Improving large-basin river routing using a differentiable Muskingum-Cunge model and physics-informed machine learning

Tadd Bindas[1], Wen-Ping Tsai[2], Jiangtao Liu[1], Farshid Rahmani[1], Dapeng Feng[1], Yuchen Bian[3], Kathryn Lawson[1], Chaopeng Shen*[,1]

[1] Civil and Environmental Engineering, The Pennsylvania State University, PA
[2] Hydraulic and Ocean Engineering, National Cheng Kung University, Tainan City
[3] Amazon Search, Palo Alto, CA

* Corresponding author: Chaopeng Shen, cshen@engr.psu.edu

Abstract

Recently, rainfall-runoff simulations in small headwater basins have been improved by methodological advances such as deep neural networks (NNs) and hybrid physics-NN models --- particularly, a genre called differentiable modeling that intermingles NNs with physics to learn relationships between variables. However, hydrologic routing, necessary for simulating floods in stem rivers downstream of large heterogeneous basins, had not yet benefited from these advances and it was unclear if the routing process can be improved via coupled NNs. We present a novel differentiable routing model that mimics the classical Muskingum-Cunge routing model over a river network but embeds an NN to infer parameterizations for Manning's roughness (*n*) and channel geometries from raw reach-scale attributes like catchment areas and sinuosity. The NN was trained solely on downstream hydrographs. Synthetic experiments show that while the channel geometry parameter was unidentifiable, *n* can be identified with moderate precision. With real-world data, the trained differentiable routing model produced more accurate long-term routing results for both the training gage and untrained inner gages for larger subbasins (>2,000 km^2) than either a machine learning model assuming homogeneity, or simply using the sum of runoff from subbasins. The *n* parameterization trained on short periods gave high performance in other periods, despite significant errors in runoff inputs. The learned *n* pattern was consistent with literature expectations, demonstrating the framework's potential for knowledge discovery, but the absolute values can vary depending on training periods. The trained *n* parameterization can be coupled with traditional models to improve national-scale flood simulations.

Main points:

1. A novel differentiable routing model can learn effective river routing parameterization, recovering channel roughness in synthetic runs.
2. With short periods of real training data, we can improve streamflow in large rivers compared to models not considering routing.
3. For basins >2,000 km^2, our framework outperformed deep learning models that assume homogeneity, despite bias in the runoff forcings.

1. Introduction

Riverine floods pose a major risk to human safety and infrastructure (Douben, 2006; François et al., 2019; International Panel on Climate Change (IPCC), 2012; Koks & Thissen, 2016) and are linked to stream channel characteristics. Riverine floods along large stem rivers occur when the peak flow rate exceeds the stem river conveyance capacity. The timing of flood convergence and peak flood rates are influenced by the channel's geometries and flow resistance properties (Candela et al., 2005; Kalyanapu et al., 2009). In recent years, we have witnessed many deadly riverine floods, e.g., in the Mississippi River, USA (Rice, 2019) and India (France-Presse, 2022), with such disasters expected to rise significantly based on future climate projections (Dottori et al., 2018; Prein et al., 2017; Winsemius et al., 2016). The ability to better account for flood convergence and streamflow processes is urgently needed to help us better inform society of stem river flood magnitudes and timing.

In hydrologic modeling, routing describes how the stream network conveys runoff downstream while accounting for mass balances and the speed of flood wave propagation (Mays, 2010). Most routing models are based on the principle of continuity (or mass conservation) but they differ in how the momentum equation or flow velocity is calculated. For example, the widely-applied Muskingum-Cunge (MC) (Cunge, 1969) routing method is a center-in-space center-in-time finite difference solution to the continuity equation, assuming a prismatic flood wave as the constitutive relationship to simplify the momentum equation. In some other cases, the momentum equation is solved in conjunction with the continuity equation (Ji et al., 2019) with a range of simplifying assumptions, e.g., ignoring inertia (Shen & Phanikumar, 2010), ignoring both inertia and pressure gradient (only slope remaining) (Mizukami et al., 2016), or including additional formulations to handle effects of scale, e.g., Li et al. (2013). In each case, these models have parameters that need to be determined from lookup tables or calibration, e.g., roughness parameters that serve as resistance to flow.

Although routing parameters often rank among the important ones for discharge simulation (Khorashadi Zadeh et al., 2017; L. Liu et al., 2022), they been difficult to parameterize at large scales, especially in a way to both sensibly represent basin-internal spatial heterogeneity and adapt to discharge data. Using traditional roughness values tabulated for various land covers (Arcement & Schneider, 1989) requires in-situ scouting, e.g., to determine if channels have pools, weeds, grass, etc., which is currently impractical for large-scale applications. Many calibration exercises (Khorashadi Zadeh et al., 2017; L. Liu et al., 2022;

Mizukami et al., 2016) have used only one set of parameters for an entire basin, neglecting fine-scale spatial heterogeneity in river-reach characteristics. Some studies have employed Manning's roughness, *n* (a coefficient representing a channel's resistance to flow), as a linear function of river depth or other characteristics (Getirana et al., 2012; H.-Y. Li et al., 2022), but it is unclear if these relationships accurately represent the available data.

While the accuracy of basin rainfall-runoff models has improved substantially in recent years with machine learning (ML) (Adnan et al., 2021; Feng et al., 2020; Kratzert et al., 2019; Sun et al., 2022; Xiang et al., 2020), these methods have not been applied to routing modules in order to benefit the simulation of stem river floods. Neural networks (NNs) like long short-term memory (LSTM), GraphWaveNet (Sun et al., 2021), or convolutional networks (Duan et al., 2020) have demonstrated their prowess in learning hydrologic dynamics from big data. They are applicable not only to streamflow hydrology but also to variables across the entire hydrologic cycle (Shen, Chen, et al., 2021; Shen & Lawson, 2021) such as soil moisture (Fang et al., 2017, 2019; J. Liu et al., 2022; O & Orth, 2021), groundwater (Wunsch et al., 2022), snow (Meyal et al., 2020), longwave radiation (Zhu et al., 2021), and water quality parameters like water temperature, dissolved oxygen and nitrogen (He et al., 2022; Hrnjica et al., 2021; Lin et al., 2022; Rahmani, Lawson, et al., 2021; Saha et al., 2023; Zhi et al., 2021). However, these approaches are mostly suitable for relatively homogeneous headwater basins; spatial heterogeneities in forcings and basin characteristics are generally not well represented in these approaches. In our previous studies we observed that large basins often turned out to have poorer performance for LSTM models. The routing module is the key component that allows us to consider how runoff from heterogeneous subbasins converge and contribute to the stem river floods, and could be extended to support reactive transport modeling in the river network.

A recent development in integrating ML with physical understanding is the use of differentiable, physics-informed machine learning models, which can approach the performance of purely data-driven ML models but also provide interpretable fluxes and states (Feng, Liu, et al., 2022). "Differentiable” models can rapidly and accurately compute the gradients of the model outputs with respect to any input, enabling the combined training of NNs to approximate complex or unknown functions from big data while keeping physical priors. Such models can be simply supported by automatic differentiation (AD), which tracks each elementary operation of tensors through the use of a computational graph, then uses derivative rules to compute the gradient of each tensor operation (Baydin et al., 2018). This enables hybrid frameworks to learn and incorporate complex and potentially unknown functions from big data while retaining physical formulations. By connecting deep networks to reimplemented process-based models (or their NN surrogates), Tsai et al. (2021) developed a NN-based parameterization pipeline that infers physical parameters for process-based models. Differentiable models can also extrapolate better in space and time than purely data-driven deep networks (Feng, Beck, et al., 2022). These methods are also applicable to estimating parameters in ecosystem modeling (Aboelyazeed et al., 2022), and allow us to flexibly discover variable relationships within the model based on big data, enabling improved transparency compared to standard deep learning models.

Nevertheless, it was unclear if differentiable modeling could effectively learn relationships in a highly complex river network, which convolves and integrates processes over large scales and thus render small-scale processes unidentifiable. The river network forms a hierarchical graph, which is not unlike the graph networks for applications like social recommendations (Fan et al., 2019), but with a predefined spatial topology (due to a fixed river network) and a converging cascade. A complex river graph can have many nodes, which, when coupled with many time steps, could potentially lead to a training issue known as the vanishing gradient (Hochreiter, 1998), where the gradients with respect to the parameters are vanishingly small and the system becomes very difficult to train. Moreover, runoff data (required as an input for routing) are generally not available seamlessly for all subbasins and must be estimated by models, but models for runoff could incur substantial errors. It was unclear if the routing parameters could be learned, given such errors. It was further unclear if downstream discharge data alone has enough information to enable learning of reach-scale relationships. In other words, a reach-scale relationship may or may not be identifiable using downstream observations which integrate the signals from the entire catchment area.

In this work, we developed a novel differentiable modeling framework to perform routing and to learn a “parameterization scheme” (a systematic way of inferring parameters from more rudimentary information) for routing flows on the river network. Such a physically-based routing method has never been combined with NNs before. A NN-based parameterization scheme for Manning’s n and river bathymetry shape (q) is integrated with MC routing and is applied throughout the river network to provide improved understanding of both the model and the modeled system. We designed synthetic and real data experiments to answer the following research questions:

1. *Given substantial errors with estimated runoff as inputs to the routing module, can we learn effective routing parameterization schemes that can produce reliable results for long-term simulations in large river networks?*
2. *Does the learned parameterization perform well for both trained and untrained internal gages and how does the performance vary as a function of basin area?*
3. *Do short periods of downstream discharge contain sufficient information to train a reliable parameterization scheme or to identify the parameterization for channel roughness and hydraulic geometries?*

2. Data and Methods

2.1 Overview

As an overview, we describe a differentiable model that routes runoff through a river network (or “graph” in the ML terminology) similar to the traditional Muskingum-Cunge (MC) method. But unlike the traditional MC, our differentiable model is able to incorporate and train neural networks to provide reach-scale parameterization. This new routing model can be perceived as a physics-informed graph neural network (GNN) from an ML perspective. The nodes of the graph are spaced ~2000 m apart to ensure stability. We trained an embedded a Multilayer Perceptron (MLP) NN to generate spatially-distributed river parameters for each reach (or “edge” in the GNN terminology) in the river network (Figure 1b). The loss function (the model’s goal is to minimize the output of this) was calculated at the furthest downstream node of the graph. To disentangle rainfall-runoff (required information for routing) from the routing processes, lateral inflow of combined overland and groundwater flow was obtained from a pre-trained LSTM streamflow prediction model (reported in previous work). The runoff values were then disaggregated to hourly time steps via interpolation and routed throughout the river network using the proposed differentiable routing model (Figure 1a). We provide the details in the subsections that follow.

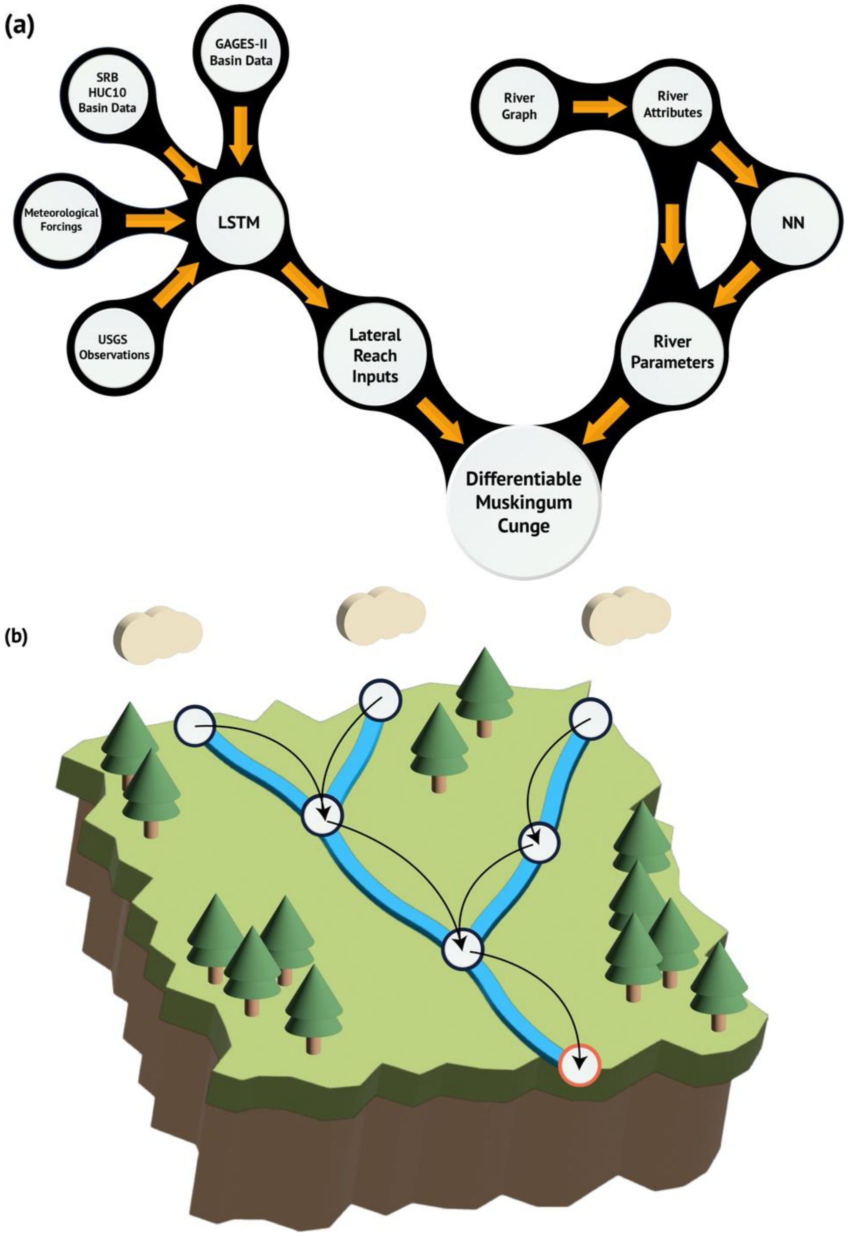

Figure 1: (a) An abstract overview of how inputs move through our workflow to eventually be run through the differentiable MC function. MC utilizes lateral flow inputs based on LSTM predictions, NN predicted river parameters n and q, and other river attributes to generate predictions. (b) An illustration of how we traverse the graph (dark blue circles) using MC to make a discharge prediction for the final node (orange circle).

