Hydrol. Earth Syst. Sci., 28, 3051–3077, 2024
https://doi.org/10.5194/hess-28-3051-2024

Hydrology and
Earth System
Sciences
Open Access
EGU

When ancient numerical demons meet physics-informed machine learning: adjoint-based gradients for implicit differentiable modeling

Yalan Song[1], **Wouter J. M. Knoben**[2], **Martyn P. Clark**[2], **Dapeng Feng**[1,3], **Kathryn Lawson**[1], **Kamlesh Sawadekar**[1], **and Chaopeng Shen**[1]

[1]Civil and Environmental Engineering, The Pennsylvania State University, University Park, PA, USA
[2]Department of Civil Engineering, Schulich School of Engineering, University of Calgary, Calgary, Alberta, Canada
[3]Department of Earth System Science, Stanford University, Stanford, CA, USA

Correspondence: Yalan Song (yxs275@psu.edu) and Chaopeng Shen (cshen@engr.psu.edu)

Received: 31 October 2023 – Discussion started: 9 November 2023
Revised: 23 March 2024 – Accepted: 6 May 2024 – Published: 15 July 2024

Abstract. Recent advances in differentiable modeling, a genre of physics-informed machine learning that trains neural networks (NNs) together with process-based equations, have shown promise in enhancing hydrological models' accuracy, interpretability, and knowledge-discovery potential. Current differentiable models are efficient for NN-based parameter regionalization, but the simple explicit numerical schemes paired with sequential calculations (operator splitting) can incur numerical errors whose impacts on models' representation power and learned parameters are not clear. Implicit schemes, however, cannot rely on automatic differentiation to calculate gradients due to potential issues of gradient vanishing and memory demand. Here we propose a "discretize-then-optimize" adjoint method to enable differentiable implicit numerical schemes for the first time for large-scale hydrological modeling. The adjoint model demonstrates comprehensively improved performance, with Kling–Gupta efficiency coefficients, peak-flow and low-flow metrics, and evapotranspiration that moderately surpass the already-competitive explicit model. Therefore, the previous sequential-calculation approach had a detrimental impact on the model's ability to represent hydrological dynamics. Furthermore, with a structural update that describes capillary rise, the adjoint model can better describe baseflow in arid regions and also produce low flows that outperform even pure machine learning methods such as long short-term memory networks. The adjoint model rectified some parameter distortions but did not alter spatial parameter distributions, demonstrating the robustness of regionalized parameterization. Despite higher computational expenses and modest improvements, the adjoint model's success removes the barrier for complex implicit schemes to enrich differentiable modeling in hydrology.

1 Background

Accurate hydrological predictions are crucial for effective water resource management around the world under a changing climate (Hannah et al., 2011; Sivapalan et al., 2003). In recent years, deep learning models such as long short-term memory (LSTM) networks have gained traction in hydrology due to their high predictive performance in various applications, including streamflow prediction, soil moisture estimation, and the modeling of stream temperature and dissolved oxygen (Fang et al., 2017; Feng et al., 2020; Kratzert et al., 2019; Ouyang et al., 2021; Rahmani et al., 2021a, b; Zhi et al., 2023). Despite their impressive capabilities, deep learning models are often criticized for their limited interpretability and dependence on extensive observations. Additionally, they are unable to provide outputs for untrained variables (those not trained using observations as targets), e.g., evapotranspiration (ET), water storage, or snow water equivalent, which are of great interest to stakeholders but are not extensively observed.

