Water Resources Research

RESEARCH ARTICLE
10.1029/2021WR030138

Multi-Task Deep Learning of Daily Streamflow and Water Temperature

J. M. Sadler1, A. P. Appling1, J. S. Read1, S. K. Oliver1, X. Jia1, J. A. Zwart1, and V. Kumar3

1 U.S. Geological Survey, Reston, VA, USA; 2 Department of Computer Science, University of Pittsburgh, Pittsburgh, PA, USA; 3 Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA

Abstract Deep learning (DL) models can accurately predict many hydrologic variables including streamflow and water temperature; however, these models have typically predicted hydrologic variables independently. This study explored the benefits of modeling two interdependent variables, daily average streamflow and daily average stream water temperature, together using multi-task DL. A multi-task scaling factor controlled the relative contribution of the auxiliary variable’s error to the overall loss during training. Our experiments examined the improvement in prediction accuracy of the multi-task approach using paired streamflow and water temperature data from sites across the conterminous United States. Our results showed that for 56 out of 101 sites, the best performing multi-task models performed better overall than the single-task models in terms of Nash-Sutcliffe efficiency for predicting streamflow with single-site models. For 43 sites, the best multi-task, single-site models made no significant difference in predicting streamflow. The multi-task approach had a smaller effect when applied to a model trained with data from 101 sites together, significantly improving performance for only 17 sites. The multi-task scaling factor was consequential in determining to what extent the multi-task approach was beneficial. A naïve selection of this factor led to significantly worse-performing models for 3 of 101 sites when predicting streamflow as the primary variable, and 47 of 53 sites when predicting stream temperature as the primary variable. We conclude that a multi-task approach can make more accurate predictions by leveraging information from interdependent hydrologic variables, but only for some sites, variables, and model configurations.

1. Introduction

Accurately predicting the quality and quantity of water in streams is extremely consequential (Hamlet et al., 2002). Among the many hydrologic variables, water temperature and streamflow are among the most important. Water temperature is an ecological master factor that affects fish abundance, the biogeochemical fate of in-stream nutrients and contaminants, and the emergence of harmful algal blooms (Barnett, 1972; Zeng et al., 2009). Streamflow is another critical factor in stream ecology affecting water quality and stream ecology (Bain et al., 1988; Carlisle et al., 2011). Streamflow forecasts are also important for understanding and forecasting the availability of water for human use, and flood and drought forecasting (Hamlet et al., 2002; Pappenberger et al., 2015; Robertson et al., 2013). Accurate streamflow and water temperature predictions can inform management decisions that have large economic and environmental effects such as: How much water should be released from a reservoir? How much water is expected to be available for agriculture? Will a stream become too warm for fish to thrive or even survive? (Alemu et al., 2011).

Process-based models and simple statistical models have been used to make predictions of hydrologic variables including streamflow and water temperature for many years (Campolo et al., 1999); however, deep learning (DL) models have recently been shown to be an accurate alternative (Bai et al., 2016; Besaw et al., 2010; Kratzert et al., 2018; Kratzert, Klotz, Herrnegger, et al., 2019; Kratzert, Klotz, Shalev, et al., 2019; P. A. Chen et al., 2013; Read et al., 2019; Shen, 2018; Shen et al., 2018; Stammler et al., 2013) advancing beyond simpler machine learning (ML) methods (Hsu et al., 1995). DL models can model complex, non-linear spatial and temporal dependencies in hydrologic processes (Shen, 2018) and have shown predictive skill with out-of-sample predictions (Kratzert, Klotz, Herrnegger, et al., 2019; Read et al., 2019). Furthermore, like other ML models, once DL models are trained they often run more quickly than their detailed, process-based counterparts, a characteristic that can be important for management timelines (Soleimani et al., 2019; Zahura et al., 2020).
Most of the DL models in the hydrology literature, to this point, have been trained to perform only one task or, in other words, predict only one variable (e.g., streamflow (Hu et al., 2018; Kratzert, Klotz, Shalev, et al., 2019), water level (Bowes et al., 2019), or water temperature (Read et al., 2019)). In this paper, we explore a variation of DL for hydrologic modeling: multi-task learning of two related hydrologic variables. In multi-task learning, a single DL model is trained to predict two related variables. Multi-task models have been shown to better generalize and reduce overfitting (Ruder, 2017) and have proven effective in several applications including natural language processing (D. Chen et al., 2014; Seltzer & Droppo, 2013) and computer vision (Girshick, 2015). In the hydrologic sciences, there are many variables that are related to each other, and a multi-task model could learn to leverage information from a related variable to make better predictions. Specifically, when trained to predict multiple variables that are driven by the same underlying physical processes, a multi-task DL model could learn to better represent the shared hydrologic processes and therefore better predict the hydrologic variable that we are interested in. In this paper, we will focus on the variables of streamflow and water temperature.

Multi-task modeling may also be a way to address, at least in part, a limitation of DL models: the need for large observational data sets with which a model can be trained. In the earth sciences, these data can be sparse given the expense with which many observations are obtained. By implementing multi-task learning we may be able to leverage the information from one variable to help improve the prediction of another related variable when it is less observed. An imbalance in amounts of observations can happen in cases where one variable is more expensive to observe compared to another. For example, long-term streamflow is much more expensive to observe compared to water temperature.

We selected streamflow and stream temperature, in part, because changes in streamflow and temperature are part of the energy transfer process. Most generally, heat is transferred to or from a stream segment via advective and non-advective processes. Advective heat transfer manifests in changes to both streamflow and stream temperature. Changes in streamflow can change water temperature and, similarly, changes in water temperature can indicate changes in streamflow. A model trained using streamflow and stream temperature observations has the potential to better learn about the advective energy transfer process and, therefore, improve predictions of both variables.

Some studies have leveraged the relationship between hydrologic variables using data-driven models. Rahmani et al. (2020) and Van Vliet et al. (2011) found that including streamflow as an input for an ML model and statistical model (respectively) for stream temperature prediction improved model performance. Our contribution relative to these is that in multi-task learning streamflow is not used as a model input but as an output along with water temperature. Furthermore, we examine the benefit of multi-task learning for improving streamflow, whereas the Rahmani et al. (2020) and Van Vliet et al. (2011) focused on improvements in water temperature predictions. Kraft et al. (2020) used a multi-task model to predict different parts of the water balance equation including evapotranspiration, snow water equivalent, and groundwater recharge but did not formally assess the benefits of the multi-task approach.

To test the value of multi-task learning for streamflow and water temperature, we evaluated whether a model trained to predict the two variables together made more accurate predictions than a model trained to predict only one variable. We performed this evaluation using sites distributed across the conterminous United States. Additionally, we tested whether multi-task performance showed improved predictions under data sparse conditions for streamflow, which are common given the high cost of streamflow observations. The results of our experiments shed light on the potential of multi-task learning as a means of using process knowledge and making use of available data to improve streamflow and stream temperature predictions.