2.2 The River Graph

We constructed a river network (or graph) for the Juniata River Basin (JRB) in the northeastern United States (Figure 2), by processing the United States Geological Survey's (USGS's) National Hydrography Dataset (NHDplus v2) (HorizonSystems, 2016; Moore & Dewald, 2016) which provide topology and some attributes of the river reaches such as upstream catchment area. We ensured stability of the MC scheme by discretizing the river network into approximately 2-km reaches, resulting in 544 junction points (or nodes) and 582 river reaches (or edges). These reaches are where the physical parameters like Manning's roughness and channel shape coefficients are defined. To reduce computational demand, we selected a subset of NHDplus v2 river reaches based on a stream density threshold (total stream length/watershed area), choosing rivers with the longest length until a stream density of 0.2 km/km^2 was reached. We then calculated slope and sinuosity for the reaches by overlaying NHDplus v2 with 10-m resolution digital elevation data (USGS ScienceBase-Catalog, 2022). Prior work describes the bulk of the extraction procedure that prepares input data for a physically-based surface-subsurface processes model (Ji et al., 2019; Shen et al., 2013, 2014, 2016; Shen & Phanikumar, 2010).

The hydrograph at the furthest downstream JRB gage, USGS gage 01563500 (node 4809 in our graph) on the Juniata River at Mapleton Depot, PA, was chosen as the training target (Figure 2a). This gage has a catchment area of 5,212 km^2 contributed from the 582 simulated reaches upstream. Seven USGS gages are located upstream of this node which enables further validation of the simulations.

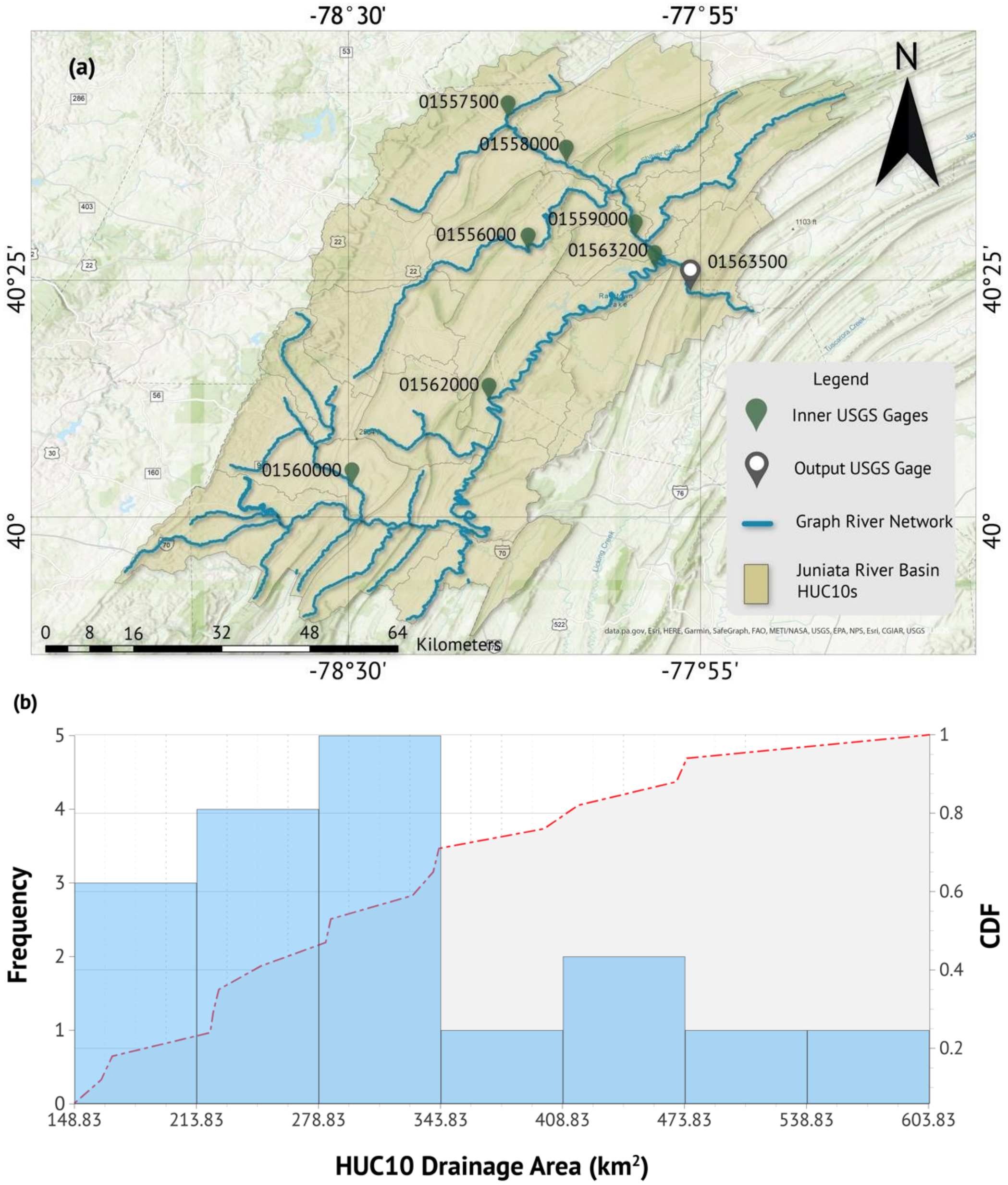

Figure 2: (a) A map of the Juniata River Basin's (JRB's) river network and HUC10 watersheds. Each eight-digit number corresponds to a USGS gage. (b) A histogram showing the distribution of HUC10 watersheds in the JRB. The x-axis shows the distribution of the HUC10 watershed area in square kilometers. The left y-axis shows the number of HUC10s that fall within the area ranges (corresponding with the blue bars), and the right y-axis shows a cumulative density function (CDF) distribution of the areas, corresponding with the red dashed line.

2.3 Implementing River Routing with Muskingum-Cunge

2.3.1 Muskingum-Cunge

The Muskingum-Cunge (MC) method is a widely-used flood routing technique that combines the Muskingum storage routing concept with the continuity and momentum equation for a river reach (Cunge, 1969), solved using a center-in-space, center-in-time finite difference scheme for each reach, at time steps *t* and *t+1:*

$$Q_{t+1} = c_1 I_{t+1} + c_2 I_t + c_3 Q_t + c_4 Q' \tag{1}$$

$$c_1 = \frac{\Delta t - 2KX}{2K(1-X) + \Delta t} \tag{2}$$

$$c_2 = \frac{\Delta t + 2KX}{2K(1-X) + \Delta t} \tag{3}$$

$$c_3 = \frac{2K(1-X) - \mathit{\Delta} t}{2K(1-X) + \mathit{\Delta} t} \tag{4}$$

$$c_4 = \frac{2\mathit{\Delta} t}{2K(1-X) + \mathit{\Delta} t} \tag{5}$$

Where I_t and Q_t are the inflow and outflow of the reach at time step t, respectively, and I_{t+1} and Q_{t+1} are the inflow and outflow at the next time step, *t+1*. *K* represents travel time based on reach length and wave celerity, *X* is a dimensionless inflow/outflow weighing parameter, and *Q'* represents lateral inflow of the incremental catchment area of the reach, and can also include tributary inflows. We adopted the simple linear form of the Muskingum equation: X is constant and K= $\mathit{\Delta} x/v$ where $\mathit{\Delta} x$ is length of the reach and v is the discharge velocity (m/s) of the current time step. More complex nonlinear forms of the MC equation could be tested in the future (Mays, 2019). To simulate a river network, we divide the network into a series of reaches to route the flow of water from upstream to downstream. The outflow from a reach is the inflow of the next downstream reach.

2.3.2 MC parameter values and variable channel dimensions

To implement MC, we chose an hourly time step (*Δt*) and a weighing coefficient (*X)* of 0.3, which was based on regional expectations, for Equations 2-5. Since discharge velocity *v* and stream top width *w* vary over time, they need to be updated in each time step with respect to discharge *Q*, which was done here with the help of a constitutive relationship used to close the equations. For this, because at-a-site hydraulic geometries (Gleason, 2015; Leopold & Maddock, 1953) leads to a power-law relation between top width (*w* [m]) and depth (*d* [m]), we can assume such a relationship:

$$w = pd^{q} \tag{6}$$

where *p* [m] and *q* [-] are linear and exponential parameters, respectively, that are potentially spatially heterogeneous and represent the shape of the channel's cross-sectional area. For a rectangular channel, *q*=0, and for a triangular channel, *q*=1. The cross-sectional area A_{CS} is the integral of *w* with respect to *d* (Equation 7). To simplify the task (and because it is not sensitive based on our observations), we assumed *p*=21 based on preliminary data fitting to USGS hydraulic geometries from field surveys of gages in the JRB. Note that even though we make this assumption here for model completeness, we do not posit that *q* is invertible from available data because it may not be that significant for the downstream discharge. Moving forward with these assumptions, we can write these relationships as Equation 7:

$$A_{CS} = \int_0^d w \, \partial d = \int_0^d pd^{q} \, \partial d = \frac{pd^{q+1}}{q+1} \tag{7}$$

Combining Equation 7 with Manning's *n* Equation, we come up with Equation 8a. Reorganizing, we derive a function that estimates *d* from *Q* (Equation 8b). With *d*, *p, and q*, we can estimate *v* and *K* using the linear form of Muskingum equation as in Equations 7, 8c, and 8d which close the equations.

$$Q = vA_{CS} = \frac{1}{n} R^{2/3} S_0^{\frac{1}{2}} \frac{pd^{q+1}}{q+1} = \frac{pd^{q+\frac{5}{3}} S_0^{\frac{1}{2}}}{n(q+1)} \tag{8a}$$

$$d = \left[\frac{Q_t n(q+1)}{pS_0^{\frac{1}{2}}} \right]^{\frac{3}{5+3q}} \tag{8b}$$

$$v = \frac{Q_t}{A_{CS}} \tag{8c}$$

$$K = \frac{\Delta x}{v} \tag{8d}$$

Here, S_0 represents the reach slope, Q_t represents the discharge exiting the reach at time t, and Δx is the reach length.

2.3.3 Differentiable modeling

By implementing MC on a differentiable coding platform (PyTorch, Tensorflow, Julia, etc.), we can train a coupled NN in an “online” way to produce physical reach-scale river parameters for the routing model, much like our earlier work in differentiable parameter learning (dPL) (Tsai et al., 2021). Here we include a NN into the MC routing framework to optimize equation parameters based on big data while maintaining physical consistency and mass balances. In this case, a Multilayer Perceptron (MLP) (Leshno et al., 1993) is incorporated. The MLP, featuring two hidden layers and a sigmoid activation function in the output layer, accepts a normalized array of attributes (c) for each reach (Table A2). Based on initial results, we saw no need to add further complexity (additional hidden layers). The network then outputs the Manning's roughness coefficient (n) and channel bathymetry shape coefficient (q):

$$n, q = NN(c) \tag{9}$$

where *n* represents a channel’s resistance to flow and *q* represents the shape of the channel’s cross-sectional area. These parameters are inferred for each reach using the attributes of that reach prior to routing, since we assumed *n* and *q* to be time-invariant. This produces *r* number of *n* and *q* values specific to each reach for all timesteps where *r* is the number of river reaches. The weights of the MLP are updated using backpropagation and the Adam optimizer (Kingma & Ba, 2017).

