_50, soil_depth_statsgo, soil_porosity, soil_conductivity, max_water_content, geol_1st_class, geol_2nd_class, geol_porostiy, geol_permeability, p_mean, pet_mean, p_seasonality, frac_snow, aridity, high_prec_freq, high_prec_dur, low_prec_freq, low_prec_dur
CAMELS-GB	precipitation, temperature, humidity, shortwave_rad, longwave_rad, windspeed	p_mean, pet_mean, aridity, p_seasonality, discharges, inter_high_perc, q_mean, runoff_ratio, stream_elas, baseflow_index, Q5, Q95, dwood_perc, ewood_perc, grass_perc, shrub_perc, crop_perc, urban_perc, inwater_perc, bares_perc, sand_perc, silt_perc, clay_perc, organic_perc, bulkdens, tawc, porosity_cosby, porosity_hypres, conductivity_cosby, conductivity_hypres, root_depth, soil_depth_pelletier, gauge_lat, gauge_lon, gauge_elev, area, dpsbar, elev_mean, elev_min
CAMELS-CL	precip_cr2met, tmax, tmin, pet_8d_modis	area, elev_mean, slope_mean, nested_inner, geol_class_1st_frac, geol_class_2nd_frac, crop_frac, nf_frac, fp_frac, grass_frac, shrub_frac, wet_frac, imp_frac, lc_barren, snow_frac, lc_glacier, fp_nf_index, forest_frac, dom_land_cover_frac, land_cover_missing, p_mean_cr2met, pet_mean, aridity_cr2met, p_seasonality_cr2met, frac_snow_cr2met, high_prec_freq_cr2met, high_prec_dur_cr2met, low_prec_freq_cr2met, low_prec_dur_cr2met, big_dam, p_mean_spread, q_mean, runoff_ratio_cr2met, stream_elas_cr2met, slope_fdc, baseflow_index, hfd_mean, Q95, Q5, high_q_freq, high_q_dur, low_q_freq, low_q_dur, zero_q_freq, sur_rights_n, interv_degree.
CHINA-MR	precipitation, solar_radiation, tmax, tmin	area, lat, lon, latitude, p_mean. Area percentages of slopes from 0 to 20, 20 to 40, 40 to 80. Area percentages of cultivated land, mixed forest, range shrubland, other forest, pasture, grassland, hay, urban area, rural area, water, bare land*. Largest patch index, area-weighted mean patch fractal dimension, interspersion and Juxtaposition index, Shannon's diversity index, Simpson's diversity index*.

* The area percentage of land use and its landscape metrics are calculated based on the classification of land use for 2000, with the classification and metrics referenced in Ma et al. (2020).

Table S2. Hyperparameter values (chosen/tested) for all models. For the tested values, square brackets indicate the range of values tested, while curly braces indicate the discrete values that were tested.

	Model	Length of training instances	LSTM dropout rate	Mini-batching size	LSTM hidden size	Number of training epochs
Local model	CHINA-MR	365/{100,200, 365}	0.5/{0, 0.3, 0.5}	2/{2,5}	64/{32,64,128}	210/[100,300]
	CAMELS-CL			16/{8,16,32}	64/{32,64,128}	150/[100,300]
	CAMELS-GB			128/{64,128,256}	256/{128,256}	200/[100,500]
	CAMELS			100/{50,100,200}	256/{128,256}	300/[100,500]
Source model	CAMELS (10-500)			5,10,20,60,100/{5,10 ,20,60,100}	256/256	240,240,240,30 0,240,270/[100, 500]
	SWAT-MR			2/{2,5}	64/{32,64,128}	300/[100,300]
TL model	TL for CHINA-MR			2/{2,5}	256/256	240/[100,300]
	TL for CAMELS-CL			16/{8,16,32}	256/256	250/[100,300]
	TL for CAMELS-GB			128/{64,128,256}	256/256	280/[100,500]

529 Figure S1. Maps of catchment datasets across the world used in this study



530