2. Materials and Methods

2.1. Data

The data for our experiments came from the Catchment Attributes and Meteorology for Large-sample Studies (CAMELS) data set (Addor et al., 2017; Newman et al., 2015). For each site in the CAMELS data set, nearly complete daily observations of streamflow and daily basin-averaged weather forcing data are available from 1 January 1980 through 31 December 2014. Of the 671 total sites in the CAMELS data set, 101 had at least 50 water temperature observations per location collected by the U.S. Geological Survey (USGS) between 1980 and 2014. These basins ranged from 5.4 to 14,269 square kilometers in area. The number of water temperature daily
Standard text content...
The overall gradient is a weighted sum of the main and auxiliary gradients, where $\lambda$ weights the influence of the auxiliary gradient relative to the main gradient. If $\lambda = 0$, the auxiliary gradient has no effect on the update of the LSTM parameters. In this case, the LSTM parameters are only adjusted in proportion to the errors in the main variable and the model is effectively a single-task model. If $\lambda > 0$, errors in both the main and auxiliary variables affect the gradient and the model is a multi-task model. For the remainder of the paper, we refer to $\lambda$ as the “multi-task scaling factor.”

The loss function was based on the Nash Sutcliffe Efficiency (NSE), a common metric for measuring streamflow prediction performance (Nash & Sutcliffe, 1970):

$$\text{NSE} = 1 - \frac{\sum_{t=1}^{T} (\hat{Y}_t (X_t, \theta, \phi_{main}) - Y_t)^2}{\sum_{t=1}^{T} (Y_t - \bar{Y})^2}$$

where $\hat{Y}$ is the predicted value, which is a function of the inputs ($X$) and the model parameters $\theta$ and either $\phi_{main}$ for the main variable or $\phi_{aux}$ for the auxiliary variable. $Y$ is the observed value, $t$ is a time step of the time series, and $T$ is the full time series length. The unmodified NSE values range from negative infinity to one, with one being a perfect model. We used the normalized Nash Sutcliffe Efficiency (NNSE) which ranges from 0 to 1:

$$\text{NNSE} = \frac{1}{2 - \text{NSE}}$$

Because the model training algorithm was minimizing the loss, our loss function was 1 minus the NNSE so that a smaller loss would correspond to a better model and vice versa:

$$\mathcal{L} = 1 - \text{NNSE}$$

### Table 2

| Experiment | Multi-task scaling factors used |
|------------|---------------------------------|
| A          | 1, 5, 10, 20, 40, 80, 160, 320   |
| B          | 0.5, 1, 1.5, 5, 40               |
| C          | 1, 5, 10, 20, 40                |
| D          | 0.001, 0.01, 0.05, 0.01, 0.02, 0.05, 0.1, 0.2, 1 |

### 2.3. Multi-Task Learning Experiments

To test the ability of a multi-task model to improve streamflow and water temperature predictions, we ran four experiments (A–D). In our experiments we compared a single-task model with multi-task models using a variety of multi-task scaling factors (see Equation 3 for the meaning of the multi-task scaling factor, $\lambda$). The multi-task scaling factors explored for each experiment are given in Table 2.

The available data were split into three sets: a training set, a validation set, and a test set. The training data were used in updating the model parameters. The validation data were used to select the multi-task learning factor. Other common hyperparameters were fixed and are described in Section 2.4. The test set was used for final model evaluation and was not used to adjust any part of the model.

The amount of data used for training varied in each experiment (Figure 2). For Experiments A, C, and D we trained individual models for each site in our data set. For Experiment B, we trained a single model with the data from all available sites. All of our experiments were scripted using the Snakemake workflow management system (Köster & Rahmann, 2012).

### 2.3.1. Experiment A: Multi-Task Modeling With Streamflow as the Main Variable Using 25 Years of Training Data and Single-Site Models

In Experiment A, we trained a separate model for each site and we evaluated the multi-task model performances using 25 years of training data (1990–2014). This experiment was intended to represent predicting streamflow at a well-monitored location. In this experiment, streamflow was the main variable and stream temperature was the auxiliary variable. We used 1 January 1980 to 31 December 1984 as a validation set and 1 January 1985 to 31 December 1989 as a test set. We used data from the latter part of the data set (1990–2014) for training because there is substantially more water temperature data available to train the multi-task model in later years compared to earlier years. There were 101 sites that had data in all three partitions for this experiment.
Given that there is a process-based relationship between streamflow and water temperature, it follows that the relationship between the two variables at a given site is expected to affect the performance of the multi-task approach at that site. Because there are catchment characteristics such as groundwater influence, land cover, and climate that could influence the relationship between streamflow and water temperature, we hypothesized that we could use these attributes to predict and/or understand how beneficial a multi-task model would be. To explore the predictability of the improvement of a multi-task model compared to a single-task model, we used a Random Forest model (Breiman, 2001) to predict the improvement of multi-task models compared to single-task models across the 101 sites. We used the catchment characteristics from the CAMELS data set as input features to the Random Forest model. The details of our implementation of the Random Forest model are in Supporting Information S1.

2.3.2. Experiment B: Multi-Task Modeling With Streamflow as the Main Variable Using 25 Years of Training Data and a Multi-Site Model

Experiment B was identical to Experiment A except instead of training separate models for each site, we trained one model using the data from all available sites. This experiment was intended to place the single-site model experiments, which permit meaningfully diverse replication within the available data set, in the context of multi-site models that have been shown to be effective in several other studies (Kratzert, Klotz, Herrnegger, et al., 2019; Rahmani et al., 2020). Training a multi-site model required that the model be given site-specific information that would allow it to differentiate one site from another. Similar to the approach Kratzert, Klotz, Herrnegger, et al. (2019) took, we provided the model catchment characteristic data in addition to the meteorological inputs. The catchment characteristics (see Table S2 in Supporting Information S1) were part of the CAMELS data set (Addor et al., 2017). These characteristics represented characteristics such as soil types, land use/land cover, and catchment size. These data were repeated for each time step and concatenated to the time-varying meteorological inputs.

2.3.3. Experiment C: Multi-Task Modeling With Streamflow as the Main Variable Using 2 Years of Training Data

In Experiment C, we trained the multi-task, single-site models with only 2 years of streamflow data each. This experiment was designed to simulate the situation where only a small period of record is available for a site. We used different 2-year periods to account for differences in events and flow regimes that may or may not be represented in one 2-year period compared to another. The data that we used to train the model in Experiment C were each non-overlapping 2-year period in the years 1990–2014 (i.e., 1 January 1990 to 31 December 1991, 1 January 1992 to 31 December 1993, 1 January 1994 to 31 December 1995, etc.). The validation and test sets were the same as in Experiment A. There were 98 sites that had data in all three partitions for this experiment.
2.3.4. Experiment D: Multi-Task Modeling With Temperature as the Main Variable

In Experiment D, we evaluated the performance of the multi-task models for improving stream temperature predictions. We performed this experiment to see how the multi-task model would perform for a different primary variable with very different dynamics than streamflow. In this experiment, streamflow was the auxiliary variable. Because there were fewer temperature observations available at these sites compared to streamflow observations, especially in the early years of the data record, we used the period 1 January 1980 to 31 December 2011 as the training period, the year 2013 as the validation period, and the year 2014 as the testing period. There were 53 sites that had data in all three partitions for this experiment.