2.4 Lateral streamflow inputs

Since spatially-distributed runoff is needed to predict runoff in downstream basins, but there is no such data, we employed a pretrained LSTM (Hochreiter & Schmidhuber, 1997) rainfall-runoff model. This LSTM model was similar to those developed and reported in previous streamflow and water quality studies (Feng et al., 2020; Ouyang et al., 2021; Rahmani, Lawson, et al., 2021; Rahmani, Shen, et al., 2021), and we refer the reader to these publications for a more detailed description of these models. After the initial training was done, we chose not to further update the LSTM in order to disentangle the rainfall-runoff and routing parts of the modeling process, testing the robustness of the methodology in the face of errors with simulated runoff. In addition, the test could tell us if other rainfall-runoff models could be used instead. Updating LSTM further could lead to its co-adaptation with the routing module, making the procedure complex.

To briefly summarize, the LSTM model used a combination of basin-averaged attributes, daily meteorological forcings, and volumetric streamflow observations as inputs, and output daily basin discharge. Meteorological forcings (total annual precipitation, downward long-wave radiation flux, downward short-wave radiation flux, pressure, temperature) were obtained from the NASA NLDAS-2 Forcing Data set (Xia et al., 2009, 2012). We selected 29 basin attributes (Table A1 in the Appendix) similar to those chosen in previous LSTM studies (Ouyang et al., 2021). Consistent with Ouyang et al. (2021), we focused on training the LSTM on 3213 gages selected from the USGS Geospatial Attributes of Gages for Evaluating Streamflow II (GAGES-II) dataset (Falcone, 2011) with input data between 1990/01/01 - 1999/12/31. We developed the workflow to obtain forcing data and inputs seamlessly for any small basin in the conterminous United States (CONUS). In this case, we extracted data from HUC8 subbasins and HUC10 watersheds to gather inputs to train our LSTM model and predict discharge, respectively.

When evaluated on the gaging stations in the study area, the model achieved a median daily Nash-Sutcliffe Efficiency (NSE) (Nash & Sutcliffe, 1970) of 0.7849 for the eight gauging stations in the JRB. After training during the period of 1990/01/01 – 1999/12/31, the model was run from 2000/01/01-2009/12/31 to predict discharge for the 17 HUC10 watersheds in the study area:

$$Q' = LSTM(x_{HUC10}, A_{HUC10}) \tag{10}$$

where *Q′* [m^3/s] is the daily runoff for the HUC10 basin, and x_{HUC10} and A_{HUC10} are HUC10-averaged atmospheric forcings and static attribute variables, respectively. Lastly, we computed a mass transfer matrix, which tabulates the fraction of a subbasin draining into a river reach. Each row of the matrix is obtained by dividing the incremental catchment area of reaches inside a subbasin by the total area of that subbasin. Runoff can be distributed to river reaches via a simple matrix multiplication.

Due to the nature of the data used to train the LSTM, it could produce seamless (having no gaps) runoff estimates for the JRB but only on a daily, not hourly, scale. Because MC routing needs to operate on smaller time steps, we quadratically interpolated (Virtanen et al., 2020) daily data into hourly time steps, where each daily measurement occurs at 12:00 hours. For training and evaluating the routing model, we collected observed discharge data for nodes intersecting USGS GAGES-II monitoring stations. Only some time periods of the most downstream gage station were used for training, and other stations were only used for evaluation. The observed discharge data were similarly disaggregated to hourly data.

2.5 Inverse-routing and hyperparameters

There are time zone differences between the forcing data (recorded using UTC) and USGS streamflow (recorded in UTC-5). To address this, we first shifted the LSTM-produced runoff outputs by 5 hours. Because LSTM was trained to predict runoff at the outlet of a basin, with catchment area being an impactful input to the model, it already implicitly considers the time of concentration to the outlet. However, as our modeled river network extends into the subbasins and contains smaller rivers, the routing module explicitly simulates the within-basin concentration process. Ideally, we can use an inverse-routing approach to revert LSTM-predicted runoff to the time before it enters the river network. However, as inverse-routing methods (Pan & Wood, 2013) can be quite involved and were not the focus of the study, we opted for a simple approach that shifted the runoff back in time by τ hours. τ is considered a hyperparameter. To avoid overfitting, we used the same τ value for all the subbasins and all experiments, and determined this value by manually tuning based on the training period. We found $\tau = 9$ (hours) to be a good choice. More complicated procedures could be employed in the future, but this straightforward approach proved to be effective in our case.

Hyperparameters and training period sizes for our differentiable routing model were chosen through repetitive trial and error based on the training period. These trials led us to choose a hidden size of 6 for our MLP, and a training size of eight weeks. Parameters were tuned for 50 epochs for synthetic and real data experiments. Mean Squared Error (MSE) was chosen as our loss function. Since our differentiable model at t=0 assumes no inflow to the river network and relies exclusively on Q' for flow inputs, a period of 72 hours is employed to "warm up" the model states in the river network, and the loss function and NSE are not calculated within this period.

2.6 Experiments

2.6.1 Synthetic Parameter Recovery

We first ran multiple synthetic parameter recovery experiments to check if the dataset and the framework could indeed recover assumed relationships with small training periods of eight weeks. Our first experiment tested if we could correctly recover a single, spatially-constant set of assumed values for both n and q for the whole river network, resulting in only two degrees of freedom. We assumed ranges from 0.01 – 0.3 and 0-3 for the synthetic values of n and q, respectively, to give a realistic value range for the MLP to learn parameters. n and q model parameters were initialized to be at the 90th and 20th percentiles for the first and second set of synthetic experiments, respectfully.

In our second experiment, we assumed constant *n* throughout the reaches but set the trained model as *n,q = NN(c)* (Equation 9) so that the *n, q* values could be different from reach to reach. In this case, ideally, the NN would learn to output a constant value regardless of the inputs.

Our third synthetic experiment examined if we could retrieve simple assumed relationships within realistic literature bounds (inverse-linear or power-law) [Equation 9-10] between *n, q,* and drainage area (DA), given that the MLP had far more inputs than just DA. The trained model is still utilizing Equation 9, as we assumed we did not know the functional relationship *a priori.*

$$n = 0.06 - 8 \times 10^{-6}(DA) \tag{11}$$
$$q = 2 - 0.00018(DA)$$

$$n = \frac{0.0915}{(DA)^{0.131}} \tag{12}$$
$$q = \frac{2.1}{(DA)^{0.357}}$$

2.6.2 Observational Data Experiments.

We trained our differentiable model (updating the weights in NN as in equation 6) against observed USGS data. We utilized eight-week training periods from different years and checked whether the resulting parameters led to satisfactory routing in other years at both the training gage and untrained, inner, gages. Training periods were selected based on times when the LSTM had high accuracy and when there were frequent discharge peaks. Routing frequently fluctuating discharge through a river network introduces more variance into the MLP, allowing it to perform better when testing over a longer time period. Additionally, high LSTM accuracy reduces the noise --- we hypothesize the system has some tolerance to the runoff errors but outsized errors can invalidate the model. Periods of such “high flashiness” in the JRB occurred during both 02/01-03/29 and 11/01-12/26, while the years 2001, 2005, 2007, and 2008 had high LSTM accuracy, giving us eight time periods on which to train NN models. We then trained the differentiable routing models on all eight selected time periods to determine the sensitivity of the model performance to the selected training time period.

When interpreting model performance at inner gages, we compared results with the LSTM that modeled the whole JRB as a uniform basin and a simple summation of the τ-shifted LSTM runoff inputs (Q’). We also explored whether using a combination of inner gages, along with the furthest downstream gage, inside of the loss function would improve model performance on all gages throughout the study area. The gages used were USGS 01560000 (edge 1053) and 01563200 (edge 2689). Internal gages were selected based on NSE metrics when using only the furthest-downstream gage in the loss calculation; we chose basins with middle-level metrics so as to not overfit the model if using highly predictive internal gages.

3. Results and Discussion

In the following, we first discuss our synthetic experiments (Section 3.1) which explore our routing framework's potential to retrieve assumed parameters from our differentiable GNN. Next, we show the results of confronting our model with LSTM-simulated runoff as observed streamflow at the furthest downstream gage, expanding the training period to other time ranges, then applying our models to different years for observation (Section 3.2). Furthermore, we discuss the stability of our trained models over several years of testing (Section 3.3). Lastly, we analyze the *n* parameters recovered for the trained models and discuss their implications (Section 3.4).

3.1 Synthetic experiments

Our first synthetic experiment (with constant parameters and only 2 degrees of freedom for the search) recovered the assumed *n* values with moderate accuracy, but not the channel geometry parameter *q* (Table 1). Recovered *n* values were within a small range of the assumed ones, with minor fluctuations, while recovered *q* values mostly stayed similar the initial guesses, showing slight changes after a number of iterations. This result was consistent across 10 runs, each with different "synthetic truth" values for *n* and *q*. The training led *n* to the assumed values rapidly, typically within 20 epochs (Figure A1). The non-identifiability of *q* was likely because *q* has only a small influence on the storage capacity of the stream and the simulated discharge is not sensitive to *q*, making *dL/dq* (where *L* is the loss function) negligible. While it is a pity that hydraulic geometry parameters cannot be estimated, the results also implied that they would not influence the routing results noticeably. Thus, in our efforts, we focused on *n.*

Table 1: Results from the constant synthetic n and q parameter recovery experiments

Run	*n*			*q*		
	Initial Guess	Synthetic Truth	Recovered Parameter	Initial Guess	Synthetic Truth	Recovered Parameter
1	0.271	0.03	0.028	2.7	2	2.327

2	0.271	0.04	0.035	2.7	2	2.37
3	0.271	0.05	0.046	2.7	2.5	2.390
4	0.271	0.06	0.059	2.7	2.5	2.456
5	0.271	0.07	0.070	2.7	3	2.480
6	0.068	0.03	0.030	0.6	1.0	0.574
7	0.068	0.04	0.042	0.6	1.0	0.592
8	0.068	0.05	0.055	0.6	1.5	0.730
9	0.068	0.06	0.067	0.6	1.5	0.777
10	0.068	0.07	0.087	0.6	2.5	0.690

Our second synthetic experiment (assuming constant *n* to be recovered by NN(A)) showed that we were able to recover the constant value that was set using an NN, but there was some scattering for the headwater reaches (Figure 3c, 3f). We noticed trends associated with drainage area (DA), which is correlated with reach positioning in the watershed; small DA often indicates a headwater reach, while large DA often indicates a reach much further downstream. There were some visible differences between the synthetic hydrographs resulting from different assumed *n* values (comparing Figures 3a and 3c), which allowed the recovered *n* values to mostly center around the assumed value. However, the scattering of points toward the lower-DA part of Figures 3b and 3d alluded to the fact that the downstream discharge was strong enough to completely constraint on the model. *n* in different ranges can fluctuate around the mean to generate essentially the same pattern as a constant *n* value.

In our third set of synthetic experiments, the simple functions could be roughly recovered for most of the reaches, while there may have been increased uncertainty for the furthest downstream reaches (Figure 3f & 3h). There were again noticeable differences in the hydrographs (Figures 3e & 3g) from previous ones. When the power-law relationship was assumed, the hydrograph matched the synthetic one almost completely (Figure 3e), and the estimated *n* outputs from the MLP overlapped to a great extent with the value to be retrieved (Figure 3f). The headwater reaches (small-DA) showed a rapid decline in *n* with respect to increasing DA. In the middle ranges of DA, the curve followed the assumed one almost exactly. Toward the higher range of DA, the recovered values were lower than the assumed relationship, but the deviation was not huge because the power-law formulation became flat in this range. Based on the closeness of hydrographs in all of Figure 3, we do not anticipate that further optimization can bring significant improvement to the estimations. Similar to the two-constant-parameter retrieval experiment, the *q* parameter was not recoverable and thus is not shown here.