2.4. Model Configuration, Training, and Evaluation Metrics

The training data were divided into samples (or sequences) and batches. A training sample consisted of a 365-day sequence for a single site. Making the sequence length 365 days is important because stream temperature and streamflow have long temporal dependencies that can extend back into previous seasons. A batch consisted of a number of samples. In the single-site experiments (Exp. A, C, and D), all of the 365-day sequences were combined into a single batch. Therefore, in Experiment A the training batch was 25 365-day sequences (one sequence for each year), in Experiment C the batch was two 365-day sequences, and in Experiment D the batch was 32 365-day sequences. In the multi-site experiment (Exp. B) the training batches were made up of one 365-day sequence from each of the 101 sites. Therefore, there were 25 batches in Experiment B, where each batch had 101 samples or sequences. In model training, each batch was used to update the model parameters in turn. For example, the gradient from the first batch was used to update the model parameters and then those parameters were used as the starting point for training using the second batch (if there is a second batch) and so on. After each batch is used to update the model parameters once, one training epoch was complete.

Each model was trained for 400 epochs for Experiments A, B, and D and 200 epochs for Experiment C. We trained the models for fewer epochs in Experiment C to reduce overfitting that was more likely with a much smaller data set (using 2 years of training data instead of 25). For all the models, we used a dropout rate of 0.15 to reduce overfitting. The optimizer used for the model training was the Adam optimizer (Kingma & Ba, 2015).

In each experiment, 30 replicates of each model were trained, each with different random initializations of the model parameters. This was to account for variation in model performance due solely to the random starting parameters. All 30 models were then given inputs from the validation and testing sets to make predictions and those predictions were compared with the observations for evaluation. The LSTM hidden states were initialized as all zeros. All models had 20 hidden units in the LSTM layer. Because our objective was to assess the effect of the multi-task approach, we did not perform a thorough hyperparameter tuning exercise and used the same set of hyperparameters for both the single-task models and the multi-task models. Although there may be a combination of other hyperparameters that may perform better, we found these hyperparameters to work well enough for the purposes of our study.

Both the weather input features (Table 1) and target data (streamflow and stream temperature) were scaled and centered by subtracting the mean from each variable and dividing by the standard deviation; in this way each scaled and centered variable had a mean of zero and a standard deviation of one. Because we were training multi-task models, it was necessary to scale the target data as well as the input data so that both output streamflow and temperature predictions would be on the same scale in model training. All the data were scaled and centered using the standard deviation and means from the training data only.

We compared the multi-task models and the single-task models using NSE. Formal statistical comparisons were done on a site-wise basis using a one-sided Welch’s t-test comparing mean NSE’s (Satterthwaite, 1946). In the t-test, our hypothesis was that, for a given site, a multi-task model is more accurate in terms of NSE than the single-task model. To calculate t in the Welch’s t-test for a given site, n, the equation is

\[
t_n = \frac{NSE_{n,\text{multi-task}} - NSE_{n,\text{single-task}}}{\sqrt{\frac{s^2_{n,\text{multi-task}}}{N} + \frac{s^2_{n,\text{single-task}}}{N}}}
\]  

(7)

SADLER ET AL. 6 of 18
where $\overline{\text{NSE}}$ is the NSE averaged across the 30 replicates, $s$ the standard deviation, and $N$ is the number of replicates. The $t$-score and $p$-value were calculated using Python's SciPy package (Virtanen et al., 2020).

3. Results

3.1. Experiment A: Multi-Task Modeling With Streamflow as the Main Variable Using 25 Years of Training Data and Single-Site Models

3.1.1. Influence of Multi-Task Scaling Factor, $\lambda$

For most of the single-site model configurations, the multi-task model performed better than the single-task model at predicting streamflow (Figures 3a and 3b). The multi-task scaling factor ($\lambda$ from Equation 4) had a large influence on the accuracy of the multi-task models. As the multi-task scaling factor increased from 0 to 80, the performance of the model generally increased. In other words, as the emphasis that the model gave to predicting stream temperature grew, the accuracy of the streamflow predictions increased. However, when multi-task scaling factors exceeded 80, overall model performances in terms of median NSE decreased with further increases in the scaling factor.

The decrease in accuracy in terms of median NSE with multi-task scaling factors greater than 80 in Experiment A likely occurred because the combined gradient used to update the parameters became more influenced by the temperature loss contribution. With the gradient more influenced by the temperature loss, the parameter adjustments in gradient descent were directed more to reducing error in the temperature predictions. When the multi-task scaling factor was high enough, those adjustments reduced error in the temperature predictions at the expense of streamflow predictive performance. This is likely the reason that there was an overall decrease in streamflow prediction accuracy in terms of median NSE with the multi-task scaling factors of 160 and 320.

The multi-task approach made substantial improvements to the worst performing single-task models. The minimum NSE for the single-task models was $-3.5$ whereas the minimum NSE for the multi-task model with a multi-task scaling factor of 80 was $-0.8$.

Figure 3. Influence of multi-task scaling factor on streamflow prediction accuracy in validation set. (a) Performance of single-task and multi-task models in terms of Nash Sutcliffe Efficiency (NSE). (b) The same data as panel (a) but zoomed in so that the variation between the models can be better seen. (c) The number of sites for which each multi-task scaling factor performed the best in terms of median NSE. The dashed horizontal line in panels (a and b) is the median performance of the single-task model. In the boxplots, the bottom of each box is the first quartile of the data (Q1), the middle bar is the median of the data, and the top of each box is the third quartile (Q3). The lower whisker is the lowest point within Q1 – 1.5 × IQR, where IQR is the interquartile range (Q3–Q1). The upper whisker is the highest point within Q3 + 1.5 × IQR. The diamonds are points that lie outside of the whiskers.
Although there was a clear overall pattern in the effect of increasing the multi-task scaling factor on model performance, there was also substantial variability site to site (Figure 3c). The multi-task scaling factor that performed best in terms of median NSE for the largest number of sites in the validation period was 80. However, for each multi-task scaling factor we evaluated, there were at least six sites at which that multi-task scaling factor performed the best. For example, for six sites, the best-performing multi-task scaling factor was 10. For eight sites, the best multi-task scaling factor was 0, meaning that the multi-task approach did not improve overall streamflow predictions compared to the single-task model. Examining the geography, catchment characteristics, and number of temperature observations of these sites showed no obvious reason that the single-task models performed best at these sites.

Given the variability in the performance of the multi-task scaling factors in the validation period across the 101 sites, we selected the best-performing, non-zero multi-task scaling factor for each site to evaluate in the test set. For the eight sites for which the best performing model was the single-task model, we used the second-best multi-task scaling factor. The best multi-task models performed better overall than the single-task models in the test set in terms of NSE (Figure 4). The median NSE of the best multi-task models in the test set was 0.66 compared to 0.61 for the single-task models.

The performance in the validation time period was better (single-task median NSE approximately 0.68) than the test period (single-task median NSE approximately 0.61). We compared the training data's flow characteristics in the upper part of the flow duration curve (q50, q60, q70, q80, q90, q95, q99, where q represents the flow quantile) to the same characteristics in the validation and test period. In this comparison, we found that the flow characteristics of the training data set were more similar to the validation set than to the test set. We believe that this is the likely reason that, overall, the models performed better in the validation period compared to the test period.

With a median NSE of 0.61, the performance of our single-site models is comparable to that of Kratzert, Klotz, Shalev, et al. (2019) whose single-site LSTM models that predicted runoff at CAMELS sites had a median of 0.59. The spatial distribution of the model performance was also similar to Kratzert, Klotz, Shalev, et al. (2019), with the models in North Dakota performing the worst (Figure 5). There was little spatial trend in the improvements made by multi-task models.
For each multi-task scaling factor except for 320, the multi-task performance was better than the single-task models for more than half of the sites (Figure 6), although the sites that improved for one multi-task scaling factor were not necessarily the same ones that improved for another. When the multi-task scaling factor was 20, 40, and 80 and when the best site-specific multi-task scaling factors were used, a majority of sites had statistically significant ($p < 0.05$) improvements (Table 3). A larger multi-task scaling factor resulted in larger improvements at some sites and a larger number of significantly improved sites up until a factor of 160. A larger multi-task scaling factor also resulted in a larger magnitude decrease in performance for some sites. The site-specific best multi-task models performed the same in terms of median root mean squared error compared to when all the sites used the multi-task scaling factor of 80; however, with a multi-task scaling factor of 80, the majority of sites (57 sites) had significant improvements and 17 sites had significantly worse performance. In contrast, only four sites had significantly worse performance when the best site-specific multi-task scaling factor was used.

### 3.1.2. Catchment Characteristics

The Random Forest model was trained to predict the t-score of the NSEs of the multi-task models compared to the single-task models when the multi-task scaling factor was 80. The model had a mean coefficient of determination of 0.2 (standard deviation of 0.29). This performance was measured on sites not present in the training data across 20 random data partitions. The model performance indicates that the model was able to learn general trends from the data that could help it learn to predict the multi-task improvements from the catchment characteristics. One likely limiting factor in the Random Forest model performance was the small data set. There were only 101 sites to draw from. We used a 75/25 test/train split, so only 77 of those could be used for training.

The importance of the Random Forest predictor variables was calculated based on the effect that perturbing a given predictor variable had on the prediction error; a description of each predictor variable is given in Table S2 in Supporting Information S1. The most important variable was base-flow index (Figure 7). Sites with a larger base-flow index will have smoother flow patterns and will have a thermal buffer against extremes in air temperature affecting water temperature compared to sites with a lower base-flow index. This buffer can alter the relative influence of different parts of the energy balance equation that governs the energy transfer process and that links

---

**Figure 5.** Spatial distribution of model Nash Sutcliffe Efficiencies and t-scores in the test set for Experiment A.

**Figure 6.** t-scores of multi-task model in test set of Experiment A. The “best” is the lambda that was best for each site based on the validation set. The distributions are across the 101 sites used in Experiment A. The bottom of each box is the first quartile of the data (Q1), the white circle is the median of the data, and the top of each box is the third quartile (Q3). The lower whisker is the lowest point within Q1 – 1.5 × IQR, where IQR is the interquartile range (Q3–Q1). The upper whisker is the highest point within Q3 + 1.5 × IQR. The dotted lines mark where the t-scores are statistically significantly worse (below zero) or better (above zero) with $p = 0.05$. 

---
the streamflow and water temperature variables. It is reasonable, therefore, that base-flow index is an important variable in predicting how much a multi-task DL approach would benefit a given site.

3.1.3. Seasonal Trends

The improvements on the single-task models made by the multi-task models had a noticeable seasonal trend, with larger improvements in higher flow months and smaller improvements in lower flow months (Figure 8). A greater portion of the training loss occurred in the high-flow periods. This is because the loss function is based on the squared error: a 10% error when the observed value is 40 cubic meters per second (CMS) will have 100× the effect on the overall loss compared to a 10% error when the observed value is 4 CMS. Therefore, the

| Experiment A | Experiment B | Experiment C | Experiment D |
|--------------|--------------|--------------|--------------|
| Number of total sites (n) | 101 | 101 | 101 | 101 |
| Multi-task scaling factor | Number of sites significantly better (p < 0.05) | Number of sites significantly worse (p < 0.05) | Number of sites significantly better (p < 0.05) | Number of sites significantly worse (p < 0.05) |
| 1 | 16 | 3 | 0.5 | 8 | 1 | 1 | 31 | 4 |
| 20 | 53 | 4 | 1 | 7 | 3 | 5 | 35 | 11 |
| 40 | 58 | 7 | 1.5 | 11 | 3 | 10 | 43 | 13 |
| 80 | 56 | 17 | 5 | 5 | 15 | 20 | 45 | 16 |
| 320 | 39 | 39 | 40 | 6 | 72 | 40 | 37 | 20 |
| Best | 56 | 4 | best | 17 | 4 | best | 46 | 4 |

Note. Significance was tested using the students t-test and based on p-value < 0.05. For brevity, not all multi-task scaling factors for all experiments are reported.

![Figure 7](image-url). Feature importance for predicting t-scores for Experiment A when the multi-task scaling factor is 80. Error bars are the standard deviation across 20 random train/test splits.
multi-task (as well as the single-task) model parameters adjusted to reduce error in springtime. These results indicate that the multi-task models were able to exploit the relationship between streamflow and water temperature to further improve model performance in the spring months. The parameter values that adjusted to fit that relationship, however, did not seem to benefit the model in the summer months where, overall, the multi-task models performed worse than the single-task models. This difference indicates that the relationship learned in the high-flow months of the spring did not apply in the summertime.

3.2. Experiment B: Multi-Task Modeling With Streamflow as the Main Variable Using 25 Years of Training Data and a Multi-Site Model

In Experiment B, the overall performance of the single-task, multi-site models (median NSE of 0.69) was better than the single-task, single-site models (median NSE of 0.61; Figure 4). The multi-task approach had a much smaller effect on the model performance in the multi-site models: in the test period, the best site-specific multi-task models had a median NSE of 0.69 (the same as the single-task model). The is likely in large part because the amount of training data for each model in Experiment B was roughly 100× the training data in Experiment A because the Experiment B models learned from data from all of the sites.

Fewer sites had statistically improved performance for the multi-site models. As in Experiment A, we found the multi-task scaling factor for the multi-site model that performed best for each site. The site-specific best multi-task configurations of the multi-site model were significantly better than the single-task configuration for 17 of the 101 sites (see Table 3). When data are available, using a multi-site approach with or without multi-tasking may make larger improvements on model performance compared to using single-site approach with even the best multi-task configuration. Additionally, the best multi-task configuration of the multi-site model were slightly more accurate than the single-task multi-site configuration.

3.3. Experiment C: Multi-Task Modeling With Streamflow as the Main Variable Using 2 Years of Training Data

In general, the multi-task models with the best multi-task scaling factors improved upon the single task models in Experiment C (Figure 4 and Table 3). This experiment was done to test whether a multi-task could be used to improve predictions when only sparse data is available. The effect of the multi-task approach was smaller in the sparse data conditions. The model performance across the single-task models and all the multi-task scaling factors was worse in Experiment C (single-site median 0.45) compared to Experiment A (overall median of 0.61). This was expected given that there was a much smaller amount of training data from which the model could learn the flow dynamics (and temperature dynamics in the case of the multi-task models). There was more variability in the results and the effect of the multi-task model when the data were sparser. This variation was likely due to the variation in training data that the model used, because only 2 years of training data were given and there can be a lot of variability between one 2-year period of streamflow record and another. The multi-task scaling factor that was the best for the most sites in Experiment C was much smaller compared to Experiment A: in Experiment A it was 80, in Experiment C it was 10.