Based on these simple experiments, it seems training on the river graphs has some promise but also some limitations. It is promising because it is likely that *n* is related to DA which is, to some extent, recoverable. It is simultaneously challenging because, as a large number of reaches contribute to one gage, it is an underdetermined system. This method was not able to fully reproduce the drastic change in the low-DA range presumably because this sharp slope was inconsistent with the rest of the curve, and NNs generally do not output extreme values. It also ran into difficulty toward the high-DA range because there were simply far fewer reaches with large DA so their roles in routing were relatively minor, making the curve unconstrained in this range. This experiment informed us we should not expect values of reach-scale *n*, particularly in the high-DA range, to be reliable, but the overall trend may have merit, especially when we also have other constraints. These findings formed the basis for the next stage of the work where we trained n=NN($\boldsymbol{c}$) for real-world data. We thus expected to extract the overall patterns of *n* distribution but for the recovered *q* not to be meaningful.

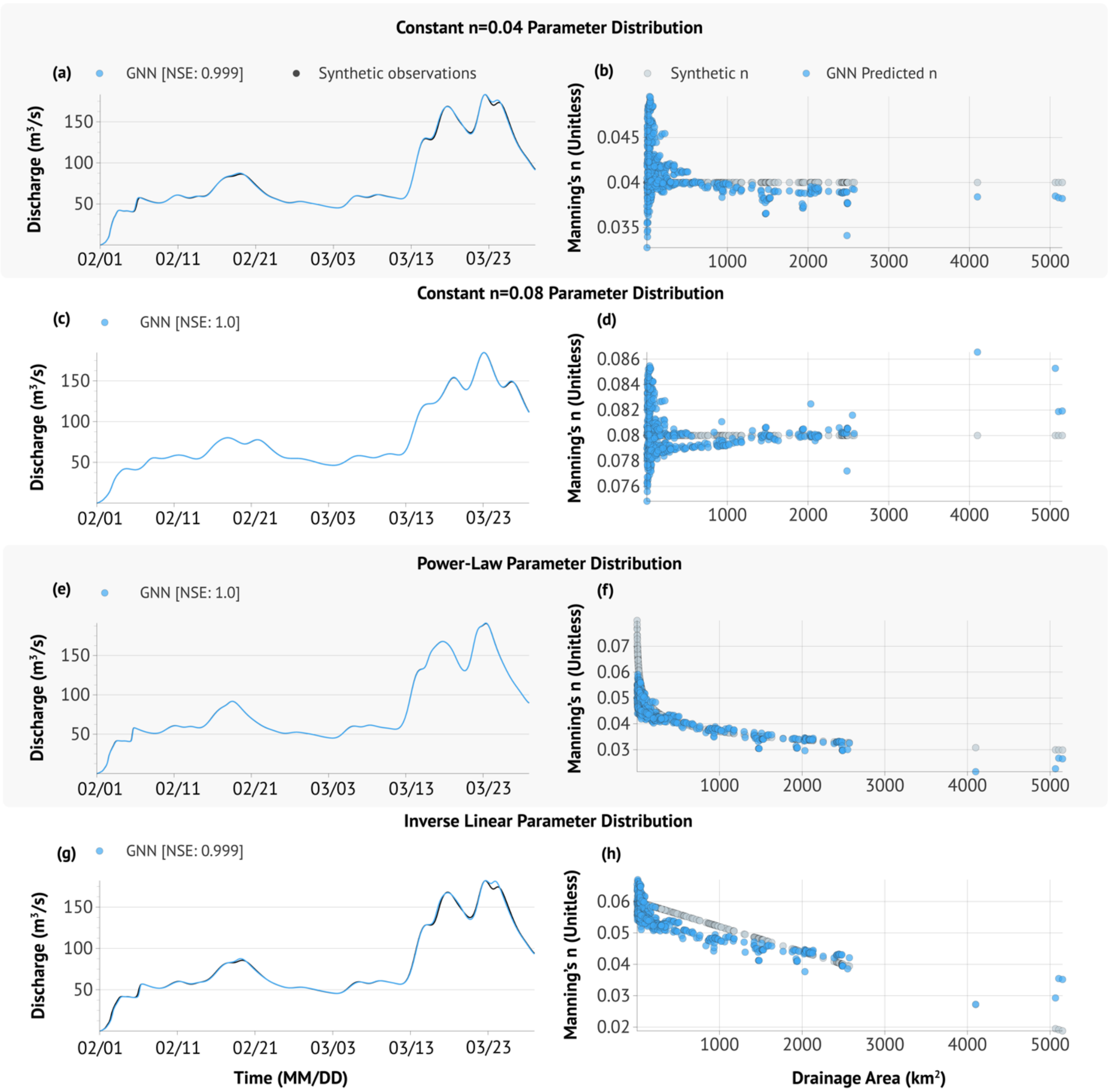

Figure 3: Synthetic discharge distribution experiments. (a, c, e, g) Synthetic and modeled discharge over time for various assumed relationships between n and drainage area. (b, d, f, h) Synthetic modeled values of n with respect to the reach's total drainage area (km^2*). The NN can recover the overall pattern but is not accurate near sharp changes or for reaches with large drainage areas. Each dot in the scatter plots represents a 2-km river reach in the river network.*

3.2. Training on eight weeks of real data

The real-world data experiment showed satisfactory streamflow routing in the training period, with improvements compared to approaches that did not employ the routing scheme, even though there was significant bias in the rainfall input (Figure 4a). The hydrograph generated by the differentiable routing model is, as expected, smoothed and delayed compared to the summation of runoffs during the training period. Unlike the direct summation of the runoff, which has a timing difference from the observation, the peaks of the routed hydrograph are placed almost exactly under the observed peaks, leading to a high training NSE of 0.834. We noticed a substantial low bias in this training period, witnessed by much lower peaks with the simulated flow compared to the observed flow. This is due to bias in the rainfall-runoff modeling component and the mass-balance dictated by the MC formulation, which prevents the model from adding or removing mass to remove the bias. In traditional hydrologic model calibration, bias can be a significant concern as it can distorts other parameters. In this case, we found the model performed well even with such bias, and appropriately focused on adjusting the timing of the flood waves. This is because the allowable adjustments were limited to routing parameters, which blocked the model from distorting other processes.

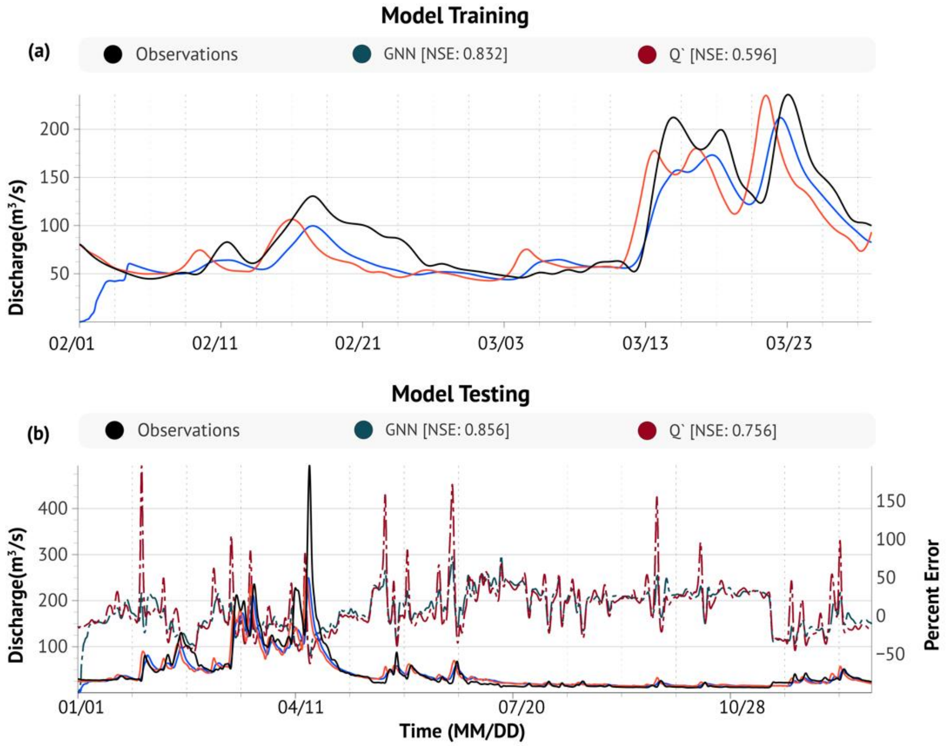

Figure 4: (a) Results from training the differentiable model during an eight-week period (2001) against USGS observations compared with the summation of lateral inputs (denoted by Q'). (b) Results from testing the trained model from Figure 4(a) over a year period (2001) compared with the summation of lateral inputs. A percent error has been overlaid to the graph to show how river routing is more stable than using a summation of lateral inputs.

The year-long test of the differentiable model yielded high metrics compared to the alternatives (Figure 4b), suggesting a short calibration period could yield parameterization suitable for long-term simulations. The differentiable model obtained a year-long NSE of 0.857, which is consistent with the median NSE in the JRB. In contrast, the summation of $Q'(\tau = 9)$ and the whole-basin $LSTM$ were at 0.756 and 0.801, respectively. This comparison shows that if we merely added the runoffs together (which already resolved spatial heterogeneity in runoff but not the flow process), the error due to timing could reduce NSE at the downstream gage. While the model had success with correctly timing the peak flows, it could not compensate for LSTM's errors, resulting in significant underestimation of the peak events. By design, the routing module should be detached from the errors in the runoff module.

Interestingly, without specific instructions, the scheme recovered a power-law-like relationship between *n* and drainage area (DA) (Figure 5), similar to the one assumed in the synthetic case (Figure 3e &3f). The *n* values were highest (near *n*=0.04) for smaller DA and declined gradually, approaching 0.015 at the lower end. The change rate of *n* as a function of DA then became more gentle as DA increased. This distribution agreed well with the general understanding that headwater streams running down ridges (this region is characterized by Ridge and Valley formations) have larger slopes, higher roughness, more vegetation, and thus higher *n*, while the high-order streams in the valley tend to have smaller slopes and smoother beds, corresponding with lower *n*. In most hydrologic handbooks (Mays, 2019), a smaller *n* is prescribed for larger rivers.

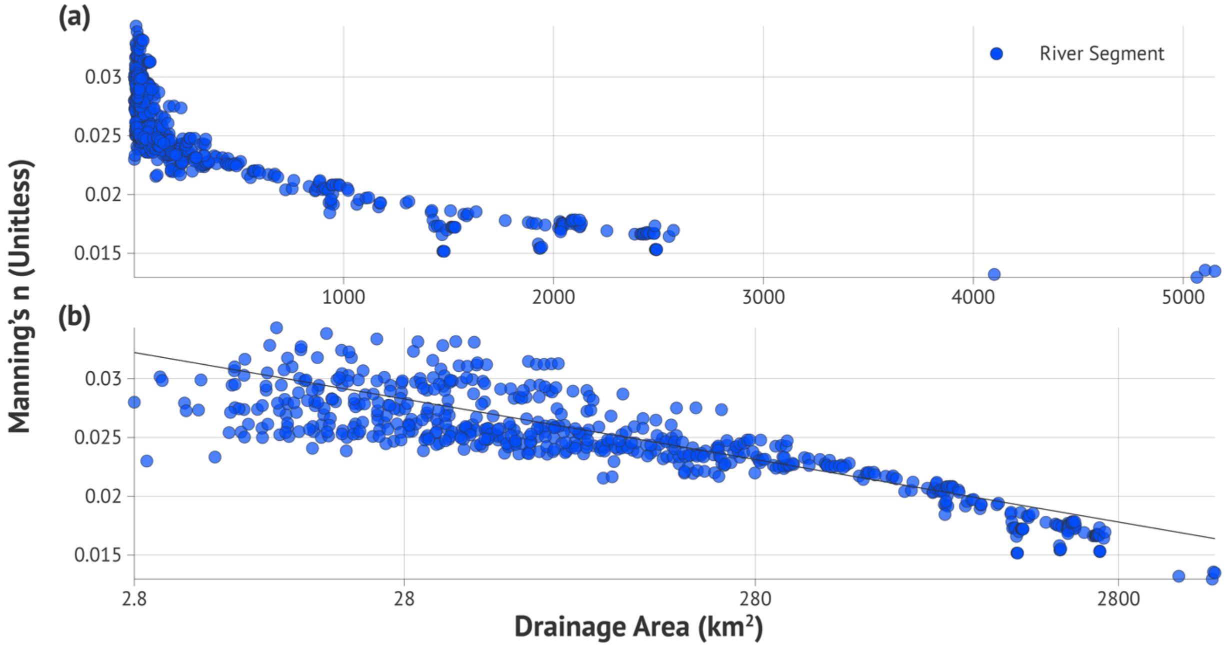

Figure 5: The learned relationship between n and drainage area (square kilometers) for the Juniata River basin according to the trained GNN. (a) The distribution on a linear scale. (b) The distribution on a logarithmic scale. The network was trained for the period of 2001/02/01-2001/03/29. Each dot in the scatter plot represents a 2-km river reach.

3.3. Inner gage evaluation and effects of different training periods

Evaluating the model on the inner, untrained gages showed that the routing scheme became more competitive compared to benchmark levels as for downstream gages (Table 2). As for the benchmarks, the uniform LSTM (the catchment area of each gage is consider a basin and basin-averaged forcing/attributes were used as inputs to the trained LSTM to simulate flow at the gage) already attempts to consider routing internally but does not consider rainfall/attribute spatial heterogeneity, while the summation of Q' (runoffs were simulated from multiple HUC10 basins and added together) considers the spatial heterogeneity but not routing in the stem river. For 2 of the 4 gages with larger than ~2000 km^2 of catchment area, the differentiable routing model performed noticeably better than the uniform LSTM models for them (for the other two, they were about the same). For the three midsized subbasins (500-2000 km^2), the comparisons were mixed. For the small subbasins, and especially gage 01557500 (94.8 km^2), the uniform LSTM was noticeably better. The subbasin for 01557500 is smaller than our runoff-producing unit (HUC10s, with the smallest one ~200 km^2). This means predictions below this threshold can be error-prone. Our model was also better than the summation of Q’ for 7 of the 8 gages and the gap was larger for downstream gages (Table 2), suggesting the flow convergence process matters more and more as we go downstream.

When we used multiple internal gages within the NN loss function, results improved very slightly at smaller DA gages, while degraded barely noticeably at larger DA reaches. Overall, the differences are too small to have real-world implications, but we can still observe the pattern that the multi-gage calibration appears to produce a slightly more balanced model that improves simulations at some previously weakly-simulated tributaries, at a (very minor) cost at the most downstream one. This small tradeoff may be due to spatial errors in forcing data. As the model explicitly simulates flows in all modeled reaches, the differentiable model provides a way to absorb data from as many stations as possible, if the ungauged regions are important to the users.

Table 2: Internal gage NSE values for the year 2001, with the rows ranked by the size of the subbasin from small to large. The differentiable routing model was trained on the period from 2001/02/01-2001/03/29 calculating loss from the final gage but the LSTM was trained using >3000 CONUS gages. We include the LSTM NSE to show how the use of routing compares to just using LSTM predictions. Bold font indicates the top performing model for each gage.

Edge ID	Gage Number	Basin Drainage Area (km^2)	Uniform LSTM	Q` Runoff NSE (τ = 9)	Differentiable routing model (τ = 9)	Multiple Gage Loss for differentiable routing (τ = 9)
1280	01557500	94.8	**0.8149**	0.5575	0.5623	0.5627
1053	01560000	440.5	**0.7028**	0.6054	0.6578	0.6625
2799	01558000	542.1	**0.8201**	0.7473	0.6963	0.6981
4780	01556000	723.5	0.6624	0.6568	0.6937	**0.6957**
2662	01562000	1943.5	0.7957	0.6857	0.7942	**0.7977**
4801	01559000	2103.0	0.7815	0.7449	0.8136	**0.8172**

2689	01563200	2482.9	0.5703	0.6497	**0.7831**	0.7773
4809	01563500	5212.8	0.8024	0.7563	**0.857**	0.8546

The above comparisons informed us of the favorable and unfavorable ranges of applicability for our workflow: the differentiable model found competitive advantages for stem rivers with catchments greater than 2,000 km^2, but may run into issues for scales smaller than the smallest runoff-producing unit (HUC10, around 200 km^2). The issues for the smallest basins could be attributed to the procedure that transfers mass from subbasin to regular grids on the river network, which should be improved in future work. As a result, the smallest headwater basins are best to be directly simulated by the uniform LSTM models. Also, smaller runoff-generating units could be used in the future to mitigate this issue. The advantages of the differentiable routing model over the uniform LSTM for larger basins were due to resolving the heterogeneity in rainfall and basin static attributes as well as better representing routing. The uniform LSTM can internally represent some flow lags but it appears less effective as basin size increases.

The results imply that the advantages will increase for even larger basins, where currently LSTM does not apply well, along with basins where rainfall heterogeneity makes a big difference. The JRB is situated in the northeastern part of the CONUS; many other regions may exhibit more prominent effects of heterogeneity. For example, past studies have always found it difficult to simulate large basins on the northern and central Great Plains (Feng et al., 2020; Martinez & Gupta, 2010), potentially due to spatially-concentrated rainfall and runoff generation (Fang & Shen, 2017). Also, in the mountainous areas of the CONUS Northwest and Southeast, orographic precipitation could have significant spatial concentration. We hypothesize applying models to smaller basins and incorporating the routing scheme will allow these regions to be better modeled.

As expected, the training periods selected can exert an influence on the model, but as long as we used reasonable training periods, the results were acceptable. When the scheme was trained on eight-week periods from different years, it generated somewhat different but mostly functional parameterizations (Figure A2 in the Appendix), unless it was trained in some unreasonable training periods where the LSTM had drastic differences from the observed outflows (Table 3). The maximum achievable NSEs for the years of 2001, 2005, 2007, and 2008 were 0.857, 0.87, 0.827, and 0.787, respectively, with all models outperforming Q` NSE values for their respective periods (Table A3 in the Appendix). We found that if the models were trained on other periods (2001a, 2001b, 2005b, 2007a), the test NSEs were mostly decent, and at least not drastically worse. However, choosing 2007b or 2008a led to notably inferior results (Figure 6b-e). Examining the characteristics of the different training periods, we see that the problematic training periods did not contain full flood rise and recession phases (Figure 6a & 6b). As a result, 2007b and 2008a as training periods led to either the lowest or the highest *n* values and also had relatively low NSE values (Figure A2 in the Appendix). Similarly, training period 2005a gave relatively large *n* values which also resulted in suboptimal (although still decent) results in all the years. Hence, we need to pick periods that (i) contain full flood rise and recession phases; and (ii) have high runoff NSEs. In addition, even though the routing simulation can be improved by short training periods, the spread of estimated *n* again shows that the identification of *n* via small training periods can be difficult. Future work could employ longer training periods to compromise across different periods and obtain broadly-performant parameterization. However, another possibility is that *n* itself can vary over time, which would be an orthodoxy but not unthinkable idea.

Table 3. The NSE values correspond to testing differentiable models on different test years. Bold font indicates the highest NSE, while underlined metrics indicate the lowest (noticeably worse than obtained from other periods) for the testing period.

Testing Period	Training Period							
	2001a 02/01-3/29	2001b 11/01-12/26	2005a 02/01-3/29	2005b 11/01-12/26	2007a 02/01-3/29	2007b 11/01-12/26	2008a 02/01-3/29	2008b 11/01-12/26
2001	**0.857**	0.845	0.850	0.853	0.857	0.831	<u>0.782</u>	0.856
2005	0.797	0.828	0.843	**0.870**	0.816	<u>0.713</u>	0.785	0.785
2007	0.815	0.812	0.821	**0.827**	0.819	0.774	<u>0.753</u>	0.813
2008	0.643	0.715	0.723	0.762	0.676	<u>0.534</u>	**0.787**	0.623
Average	0.778	0.800	0.809	**0.828**	0.792	<u>0.713</u>	0.777	0.769

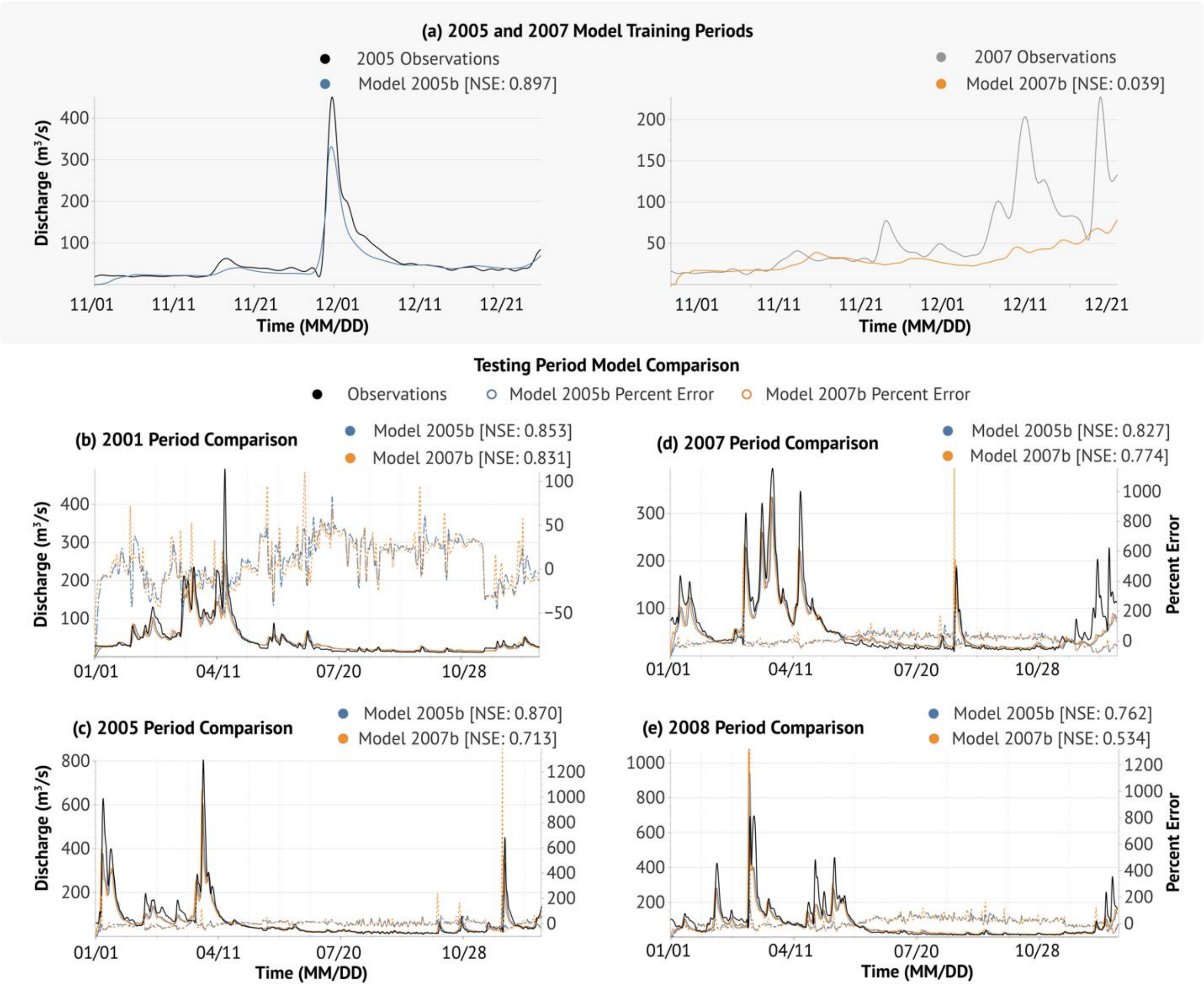

Figure 6: (a) Two training periods: 2005 and 2007a. The former contains a full rising-recession cycle while the latter does not have a complete cycle for training, thus leading to larger errors during test. The solid line indicates the training of Model 2005b while the dashed line indicates Model 2007b during the time period of 11/01-12/27 during the years 2005 and 2007, respectively. (b-c) Test periods for these two models: (b) 2001, (c) 2005, (d) 2007, and (e) 2008. For (b-e) the solid line indicates discharge while the dashed line indicates percent error of each model's output compared with the observations.

3.4. Further discussion

Although the estimated *n* values were both functional for routing streamflow and physically meaningful, the results suggest the downstream discharge only poses a moderate constraint on the *n* values, and short training periods may not be sufficient to identify the true *n* values. Hence, while our procedure can obtain *n* parameterization performant for long-term simulations, we do not claim that the procedure retrieved the “true” *n* parameterization. Especially considering there are many input variables to the NN that covary in space, it may be difficult to disentangle causation from correlation. Due to the lack of ground truth for *n* in the real-data case, we leave this evaluation for future effort as we compile more measurement data. Recall that we were able to retrieve the overall pattern of *n* in the synthetic experiments but faced large uncertainties in some areas of the parameter space. This is attributed to the numerous degrees of freedom (a high-dimensional input space for the NN, influencing many reaches) constrained by only one downstream output with a relatively short training period. Nevertheless, this training is valuable because discharge data can be widely available, and we will be able to employ it in conjunction with other constraints, e.g., scattered measurements or expert-specified relationships.