3.4. Experiment D: Multi-Task Modeling With Temperature as the Main Variable

The multi-task approach did not improve upon the single-task model as substantially for predicting water temperature as it did for predicting streamflow (Figure 4). The multi-task scaling factor that was the best for the most sites in Experiment D was much smaller compared to Experiment A: in Experiment A it was 80, in Experiment D it was 0.01. When using the best site-specific multi-task scaling factor, the multi-task models significantly

Figure 8. Monthly t-score distributions in the test set for predicting streamflow using 25 years of training data. The distributions are across the 101 sites used in the experiments. Also shown are the monthly median percent of annual flow across the 101 sites. In the boxplots, the bottom of each box is the first quartile of the data (Q1), the middle bar is the median of the data, and the top of each box is the third quartile (Q3). The lower whisker is the lowest point within Q1 – 1.5 × IQR, where IQR is the interquartile range (Q3–Q1). The upper whisker is the highest point within Q3 + 1.5 × IQR. The diamonds are points that lie outside of the whiskers. Error bars in monthly median flows are the standard deviation across sites.
improved upon the single-task models for seven sites. When a multi-task scaling factor of 1 was used (as would be the default if no scaling factor was selected) the multi-task models performed significantly worse for 47 of the 53 sites (see Table 3).

3.5. Performance of Auxiliary Variable

Although we focused on how multi-task learning improves the main variable’s predictions, Figure 9 shows the results of the change in performance for the auxiliary variable in Experiment D (streamflow) in addition to the performance for the main variable (stream temperature). We see that as expected, the performance of the auxiliary variable was poorest when the scaling factor is zero. As the scaling factor increased, the flow predictions improved. However, when the site-specific best multi-task scaling factors based on temperature were used, the performance was sub-optimal for the streamflow predictions (median NSE of 0.72 compared to 0.75 if a scaling factor of 0.05 was used). Therefore, although there was a tradeoff between optimizing solely for streamflow performance and optimizing solely for stream temperature performance, a configuration for this model setup produced good (if not optimal) predictions for both variables. This tradeoff is a reality with all multi-objective optimization approaches (Efstratiadis & Koutsoyiannis, 2010).

4. Discussion

The goal of our experiments was to evaluate the performance of multi-task modeling for predicting streamflow and water temperature. We compared the accuracy of the multi-task approach (predicting streamflow and water temperature simultaneously with one model) to a single-task modeling approach (predicting streamflow or water temperature with their own independent models). Because streamflow and water temperature are related, we hypothesized that a multi-task model would be more accurate than a single-task model. The results of our experiments were nuanced. In some cases, the multi-task approach performed significantly better, in some there was no significant difference, and in some the multi-task model performed significantly worse. The improvements made by the multi-task approach seemed to depend on (a) seasonal factors, (b) which variable was being used as the primary variable, (c) the site being modeled, and (d) data availability. Together these affected the optimal value of the multi-task scaling factor. Given these nuances, we suggest that a multi-task modeling approach should be pursued with care. However logical and reasonable a multi-task approach may seem at the outset, it cannot not be assumed that the approach will be beneficial without exploration of the factors above.

4.1. Seasonality of Experiment a Results

There was an apparent seasonal pattern to the improvements made by the multi-task approach where, in general, the improvements were greater in the high flow months and lesser in the low flow months (Figure 8). The influence of snowmelt and base flow are two possible physical mechanisms that could cause differences in the streamflow-water temperature relationship in spring compared to summer and therefore cause this seasonal pattern. Snowmelt and base flow have seasonal patterns and can be strong influences on both streamflow and water temperature (Anderson, 2005; Constantz, 2008; Lisi et al., 2015). For a site that is substantially affected by snowmelt, a multi-task model might have parameters that are trained to predict a decrease in stream temperature for conditions in which an increase in streamflow is also predicted, as would happen during a snowmelt event. Those parameters could make performance worse, however, when increases in streamflow are not due to snowmelt but to rainfall as would happen in the summer months. Similarly, a multi-task model whose parameters capture the flow and temperature dynamics of high-flow months may be less well-suited to predict those dynamics in the summertime when the relative influence of base flow on streamflow and water temperature changes.
The tendency to favor the high-flow periods would not be desirable for certain use cases such as drought. This issue could be addressed in future work in several ways. As a first approach, the model could be trained to predict logged streamflow which would substantially lessen the influence of high flows on the loss function. A more tailored loss function could also be used such as log NSE. As another approach, the loss could be changed to an average of the NSEs of each season. That way the errors in the model during the low-flow summer seasons would be considered separate from, and would not be overshadowed by, the errors during the higher-flow spring season. This could, however, make a model that is mediocre in performance for each of the seasons instead of better in springtime and worse in summertime. As a more extensive solution, one could train season-specific models where each model was only trained using data from a certain season. This approach would produce a model for each season. When making predictions, the season-specific model outputs would be concatenated together. In this way, the model would be able to focus on season-specific dynamics between streamflow and water temperature. One draw-back of this approach, however, could be the significantly smaller data set that would result from dividing the training record by season. Another opportunity is to have the multi-task scaling factor dynamically adjusted for different time steps. For example, the multi-task scaling factor could be a function of the day of year.

4.2. Smaller Improvements in Predicting Water Temperature

The multi-task models had much smaller improvements on the single-task models when the main variable was stream temperature compared to when the main variable was streamflow. One reason for this difference may be that, in terms of NSE, water temperature is easier to predict than streamflow, so less improvement can be made in predicting water temperature. Both the seasonal trends in water temperature and the constrained range (between 0 and about 30 degrees C), make water temperature predictions easier in terms of annual NSE. In general, stream temperatures have a predictable sinusoidal pattern with highest temperatures in the summer and lowest temperatures in the winters.

4.3. Selecting an Appropriate Multi-Task Scaling Factor

Our results anecdotally suggest that a larger multi-task scaling factor would be best if the auxiliary variable is easier to predict. In terms of NSE, predicting stream temperature is easier than predicting streamflow; therefore, when streamflow was the main variable, the best multi-task scaling factors were larger (most commonly 80 in Experiment A). When predicting stream temperature, the best multi-task scaling factors were smaller (most commonly 0.01 in Experiment D). One way to get a general sense of the relative predictability of the two variables could be the ratio of the variables’ standard deviations or ranges. For example, the ratio of standard deviations between streamflow and stream temperature averaged across all sites was 3.3 and the ratio of the scaled ranges was 6.7. Because both values are greater than one, this indicates that the unscaled multi-task loss will favor streamflow. If streamflow is the primary variable, a multi-task scaling factor greater than one would likely be needed for stream temperature errors to noticeably impact the overall gradient. Although more tuning would be needed to find the optimal factor, this simple comparison could provide a starting point for the search.

Our results also suggest that a larger multi-task scaling factor should be used if the auxiliary variable is represented in only a small percentage of training samples. In our data some training samples (365-day sequences) were without any temperature observations. These sequences produced gradients in model training that were only informed by the streamflow data; these were effectively single-task gradients. In Experiment A, we saw, in general, an inverse relationship between the number of training samples with water temperature data and the best multi-task scaling factor (Figure 10). It seemed that when there were fewer training samples with auxiliary data, a larger multi-task scaling factor was needed to influence the overall gradient used to update the model weights. The larger multi-task scaling factor counter-balanced the influence from the larger number of gradients from the samples with only data from the main variable. This explanation also holds for Experiment C in which
the percentage of training samples with auxiliary data was higher (either 50% or 100%) and the best multi-task scaling factors were typically smaller (most commonly 10).

4.4. Multi-Task Learning Versus Using the Auxiliary Information as Input

Multi-task learning is but one approach for using information contained in an auxiliary variable to improve main variable performance. Another approach is to use the auxiliary variable as an input to the model. For example, when Rahmani et al. (2020) used streamflow observations as an input variable to predict water temperature, water temperature predictions improved. When the auxiliary variable is used as an input to a model, the model parameters are trained to leverage the information therein during the model training phase, an automated process. This approach uses the auxiliary information without the time cost of tuning a separate hyperparameter needed in multi-task learning (i.e., time needed to tune the multi-task scaling factor). Although the approach of using the auxiliary data as an input does not require hyperparameter tuning, it does require a value for the auxiliary variable for each time step at each location modeled in the training, validation, and testing periods.

Compared to using the auxiliary data as input, multi-task modeling may be a more efficient use of available auxiliary data when locations only have sparse observations. For example, in our data set, only 4 out of 101 sites had more than 80% coverage for water temperature observations in the training, validation, and test time periods. A forecasting scenario is another case where auxiliary observations are not available. Although in both sparse data and forecast scenarios a secondary model could be used to predict future or missing data, a multi-task approach has the advantage of being able to learn from the available auxiliary data without relying on the existence or accuracy of a secondary model.

Convenience aside, multi-task learning appears to produce more accurate predictions than using auxiliary data as input in some sparse data conditions. We performed a preliminary investigation comparing streamflow prediction results between the best-configured multi-task models and models that used stream temperature data as input. Our results showed that when a substantial number of water temperature observations were missing, the best-configured multi-task models were more accurate overall (see Text S2, Table S2, and Figure S1 in Supporting Information S1).

4.5. Limitations and Potential Future Work

As with most data-driven approaches, the models we used were limited by the training data available. Likewise, the effectiveness of the approach rests on the assumption that the training conditions are a good predictor of the prediction conditions. Evidence indicates that this assumption of stationarity is becoming less valid given the observed and expected climate change (McCabe & Wolock, 2002). Nevertheless, in our results the models performed well even when trained on data from certain decades (e.g., 1990–2014 for Experiment A) and evaluated on data from other decades (e.g., 1980–1989). Despite any changes to the climate that may have already occurred and that may occur in the future, the underlying physical, chemical, and biological mechanisms that drive the processes that are modeled will not change. This highlights the importance of the infusion of process understanding into data-driven models rather than relying wholly on data that cannot adequately capture the full process being modeled.

There may be more efficient methods of deciding on an appropriate multi-task scaling factor compared to brute-force hyperparameter tuning. For example, Kraft et al. (2020) used the uncertainty in each variable to scale the influence of the variable in the loss function. One reason we did not explore this approach is that it does not allow for the explicit designation of a main variable of interest, whose performance should not be sacrificed at the expense of the other variables being predicted. We were instead interested in an approach that would only benefit the prediction of the main variable. Some more automated approaches for achieving this objective could also be explored, such as those by Du et al. (2020) and Liu et al. (2019).

It can reasonably be assumed that the appropriateness of a multi-task scaling factor is largely dependent on the process relationship between the two variables of interest. Because the statistical relationship between two variables can be influenced (if not determined) by physical characteristics and physical, chemical, and/or biological processes, it could be possible to leverage our process understanding to estimate an appropriate multi-task scaling factor for a given site and combination of variables. Although we did not discover a predictable pattern
for estimating an appropriate multi-task scaling factor value based on catchment characteristics and data availability, that pattern may exist for streamflow and stream temperature with another set of catchment characteristics or different method of relating them to the multi-task scaling factor. If multi-task learning were being explored for other variables, there may be more easily discoverable patterns that would narrow the search for a beneficial multi-task scaling factor.

This paper examines multi-task learning for just two hydrologic variables; however, the approach could be applied to predicting other variables. We hypothesize that for multi-task learning to have benefit, the variables would have to have some common drivers (stemming from a physical/chemical/biological relationship). There would also need to be sufficient data for both variables from which the multi-task model could learn. For example, other parts of the water balance equation that are observed such as soil moisture could be used to train a multi-task model with a main objective of predicting streamflow. Another example is the modeling of dissolved oxygen and water temperature together.

Multi-task learning could be used to predict any number of observed variables. For example, a model could be trained to predict streamflow, water temperature, and soil moisture. As we discussed above, the selection of the multi-task scaling factor would be an important part of a successful multi-task model. This selection would be even more important (and likely more difficult) if multiple variables were used as auxiliary variables because each auxiliary variable would need its own multi-task scaling factor. This would also open the possibility of assimilating observations of multiple observed variables in a forecast application.

Although the variables for multi-task learning must have enough data for training, that data could be simulated data. For example, it may be fruitful to train a multi-task model on all water balance outputs produced from a process-based hydrologic model such as the Soil and Water Assessment Tool (SWAT) model (e.g., evapotranspiration, runoff, infiltration; USDA, 2018). Then the model could be fine-tuned to just the data for which observations are available. The model would presumably learn the dynamics of the hydrologic cycle more completely from the process model output and could then better predict the variable of interest when given unseen inputs.

5. Conclusions

Our study examined the value of multi-task DL for leveraging information from related hydrologic variables to improve predictions. In the multi-task approach, one DL model was trained to predict both streamflow and water temperature. The effectiveness of the approach was tested using four experiments: (a) predicting streamflow as the main variable with 25 years of training data using site-specific models, (b) predicting streamflow as the main variable with 2 years of training data using one multi-site model, (c) predicting streamflow as the main variable with 25 years of training data using one multi-site model, (d) and predicting water temperature as the main variable using site-specific models. For the experiments we used data from sites throughout the conterminous United States where coincidental streamflow and water temperature observations were available.

When using single-site models, the best site-specific, multi-task configuration significantly improved streamflow predictions for 56 of 101 the monitoring sites with the full 25 years of training data in terms of NSE. At 4 sites, the best multi-task models were significantly worse than the single-task models and at the remaining 43 sites there was no significant difference. The improvements made by the multi-task approach when using the multi-site model were much smaller: with the site-specific best configuration, 89 sites had no significant difference and 11 were significantly improved. With only 2 years of training data, the best site-specific multi-task approach significantly improved streamflow predictions at 46 out of 98 sites. When predicting water temperature, the multi-task approach improved predictions at only seven sites. The main reason that the multi-task approach had a smaller benefit for stream temperature predictions was likely that the stream temperature predictions were already very good in terms of NSE (median NSE > 0.95).