Regarding other potential recoverable parameters, we suspect the dimensionless MC inflow/outflow weighing parameter X, which indicates the shape of the assumed flood prism, cannot be identified for the same reason as q --- the geometries of the channel do not impact flow rates in a meaningful way. Future work could investigate if learning it produces any benefit. Similarly, linear channel coefficient *p* values were also never recoverable in single parameter tests and decreased resulting NSE values when used as a tunable parameter. Thus, we did not include it in this study. In addition, we hypothesize using more complex MC formula, e.g., the nonlinear form of the Cunge equation (the celerity is defined as *dQ/dA*), which might add to numerical challenges for large-scale simulations, would lead to different *n* values, as the recovered values are inherently linked to the inverse model employed.

Here we employed a static parameterization scheme for *n*, following the conventional approach. However, the framework allows for the use of a dynamic *n* (likely dependent on Q). It is not clear if we must use a static parameterization as done conventionally, as some previous studies have found a dynamic *n* to offer better results (Ye et al., 2018). In the future, it will be interesting to see if a dynamical *n* parameterization could significantly impact the routing results. On another note, we chose an eight-week time period as our training length as a probe to assess the required training duration and selection criteria for such periods. We trained eight different models (Section 3.3) on different time periods and showed that the choice of training period timing, and LSTM performance for the inputs played important roles. Future effort should include longer training periods to most robustly estimate the parameters.

When investigating the impact of multiple gages, rather than a single downstream-most gage (in model loss calculation and parameter updates), results were very similar in terms of NSE score and recovered Manning's *n* parameters. We believe this may be because the JRB is a relatively small river network, so internal gage observations are highly correlated in discharge volume (m^3/s) and fluctuation (storm event timing). Adding more gages could be useful if flows in different parts of the basin need to be accurately reported, but may be less important if only the downstream gage is of concern.

Our approach, akin to a classical routing scheme, is modular --- the trained weights of the NN that generates *n* are not tied to a particular runoff model. Our work can be coupled to traditional models in multiple ways. Firstly, the trained network can be used to generate *n* for traditional models. In this way, no change is required on the part of the traditional models. Secondly, the neural network and the trained weights can be ported to other programming environments like Fortran. This makes it possible to use the trained parameterizations as a built-in module in continental-scale models (Greuell et al., 2015; Johnson et al., 2019; Regan et al., 2018). An alternative approach is to lump both the routing and runoff simulations into one problem and optimize them together, as demonstrated in some other studies (Jia et al., 2021). In our case, this would mean that we would train both the runoff LSTM and the routing module together. In many big-data DL case studies, lumped models tend to have higher performance compared to a workflow that separates the tasks into multiple minor tasks. However, in our case here, this leads to coadaptation concerns. Moreover, our approach is modular so it can be easily coupled to other runoff models, e.g., a non-differentiable traditional model, or a differentiable one (Feng, Beck, et al., 2022; Feng, Liu, et al., 2022).

4. Conclusions

In this work, we used a combination of a pre-trained LSTM rainfall-runoff model and Muskingum-Cunge routing to create a learnable routing model, or, from the perspective of machine learning, a physics-informed graph neural network. This model predicts streamflow in stem rivers and learn river parameters throughout a river network, which is urgently needed to improve the next-generation large-scale hydrologic models. Because our framework is built on physical principles and estimates widely-used *n* values, it can be easily ported to work with other models. For example, the trained NN and the weights can be loaded into Fortran or C programs to support traditional hydrologic models or routing schemes, e.g. (H. Li et al., 2013; Mizukami et al., 2016). Our synthetic experiments recovered the overall spatial pattern of *n* with moderate accuracy but could not recover the channel cross-sectional geometry parameter (*q*). Furthermore, our synthetic experiments yielded promising results in recovering synthetic *n* and drainage area relationships, implying there is potential to learn reach-scale physics in the river network using differentiable modeling.

With the real-world data, short-term training periods of downstream hydrographs can produce *n* parameterization that improve long-term routing results, but may be insufficient to constrain the *n* values more precisely than a general spatial pattern. Eight weeks of real-world data produced decent long-term streamflow routing and improved upon approaches that did not use routing, yet training on different periods could result in somewhat different distributions. When looking at the *n* vs drainage area distribution attained by our trained model against USGS observations, we found that the *n* values agreed with the literature bounds for the area, but the absolute magnitudes may fluctuate depending on the training period. Besides using longer training periods to obtain *n* values that compromise across periods, future work should also consider if *n* should be treated as dynamic in time. Further work can expand this analysis to other basins with different conditions (streams outside of the Ridge and Valley physiographic division of the CONUS) to see if the model can still identify their trends correctly. Reviewing the internal gage NSE scores over a full year of data showed a correlation between drainage area and the relative advantage of our routing scheme, highlighting the impacts of heterogeneity and flow convergence.

Open Research

The LSTM streamflow model code (Feng et al., 2020; Ouyang et al., 2021) relevant to this work can be accessed via Zenodo (Shen, Fang, et al., 2021). The differentiable routing model will be made available to reviewers upon a paper revision request, and a new Zenodo release will be published upon paper acceptance. All datasets used are publicly available, including the GAGES-II dataset (Falcone, 2011), NHDPlus (HorizonSystems, 2016), and NLDAS (Xia et al., 2012). Other data sources can be found in Table A1.

Funding Acknowledgements

Funding was awarded to Cooperative Institute for Research to Operations in Hydrology (CIROH) through the NOAA Cooperative Agreement with The University of Alabama (NA22NWS4320003).

References

Aboelyazeed, D., Xu, C., Hoffman, F. M., Jones, A. W., Rackauckas, C., Lawson, K. E., & Shen, C. (2022). A differentiable ecosystem modeling framework for large-scale inverse problems: demonstration with photosynthesis simulations. *Biogeosciences Discussions*. https://doi.org/10.5194/bg-2022-211

Adnan, R. M., Petroselli, A., Heddam, S., Santos, C. A. G., & Kisi, O. (2021). Comparison of different methodologies for rainfall–runoff modeling: machine learning vs conceptual approach. *Natural Hazards*, *105*(3), 2987–3011. https://doi.org/10.1007/s11069-020-04438-2

Arcement, G. J., & Schneider, V. R. (1989). *Guide for Selecting Manning's Roughness Coefficients for Natural Channels and Flood Plains* (Water-Supply Paper No. 2339). U.S. Geological Survey. Retrieved from https://pubs.usgs.gov/wsp/2339/report.pdf

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, *18*(153), 1–43. Retrieved from http://jmlr.org/papers/v18/17-468.html

Candela, A., Noto, L. V., & Aronica, G. (2005). Influence of surface roughness in hydrological response of semiarid catchments. *Journal of Hydrology*, *313*(3), 119–131. https://doi.org/10.1016/j.jhydrol.2005.01.023

Carabajal, C. C., & Harding, D. J. (2006). SRTM C-Band and ICEsat laser altimetry elevation comparisons as a function of tree cover and relief. *Photogrammetric Engineering & Remote Sensing*, *72*(3), 287–298. https://doi.org/10/ggj69r

Chaney, N. W., Minasny, B., Herman, J. D., Nauman, T. W., Brungard, C. W., Morgan, C. L. S., et al. (2019). POLARIS Soil Properties: 30-m Probabilistic Maps of Soil Properties Over the Contiguous United States. *Water Resources Research*, *55*(4), 2916–2938. https://doi.org/10/ggj68b

Cunge, J. A. (1969). On the subject of a flood propagation computation method (Musklngum method). *Journal of Hydraulic Research*, *7*(2), 205–230. https://doi.org/10.1080/00221686909500264

Dottori, F., Szewczyk, W., Ciscar, J.-C., Zhao, F., Alfieri, L., Hirabayashi, Y., et al. (2018). Increased human and economic losses from river flooding with anthropogenic warming. *Nature Climate Change*, *8*(9), 781–786. https://doi.org/10.1038/s41558-018-0257-z

Douben, K.-J. (2006). Characteristics of river floods and flooding: a global overview, 1985–2003. *Irrigation and Drainage*, *55*(S1), S9–S21. https://doi.org/10.1002/ird.239

Duan, S., Ullrich, P., & Shu, L. (2020). Using convolutional neural networks for streamflow projection in California. *Frontiers in Water*, *2*. https://doi.org/10.3389/frwa.2020.00028

Falcone, J. A. (2011). GAGES-II: Geospatial Attributes of Gages for Evaluating Streamflow [Data set]. *GAGES-II: Geospatial Attributes of Gages for Evaluating Streamflow*. USGS Unnumbered Series, Reston, VA: U.S. Geological Survey. https://doi.org/10.3133/70046617

Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., & Yin, D. (2019, November 22). Graph neural networks for social recommendation. arXiv. https://doi.org/10.48550/arXiv.1902.07243

Fang, K., & Shen, C. (2017). Full-flow-regime storage-streamflow correlation patterns provide insights into hydrologic functioning over the continental US. *Water Resources Research*, *53*(9), 8064–8083. https://doi.org/10.1002/2016WR020283

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

Fang, K., 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

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

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

Feng, D., Beck, H., Lawson, K., & Shen, C. (2022). The suitability of differentiable, learnable hydrologic models for ungauged regions and climate change impact assessment. *Hydrology and Earth System Sciences Discussions*, 1–28. https://doi.org/10.5194/hess-2022-245

France-Presse, A. (2022, June 19). At least 59 dead and millions stranded as floods devastate India and Bangladesh. *The Guardian*. Retrieved from https://www.theguardian.com/world/2022/jun/18/at-least-18-dead-and-millions-stranded-as-floods-devastate-india-and-bangladesh

François, B., Schlef, K. E., Wi, S., & Brown, C. M. (2019). Design considerations for riverine floods in a changing climate – A review. *Journal of Hydrology*, *574*, 557–573. https://doi.org/10.1016/j.jhydrol.2019.04.068

Friedl, M., & Sulla-Menashe, D. (2019). MCD12Q1 MODIS/Terra+Aqua Land Cover Type Yearly L3 Global 500m SIN Grid V006 [Data set]. https://doi.org/10.5067/MODIS/MCD12Q1.006

Getirana, A. C. V., Boone, A., Yamazaki, D., Decharme, B., Papa, F., & Mognard, N. (2012). The Hydrological Modeling and Analysis Platform (HyMAP): Evaluation in the Amazon Basin. *Journal of Hydrometeorology*, *13*(6), 1641–1665. https://doi.org/10/f4jbcx

Gleason, C. J. (2015). Hydraulic geometry of natural rivers: A review and future directions. *Progress in Physical Geography*. https://doi.org/10/f7dsqm

Greuell, W., Andersson, J. C. M., Donnelly, C., Feyen, L., Gerten, D., Ludwig, F., et al. (2015). Evaluation of five hydrological models across Europe and their suitability for making projections under climate change. *Hydrology and Earth System Sciences Discussions*, *12*(10), 10289–10330. https://doi.org/10.5194/hessd-12-10289-2015

He, M., Wu, S., Huang, B., Kang, C., & Gui, F. (2022). Prediction of total nitrogen and phosphorus in surface water by deep learning methods based on multi-scale feature extraction. *Water*, *14*(10), 1643. https://doi.org/10.3390/w14101643

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *06*(02), 107–116. https://doi.org/10.1142/S0218488598000094

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

HorizonSystems. (2016). NHDPlus version 2 [Data set]. Retrieved from http://www.horizon-systems.com/nhdplus/NHDplusV2_home.php

Hrnjica, B., Mehr, A. D., Jakupović, E., Crnkić, A., & Hasanagić, R. (2021). Application of deep learning neural networks for nitrate prediction in the Klokot River, Bosnia and Herzegovina. In *2021 7th International Conference on Control, Instrumentation and Automation (ICCIA)* (pp. 1–6). https://doi.org/10.1109/ICCIA52082.2021.9403565

Huscroft, J., Gleeson, T., Hartmann, J., & Börker, J. (2018). Compiling and mapping global permeability of the unconsolidated and consolidated Earth: GLobal HYdrogeology MaPS 2.0 (GLHYMPS 2.0). *Geophysical Research Letters*, *45*(4), 1897–1904. https://doi.org/10.1002/2017GL075860

International Panel on Climate Change (IPCC). (2012). *Managing the Risks of Extreme Events and Disasters to Advance Climate Change Adaptation* (p. 582). Retrieved from https://www.ipcc.ch/report/managing-the-risks-of-extreme-events-and-disasters-to-advance-climate-change-adaptation/

Ji, X., Lesack, L., Melack, J. M., Wang, S., Riley, W. J., & Shen, C. (2019). Seasonal and inter-annual patterns and controls of hydrological fluxes in an Amazon floodplain lake with a surface-subsurface processes model. *Water Resources Research*, *55*(4), 3056–3075. https://doi.org/10/gghp4s

Jia, X., Zwart, J., Sadler, J., Appling, A., Oliver, S., Markstrom, S., et al. (2021). Physics-Guided Recurrent Graph Model for Predicting Flow and Temperature in River Networks. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)* (pp. 612–620). Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611976700.69

Johnson, J. M., Munasinghe, D., Eyelade, D., & Cohen, S. (2019). An integrated evaluation of the National Water Model (NWM)–Height Above Nearest Drainage (HAND) flood mapping methodology. *Natural Hazards and Earth System Sciences*, *19*(11), 2405–2420. https://doi.org/10.5194/nhess-19-2405-2019

Kalyanapu, A. J., Burian, S. J., & McPherson, T. N. (2009). Effect of land use-based surface roughness on hydrologic model output. *Journal of Spatial Hydrology*, *9*(2), 51–71. Retrieved from https://scholarsarchive.byu.edu/josh/vol9/iss2/2

Khorashadi Zadeh, F., Nossent, J., Sarrazin, F., Pianosi, F., van Griensven, A., Wagener, T., & Bauwens, W. (2017). Comparison of variance-based and moment-independent global sensitivity analysis approaches by application to the SWAT model. *Environmental Modelling & Software*, *91*, 210–222. https://doi.org/10.1016/j.envsoft.2017.02.001

Kingma, D. P., & Ba, J. (2017, January 29). Adam: A method for stochastic optimization. arXiv. https://doi.org/10.48550/arXiv.1412.6980

Koks, E. E., & Thissen, M. (2016). A multiregional impact assessment model for disaster analysis. *Economic Systems Research*, *28*(4), 429–449. https://doi.org/10.1080/09535314.2016.1232701

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

Leopold, L. B., & Maddock, T. Jr. (1953). The hydraulic geometry of stream channels and some physiographic implications. *USGS Professional Paper*, *252*. https://doi.org/10/ggj7hw

Leshno, M., Lin, V. Ya., Pinkus, A., & Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, *6*(6), 861–867. https://doi.org/10/bjjdg2

Li, H., Wigmosta, M. S., Wu, H., Huang, M., Ke, Y., Coleman, A. M., & Leung, L. R. (2013). A physically based runoff routing model for land surface and earth system models. *Journal of Hydrometeorology*, *14*(3), 808–828. https://doi.org/10/ggj7ph

Li, H.-Y., Tan, Z., Ma, H., Zhu, Z., Abeshu, G. W., Zhu, S., et al. (2022). A new large-scale suspended sediment model and its application over the United States. *Hydrology and Earth System Sciences*, *26*(3), 665–688. https://doi.org/10.5194/hess-26-665-2022

Lin, G.-Y., Chen, H.-W., Chen, B.-J., & Yang, Y.-C. (2022). Characterization of temporal PM2.5, nitrate, and sulfate using deep learning techniques. *Atmospheric Pollution Research*, *13*(1), 101260. https://doi.org/10.1016/j.apr.2021.101260

Liu, J., Rahmani, F., Lawson, K., & Shen, C. (2022). A multiscale deep learning model for soil moisture integrating satellite and in situ data. *Geophysical Research Letters*, *49*(7), e2021GL096847. https://doi.org/10.1029/2021GL096847

Liu, L., Ao, T., Zhou, L., Takeuchi, K., Gusyev, M., Zhang, X., et al. (2022). Comprehensive evaluation of parameter importance and optimization based on the integrated sensitivity analysis system: A case study of the BTOP model in the upper Min River Basin, China. *Journal of Hydrology*, *610*, 127819. https://doi.org/10.1016/j.jhydrol.2022.127819

Martinez, G. F., & Gupta, H. V. (2010). Toward improved identification of hydrological models: A diagnostic evaluation of the "abcd" monthly water balance model for the conterminous United States. *Water Resources Research*, *46*(8). https://doi.org/10.1029/2009WR008294

Mays, L. W. (2010). *Water Resources Engineering* (2nd edition). Tempe, AZ: Wiley.

Mays, L. W. (2019). *Water Resources Engineering* (3rd edition). Tempe, AZ: Wiley. Retrieved from https://www.wiley.com/en-us/Water+Resources+Engineering%2C+3rd+Edition-p-9781119493167

Meyal, A. Y., Versteeg, R., Alper, E., Johnson, D., Rodzianko, A., Franklin, M., & Wainwright, H. (2020). Automated cloud based long short-term memory neural network based SWE prediction. *Frontiers in Water*, *2*. https://doi.org/10.3389/frwa.2020.574917

Mizukami, N., Clark, M. P., Sampson, K., Nijssen, B., Mao, Y., McMillan, H., et al. (2016). mizuRoute version 1: A river network routing tool for a continental domain water resources applications. *Geoscientific Model Development*, *9*(6), 2223–2238. https://doi.org/10.5194/gmd-9-2223-2016

Moore, R. B., & Dewald, T. G. (2016). The road to NHDPlus — Advancements in digital stream networks and associated catchments. *JAWRA Journal of the American Water Resources Association*, *52*(4), 890–900. https://doi.org/10.1111/1752-1688.12389

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/fbg9tm

O, S., & Orth, R. (2021). Global soil moisture data derived through machine learning trained with in-situ measurements. *Scientific Data*, *8*(1), 170. https://doi.org/10.1038/s41597-021-00964-1

Ouyang, W., Lawson, K., Feng, D., Ye, L., Zhang, C., & Shen, C. (2021). Continental-scale streamflow modeling of basins with reservoirs: Towards a coherent deep-learning-based strategy. *Journal of Hydrology*, *599*, 126455. https://doi.org/10.1016/j.jhydrol.2021.126455

Pan, M., & Wood, E. F. (2013). Inverse streamflow routing. *Hydrology and Earth System Sciences*, *17*(11), 4577–4588. https://doi.org/10/f5k6nq

Prein, A. F., Rasmussen, R. M., Ikeda, K., Liu, C., Clark, M. P., & Holland, G. J. (2017). The future intensification of hourly precipitation extremes. *Nature Climate Change*, *7*(1), 48–52. https://doi.org/10.1038/nclimate3168

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

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

Regan, R. S., Markstrom, S. L., Hay, L. E., Viger, R. J., Norton, P. A., Driscoll, J. M., & LaFontaine, J. H. (2018). *Description of the National Hydrologic Model for use with the Precipitation-Runoff Modeling System (PRMS)* (No. 6-B9). *Techniques and Methods*. U.S. Geological Survey. https://doi.org/10.3133/tm6B9

Rice, D. (2019, May 28). Mississippi River flood is longest-lasting in over 90 years, since “Great Flood” of 1927. *USA Today*. Retrieved from https://www.usatoday.com/story/news/nation/2019/05/28/mississippi-river-flooding-longest-lasting-since-great-flood-1927/1261049001/

Saha, G. K., Rahmani, F., Shen, C., Li, L., & Cibin, R. (2023). A deep learning-based novel approach to generate continuous daily stream nitrate concentration for nitrate data-sparse watersheds. *Science of The Total Environment*, *878*, 162930. https://doi.org/10.1016/j.scitotenv.2023.162930

Shen, C., & Lawson, K. (2021). Applications of Deep Learning in Hydrology. In *Deep Learning for the Earth Sciences* (pp. 283–297). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119646181.ch19

Shen, C., & Phanikumar, M. S. (2010). A process-based, distributed hydrologic model based on a large-scale method for surface–subsurface coupling. *Advances in Water Resources*, *33*(12), 1524–1541. https://doi.org/10/c4r8k5

Shen, C., Niu, J., & Phanikumar, M. S. (2013). Evaluating controls on coupled hydrologic and vegetation dynamics in a humid continental climate watershed using a subsurface - land surface processes model. *Water Resources Research*, *49*(5), 2552–2572. https://doi.org/10/f5gcrx

Shen, C., Niu, J., & Fang, K. (2014). Quantifying the effects of data integration algorithms on the outcomes of a subsurface–land surface processes model. *Environmental Modelling & Software*, *59*, 146–161. https://doi.org/10/ggj7mp

Shen, C., Riley, W. J., Smithgall, K. M., Melack, J. M., & Fang, K. (2016). The fan of influence of streams and channel feedbacks to simulated land surface water and carbon dynamics. *Water Resources Research*, *52*(2), 880–902. https://doi.org/10/f8gppj

Shen, C., Chen, X., & Laloy, E. (2021). Editorial: Broadening the use of machine learning in hydrology. *Frontiers in Water*, *3*. https://doi.org/10.3389/frwa.2021.681023

Shen, C., Fang, K., Feng, D., & Bindas, T. (2021). mhpi/hydroDL: MHPI-hydroDL [Data set]. Zenodo. https://doi.org/10.5281/zenodo.5015120

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

Sun, A. Y., Jiang, P., Yang, Z.-L., Xie, Y., & Chen, X. (2022). A graph neural network approach to basin-scale river network learning: The role of physics-based connectivity and data fusion. *Hydrology and Earth System Sciences Discussions*. https://doi.org/10.5194/hess-2022-111

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

US Army Corps of Engineers. (2018). National Inventory of Dams (NID) [Data set]. Retrieved from https://nid.sec.usace.army.mil/

USGS ScienceBase-Catalog. (2022). National Elevation Dataset (NED). Retrieved September 13, 2022, from https://www.sciencebase.gov/catalog/item/4fcf8fd4e4b0c7fe80e81504

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

Winsemius, H. C., Aerts, J. C. J. H., van Beek, L. P. H., Bierkens, M. F. P., Bouwman, A., Jongman, B., et al. (2016). Global drivers of future river flood risk. *Nature Climate Change*, *6*(4), 381–385. https://doi.org/10.1038/nclimate2893

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

Xia, Y., Mitchell, K., Ek, M., Sheffield, J., Cosgrove, B., Wood, E., et al. (2009). NLDAS Primary Forcing Data L4 Hourly 0.125 x 0.125 degree V002 (NLDAS_FORA0125_H) [Data set].

Goddard Earth Sciences Data and Information Services Center (GES DISC). https://doi.org/10.5067/6J5LHHOHZHN4

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

Xiang, Z., Yan, J., & Demir, I. (2020). A rainfall-runoff model with LSTM-based sequence-to-sequence learning. *Water Resources Research*, *56*(1), e2019WR025326. https://doi.org/10.1029/2019WR025326

Ye, A., Zhou, Z., You, J., Ma, F., & Duan, Q. (2018). Dynamic Manning's roughness coefficients for hydrological modelling in basins. *Hydrology Research*, *49*(5), 1379–1395. https://doi.org/10.2166/nh.2018.175

Zhi, W., Feng, D., Tsai, W.-P., Sterle, G., Harpold, A., Shen, C., & Li, L. (2021). From hydrometeorology to river water quality: Can a deep learning model predict dissolved oxygen at the continental scale? *Environmental Science & Technology*, *55*(4), 2357–2368. https://doi.org/10.1021/acs.est.0c06783

Zhu, F., Li, X., Qin, J., Yang, K., Cuo, L., Tang, W., & Shen, C. (2021). Integration of multisource data to estimate downward longwave radiation based on deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 1–15. https://doi.org/10.1109/TGRS.2021.3094321