One of the most important factors in the effectiveness (or lack of effectiveness) of the approach for a given site was the multi-task scaling factor. This model hyperparameter scaled the influence of the errors in the secondary (or auxiliary) variable in the overall error used to update the model parameters. When streamflow was the primary (or main) variable, the scaling factor had to be quite high (most commonly the best factor was 80) for the single-site models. In contrast, when the main variable was stream temperature the scaling factor had to be very low (most commonly the best factor was 0.01). A naïve factor of 1 led to models that performed much worse
overall for predicting stream temperature than the single-task models. The main reason for the difference in the scaling factors could be attributed to the difference in variability between the two variables. Seasonal trends in the effectiveness of the multi-task model as well as examining the characteristics of the catchments that benefited most and least from the approach indicate that base flow is an important factor in whether this approach is successful or not for improving streamflow predictions.

Streamflow and water temperature are but two among many variables that are part of the physical/chemical/biological processes that are ecologically and societally consequential. Our findings indicate that multi-task learning, when configured properly, can be an effective way to leverage knowledge of the relationship between hydrologic variables in a DL model for improved predictions. In our findings the multi-task approach did not improve predictions in all cases tried and where there were benefits, proper configuration was needed to realize those benefits.

Conflict of Interest
The authors declare no conflicts of interest relevant to this study.

Data Availability Statement
The data and code used to obtain the results of this paper can be obtained from the corresponding USGS Science Base item (Sadler et al., 2021). Thanks to Janet Barclay of the U.S. Geological Survey for the time spent in providing feedback that improved this data and code release. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Acknowledgments
The authors thank Rasha Atshan who provided advice for the use of the Welch’s t-test. The results figures were generated using the Matplotlib (Hunter, 2007), Seaborn (Waskom & The Seaborn Development Team, 2020), and GeoPandas (geopandas.org) Python libraries. Our methods also relied heavily on the xarray (Hoyer & Hamman, 2017), Dask (Dask Development Team, 2016), Pandas (Pandas Development Team, 2020), and NumPy (Harris et al., 2020) Python libraries. The authors acknowledge the Pangeo team (pangeo.io) who provided cloud computational resources used in prototyping the DL models. The USGS Advanced Research Computing group supplied the computational resources for running the experiments in the paper. Special thanks to John Hammond, Melissa Lombard, and Janet Carter of the U.S. Geological Survey for their detailed reading and helpful comments in the revision process of the manuscript. Thanks also to Simon Topp for his comments and review of the Supporting Information S1. Vipin Kumar was supported by the grant #1934721 from the U.S. National Science Foundation. Xiaowei Jia is supported by the USGS awards G21AC10207 and G21AC10564.

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/10.5194/hess-21-5293-2017
Alemu, E. T., Palmer, R. N., Polebitksi, A., & Meaker, B. (2011). Decision support system for optimizing reservoir operations using ensemble streamflow predictions. *Journal of Water Resources Planning and Management*, 137(1), 72–82. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000088
Anderson, M. P. (2005). Heat as a ground water tracer. *Ground Water*, 43(6), 951–968. https://doi.org/10.1111/j.1745-6584.2005.00052.x
Bai, Y., Chen, Z., Xie, J., & Li, C. (2016). Daily reservoir inflow forecasting using multiscale deep feature learning with hybrid models. *Journal of Hydrology*, 532, 193–206. https://doi.org/10.1016/j.jhydrol.2015.11.011
Bain, M. B., Finn, J. T., & Booke, H. E. (1988). Streamflow regulation and fish community structure. *Ecology*, 69(2), 382–392. https://doi.org/10.2307/1940436
Barnett, P. R. (1972). Effects of warm water effluents from power stations on marine life. *Proceedings of the Royal Society of London – Series B: Biological Sciences*, 180(61), 497–509. https://doi.org/10.1098/rspb.1972.0034
Besaw, L. E., Rizzo, D. M., Bierman, P. R., & Hackett, W. R. (2010). Advances in ungauged streamflow prediction using artificial neural networks. *Journal of Hydrology*, 386(1–4), 27–37. https://doi.org/10.1016/J.JHYDROL.2010.02.037
Bowes, B. D., Sadler, J. M., Morsy, M. M., Behl, M., & Goodall, J. J., (2019). Forecasting groundwater table in a flood prone coastal city with long short-term memory and recurrent neural networks. *Water (Switzerland)*, 1(5), 1098. https://doi.org/10.3390/w11051098
Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
Campolo, M., Andressi, P., & Soldati, A. (1999). River flood forecasting with a neural network model. *Water Resources Research*, 35(4), 1191–1197. https://doi.org/10.1029/1998WR900086
Carlisle, D. M., Wolock, D. M., & Meador, M. R. (2011). Alteration of streamflow magnitudes and potential ecological consequences: A multiregional assessment. *Frontiers in Ecology and the Environment*, 9(5), 264–270. https://doi.org/10.1890/100053
Chen, D., Mak, B., Leung, C., & Sivadas, S. (2014). Joint acoustic modeling of triphones and trigram phones by multi-task learning deep neural networks for low-resource speech recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5592–5596). https://doi.org/10.1109/ICASSP.2014.6854673
Chen, P. A., Chang, L. C., & Chang, F. J. (2013). Reinforced recurrent neural networks for multi-step-ahead flood forecasts. *Journal of Hydrology*, 497, 71–79. https://doi.org/10.1016/J.JHYDROL.2013.05.038
Constantz, J. (2008). Heat as a tracer to determine streambed water exchanges. *Water Resources Research*, 46(4). https://doi.org/10.1029/2008WR006996
Dask Development Team. (2016). Dask: Library for dynamic task scheduling. Retrieved from https://dask.org
Du, Y., Czarniecki, W. M., Jayakumar, S. M., Farajtabar, M., Pascani, R., & Lakshminarayanan, B. (2020). Adapting auxiliary losses using gradient similarity. arXiv. https://doi.org/10.48550/arXiv.1812.02224
Efstratidis, A., & Koutsoyiannis, D. (2010). Une décennie d’approches de calage multi-objectifs en modélisation hydrologique: Une revue. *Hydrological Sciences Journal*. https://doi.org/10.1002/hysj.26660903526292
Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*. Retrieved from https://github.com/rbgirshick/
Hamlet, A. F., Huppert, D., & Lettenmaier, D. P. (2002). Economic value of long-lead streamflow forecasts for Columbia river hydropower. *Journal of Water Resources Planning and Management*, 128(2), 91–101. https://doi.org/10.1061/(ASCE)0733-9496(2002)128:2(91)
Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., et al. (2020). Array programming with NumPy. Nature, 585(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

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

Hoyer, S., & Janmann, J. (2017). xarray: {N-D} labeled arrays and datasets in {Python}. Journal of Open Research Software, 5(1). https://doi.org/10.5334/jors.148

Hsu, K.-I., Gupta, H. V., & Sorooshian, S. (1995). Artificial neural network modeling of the rainfall-runoff process. Water Resources Research, 31(10), 2517–2530. https://doi.org/10.1029/95WR01955

Hu, C., Wu, Q., Li, H., Jian, S., Li, N., & Lou, Z. (2018). Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. Water, 10(11), 1543. https://doi.org/10.3390/w10111543

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Kingma, D. P., & Ba, J. L. (2014). Adam: A method for stochastic optimization. In ICLR 2015 – Conference Track Proceedings.