Appendix

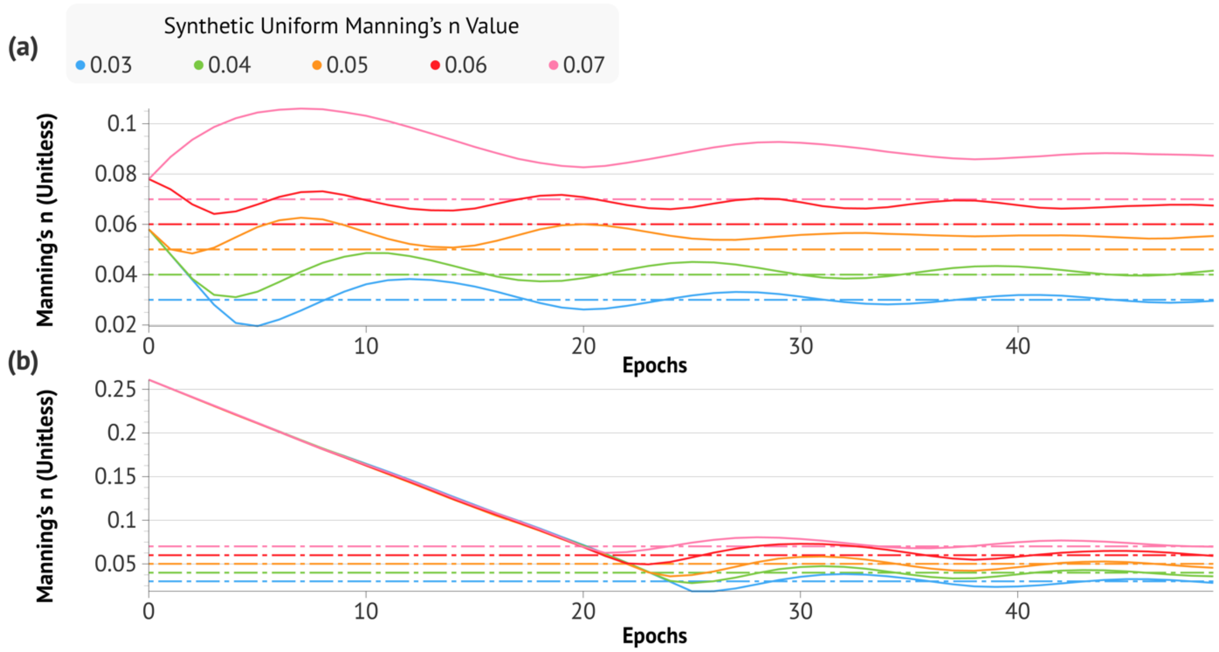

Figure A1: The synthetic parameter recovery of Manning's *n* after each epoch run, with each colored line representing a different recovered value. (a) The initial value of *n* is set to 0.068 (b) the initial value of *n* is set to 0.271

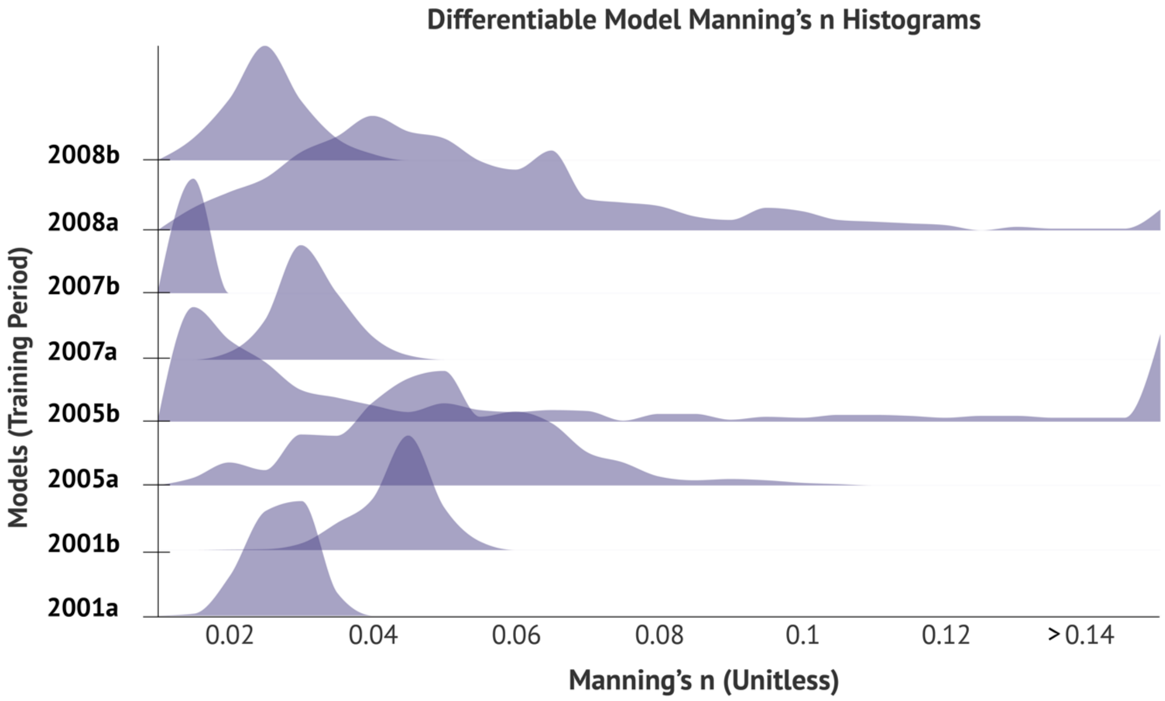

Figure A2: Histograms visualizing the frequency, and variability, of Manning's *n* values for all river reaches (582 total) for all eight GNN models. The lower bound is 0.01, while the upper bound contains all Manning's *n* values >0.14.

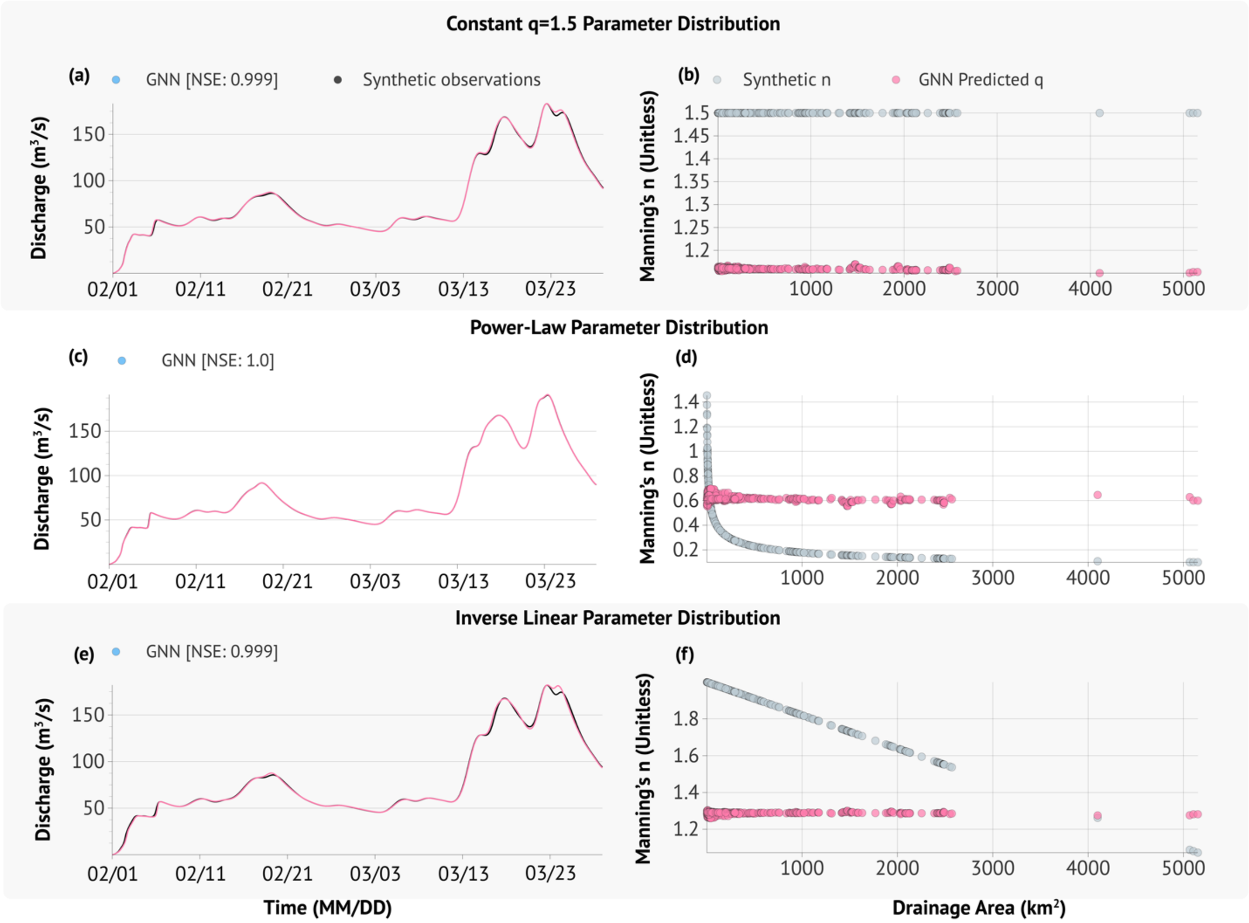

Figure A3: Results from *q* parameter recovery experiments. We tried to recover both constant and distributed parameters, but were unable to ever recover the synthetic truth.

Table A1: The attributes and forcings used by the pre-trained LSTM to predict streamflow. Links to the data can be found below the table

Attribute/Meteorological Forcing	Unit	Dataset	Citation
Mean Elevation	m	SRTMGL1	(Carabajal & Harding, 2006)
Mean Slope	unitless	SRTMGL1	(Carabajal & Harding, 2006)

Basin Area	km^2	SRTMGL1	(Carabajal & Harding, 2006)
Dominant Land Cover	Class	MODIS	(Friedl & Sulla-Menashe, 2019)
Dominant Land Cover Fraction	Percent	MODIS	(Friedl & Sulla-Menashe, 2019)
Forest Fraction	Percent	MODIS	(Friedl & Sulla-Menashe, 2019)
Root Depth (50)	m	MODIS	(Friedl & Sulla-Menashe, 2019)
Soil Depth	m	MODIS	(Friedl & Sulla-Menashe, 2019)
Ksat (0-5)	$\log_{10}$(cm/hr)	POLARIS	(Chaney et al., 2019)
Ksat (5-15)	$\log_{10}$(cm/hr)	POLARIS	(Chaney et al., 2019)
Theta s (0-5)	m^3/m^3	POLARIS	(Chaney et al., 2019)
Theta s (5-15)	m^3/m^3	POLARIS	(Chaney et al., 2019)
Theta r (5-15)	m^3/m^3	POLARIS	(Chaney et al., 2019)
Ksat average (0-15)	$\log_{10}$(cm/hr)	POLARIS	(Chaney et al., 2019)
Ksat e (0-5)	cm/hr	POLARIS	(Chaney et al., 2019)

Ksat e (5-15)	cm/hr	POLARIS	(Chaney et al., 2019)
Ksat average e (0-15)	cm/hr	POLARIS	(Chaney et al., 2019)
Theta average s (0-15)	$e^{m3/m3}$	POLARIS	(Chaney et al., 2019)
Theta average r (0-15)	$e^{m3/m3}$	POLARIS	(Chaney et al., 2019)
Porosity	Percent	GLHYMPS	(Huscroft et al., 2018)
Permeability Permafrost	m^2	GLHYMPS	(Huscroft et al., 2018)
Permeability Permafrost (Raw)	m^2	GLHYMPS	(Huscroft et al., 2018)
Major Number of Dams	Unitless	GAGES-II	(Falcone, 2011)
General Purpose of Dam	Unitless	National Inventory of Dams (NID)	(US Army Corps of Engineers, 2018)
Max of Normal Storage	Acre-ft	National Inventory of Dams (NID)	(US Army Corps of Engineers, 2018)
Standard Deviation of Normal Storage	Unitless	National Inventory of Dams (NID)	(US Army Corps of Engineers, 2018)
Number of dams within river (2009)	Unitless	GAGES-II	(Falcone, 2011)
Normal Storage (2009)	Acre-ft	National Inventory of Dams (NID)	(US Army Corps of Engineers, 2018)
Precipitation hourly total	kg/m^2	NLDAS2	(Xia et al., 2012)
Surface downward longwave radiation	W/m^2	NLDAS2	(Xia et al., 2012)

Surface downward shortwave radiation	W/m^2	NLDAS2	(Xia et al., 2012)
Pressure	Pa	NLDAS2	(Xia et al., 2012)
Air Temperature	K	NLDAS2	(Xia et al., 2012)

SRTMGL1: https://doi.org/10.14358/PERS.72.3.287
MODIS: https://modis.gsfc.nasa.gov/data/dataprod/mod12.php
POLARIS: https://doi.org/10.1029/2018WR022797
GLHYMPS: https://doi.org/10.5683/SP2/DLGXYO
NID: https://nid.usace.army.mil/
NLDAS2: https://ldas.gsfc.nasa.gov/nldas/v2/forcing

Table A2: The constant attributes (c) used by the MLP to predict *n* and *q: n,q = NN(c).*

Attribute	**Unit**
Reach Width	m
Average-Reach Elevation	m
Slope	m/m
Reach Area	km^2
Total Drainage Area	km^2
Reach Length	m
Sinuosity	m/m
Bank Elevation	m

Table A3: The Σ Q\` ($\tau = 9$) NSE scores for all eight training time periods for the most downstream gage. Since Q\` routing is a pure forward simulation using the trained LSTM, we report the NSE values for each period.

	Periods							
	2001a	2001b	2005a	2005b	2007a	2007b	2008a	2008b
NSE	0.5958	0.3534	-0.7868	-0.1687	0.6830	0.0558	-0.4297	0.3792

976