Kistler, J., & Rahmann, S. (2012). Snakemake—a scalable bioinformatics workflow engine. Bioinformatics, 28(19), 2520–2522. https://doi.org/10.1093/bioinformatics/bts480

Kraft, B., Jung, M., Körner, M., & Reichstein, M. (2020). Hybrid modeling: Fusion of a deep learning approach and a physics-based model for global hydrological modeling. BPRS – International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLIII-B2-2, 1537–1544. https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-1537-2020

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall-runoff modelling using Long Short-Term Memory (LSTM) networks. Hydrology and Earth System Sciences, 22(11), 6005–6022. https://doi.org/10.5194/hess-22-6005-2018

Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A. K., Hochreiter, S., & Nearing, G. S. (2019). Toward improved predictions in ungauged basins: Exploiting the power of machine learning. Water Resources Research, 55(12), 11344–11354. https://doi.org/10.1029/2019WR026065

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), 5809–5810. https://doi.org/10.5194/hess-23-5809-2019

Liu, S., Davison, A. J., & Johns, E. (2019). Self-supervised Generalisation with Meta Auxiliary Learning. arXiv. https://doi.org/10.48550/arXiv.1901.08933

Martin, A., Ashish, A., Paul, B., Eugenie, B., Zhifeng, C., Craig, C., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Retrieved from https://www.tensorflow.org

McCabe, G. J., & Wolock, D. M. (2002). A step increase in streamflow in the conterminous United States. Geophysical Research Letters, 29(24), 38-1. https://doi.org/10.1029/2002GL015999

Nash, J. E., & Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I – A discussion of principles. Journal of Hydrology, 10(3), 282–290. https://doi.org/10.1016/0022-1694(70)90255-6

Newman, A. J., Clark, M. P., Sampson, K., Wood, A., Hay, L. E., Bock, A., et al. (2015). Development of a large-sample watershed-scale hydro-meteorological data set for the contiguous USA: Data set characteristics and assessment of regional variability in hydrologic model performance. Hydrology and Earth System Sciences, 19(1), 209–223. https://doi.org/10.5194/hess-19-209-2015

Pandas Development Team. (2020). pandas-dev/pandas: Pandas. Zenodo. https://doi.org/10.5281/zenodo.3509134

Pappenberger, F., Cloke, H. L., Parker, D. J., Wetterhall, F., Richardson, D. S., & Thielen, J. (2015). The monetary benefit of early flood warnings in Europe. Environmental Science & Policy, 51, 278–291. https://doi.org/10.1016/j.envsci.2015.04.016

Rahmani, F., Lawson, K., Ouyang, W., Appling, A., Oliver, S., & Shen, C. (2020). Exploring the exceptional performance of a deep learning stream temperature model and the value of streamflow data. Environmental Research Letters. https://doi.org/10.1088/1748-9326/abac50

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/10.1029/2019WR024922

Robertson, D. E., Shrestha, D. L., & Wang, Q. J. (2013). Post-processing rainfall forecasts from numerical weather prediction models for short-term streamflow forecasting. Hydrology and Earth System Sciences, 17(9), 3587–3603. https://doi.org/10.5194/hess-17-3587-2013

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. ArXiv Preprint ArXiv:1706.05098.

Sadler, J. M., Appling, A. P., Read, J. S., Oliver, S. K., Jia, X., Zwart, J. A., & Kumar, V. (2021). Data release: Multi-task deep learning for water temperature and streamflow prediction. https://doi.org/10.5066/P9K0TQSL

Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components. Biometric Bulletin, 2(6), 110. https://doi.org/10.2307/3002019

Seltzer, M. L., & Droppo, J. (2013). Multi-task learning in deep neural networks for improved phoneme recognition. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 6965–6969). https://doi.org/10.1109/ICASSP.2013.6639012

Shen, C. (2018). A transdisciplinary review of deep learning research and its relevance for water resources scientists. In Water Resources Research. Blackwell Publishing Ltd. https://doi.org/10.1029/2018WR022643

Shen, C., Laloy, E., Elshorbagy, A., Albert, A., Bales, J., Chang, F. J., et al. (2018). HESS opinions: Incubating deep-learning-powered hydrologic science advances as a community. Hydrology and Earth System Sciences, 22(11), 5639–5656. https://doi.org/10.5194/hess-22-5639-2018

Soleiman, S., Bozorg-Haddad, O., Saadatpour, M., & Loaiciga, H. A. (2019). Simulating thermal stratification and modeling outlet water temperature in reservoirs with a data-mining method. Journal of Water Supply: Research & Technology – Aqua, 68(1), 7–19. https://doi.org/10.2166/aqua.2018.036

Stammler, K. L., Yates, A. G., & Bailey, R. C. (2013). Barred streams: Uncovering a potential threat to aquatic ecosystems. Landscape and Urban Planning, 114, 37–41. https://doi.org/10.1016/J.LANDURBPLAN.2013.02.008

USDA ARS Grassland Soil and Water Research Laboratory; Texas A&M AgriLife Research. (2018). SWAT – Soil and Water Assessment Tool. USDA Agricultural Research Service; Texas A&M AgriLife Research. Retrieved from https://data.nal.usda.gov/dataset/swat-soil-and-water-assessment-tool

Van Vliet, M. T. H., Ludwig, F., Zwolsman, J. J. G., Weeden, G. P., & Kabat, P. (2011). Global river temperatures and sensitivity to atmospheric warming and changes in river flow. Water Resources Research, 47(2). https://doi.org/10.1029/2010WR009198

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods, 17(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Waskom, M., & The Seaborn Development Team. (2020). Seaborn. Zenodo. https://doi.org/10.5281/zenodo.592845
Zahura, F. T., Goodall, J. L., Sadler, J. M., Shen, Y., Morsy, M. M., & Behl, M. (2020). Training machine learning surrogate models from a high-fidelity physics-based model: Application for real-time street-scale flood prediction in an urban coastal community. *Water Resources Research, 56*(10). https://doi.org/10.1029/2019WR027038

Zeng, L. Q., Cao, Z. D., Fu, S. J., Peng, J. L., & Wang, Y. X. (2009). Effect of temperature on swimming performance in juvenile southern catfish (*Silurus meridionalis*). *Comparative Biochemistry and Physiology – A Molecular and Integrative Physiology, 153*(2), 125–130. https://doi.org/10.1016/j.cbpa.2009.01.013

### References From the Supporting Information

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/10.5194/hess-21-5293-2017

Ladson, A. R., Brown, R., Neal, B., & Nathan, R. (2013). A standard approach to baseflow separation using the Lyne and Hollick filter. *Australian Journal of Water Resources, 17*(1), 25–34. https://doi.org/10.7158/13241583.2013.11465417

Newman, A. J., Clark, M. P., Sampson, K., Wood, A., Hay, L. E., Bock, A., et al. (2015). Development of a large-sample watershed-scale hydro-meteorological data set for the contiguous USA: Data set characteristics and assessment of regional variability in hydrologic model performance. *Hydrology and Earth System Sciences, 19*(1), 209–223. https://doi.org/10.5194/hess-19-209-2015