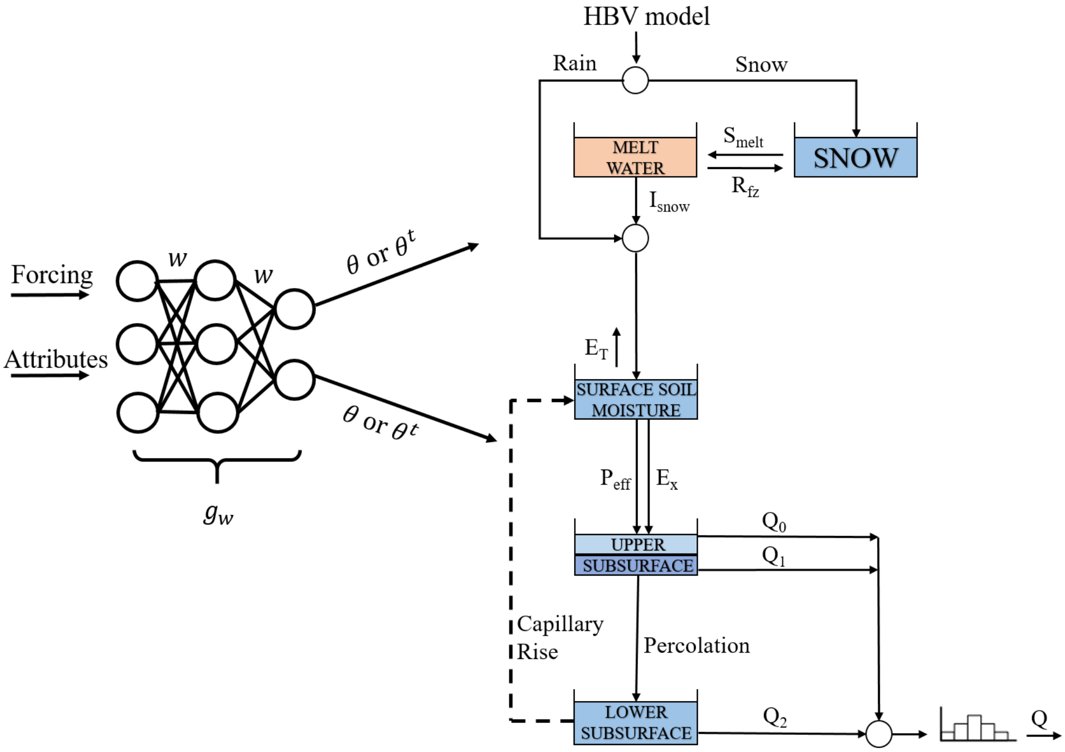


Figure 1. A schematic view of the HBV model. g_w is a parameterization neural network that seeks to capture the prevalent relationship between raw input data and the HBV parameters (w represents the weights of the neural network). θ and θ^t are the static and dynamic HBV parameters, respectively. I_snow is the snowmelt infiltration to soil moisture, R_fz is the refreezing of liquid snow, and s_melt is the snowmelt as water equivalent. ET is the actual evapotranspiration, P_eff is the effective flow to the upper subsurface zone, and E_x is the rainfall excess. Q_0, Q_1, and Q_2 are the near-surface flow, interflow, and baseflow, respectively. Q is the simulated streamflow at the catchment outlet.

In response to these limitations, alternative approaches that merge the process-based understanding of hydrological systems with various genres of physics-informed machine learning techniques have been explored. These approaches include differentiable modeling (DM), interpretable machine learning approaches (Wang et al., 2021), postprocessing (Frame et al., 2021), and embedding trained networks into existing models (Bennett and Nijssen, 2021). Notably, the neural networks must be pretrained using either observations or model simulations as the target. Not many approaches allow interpretability, continued updating of the neural networks, and knowledge discovery at the same time.

Recently, differentiable modeling (DM) (Shen et al., 2023) was proposed as a pathway to train neural networks (NNs) together with physical equations in an "end-to-end" fashion (Fig. 1), where the NNs can provide parameters or unknown relationships for the process-based components (Tsai et al., 2021; Feng et al., 2022, 2023; Aboelyazeed et al., 2023; Bindas et al., 2024). When the model is "differentiable" (explained in the next paragraph), we have a "credit assignment path" (Schmidhuber, 2015) between tunable parameters and the objective function, which enables efficient training of massive amounts of weights on big data based on outputs of the combined system. It also removes the need for direct supervising data for the output of the NN (although such data can be used as additional constraints when available) and enables the discovery of knowledge from data. To make the model differentiable, we prefer translating physical models onto differentiable platforms, which would guarantee the desired sensitivity as the physics are baked into the model, over training an NN as a physical model surrogate. As the models are interpretable, they can be used to provide a full narrative of the physical processes and have the potential to discover scientific knowledge and unrecognized linkages from data. They also extrapolate better in space, especially in data-sparse regions, due to using process-based equations as the backbone of calculations and respecting assumed physical laws like conservation of mass (Feng et al., 2023).

Essentially, differentiable models aim to train coupled NNs (by optimizing their weights, w, in Fig. 1) using gradient descent. In the end-to-end fashion, we must be able to calculate the gradients of model outputs with respect to NN weights along all the steps (physical equations and NN layers) in the model. Models supporting such gradient calculations are called differentiable models (Shen et al., 2023). Gradient descent is the only currently known way to train NNs with massive amounts of weights on big data, and such a computational infrastructure is very efficient, especially when paired with parallel GPU (graphics processing unit) processing and automatic differentiation (AD, explained further below). Modern machine learning platforms are built to support differentiability and NN training using AD, but there are other options as well. For example, while not the focus of DM, we can train NNs as surrogate models for physical models (minding all the complexities with maintaining the accuracy of a surrogate model as well as the fact that we cannot modify the internal functions of a surrogate model) and place them alongside a parameterization network (Tsai et al., 2021). As another example, adjoint state methods have been developed to solve an accompanying equation to produce gradients of an equation-solving step (Chen et al., 2018).

While there have been significant advances in differentiable models, certain numerical approximations have been introduced to facilitate their easy implementation on machine learning platforms utilizing AD. These approximations can result in numerical errors, colloquially termed the "ancient numerical demon" (Clark and Kavetski, 2010; Kavetski and Clark, 2010), with implications for the calibrated parameter values. Such approximations encompass, but are not limited to, ad hoc operator splitting (sequential operations), explicit numerical schemes without error control, and the use of threshold-like functions to avoid negative state variables, all of which can degrade the quality of gradient calculations for parameter optimization. Recently published differentiable hydrological models (Feng et al., 2022, 2023) mostly used these numerical approximations because of their straightforward implementation and a long legacy of usage (Aghakouchak and Habib, 2010; Beck et al., 2020; Bergström, 1976, 1992; Seibert and Vis, 2012). The numerical errors can be compensated for by pushing the calibrated parameters to a different part of the parameter space (Clark and Kavetski, 2010; Kavetski and Clark, 2010). With such compensation, the performance metrics such as the Nash–Sutcliffe model efficiency coefficient (NSE) can be kept high, but the interpretation of the results and the physical significance of the calibrated parameters are obscured. The implications of numerical errors were examined in detail before, where Kavetski and Clark (2010) addressed the issues using implicit schemes. However, the issue of explicit versus implicit solvers has not been examined in the context of a regionalized parameterization scheme, especially a novel differentiable model relying on regionalized parameter learning with NNs, which applies implicit regional constraints. The extent to which numerical schemes can impact regionalized parameter distributions is unclear.

As an underpinning of deep learning, AD decomposes complex calculations into a sequence of elementary arithmetic operations and then applies the chain rule of differentiation to compute the derivative of the output with respect to its input variables. AD often needs to store some intermediate results or instructions, thus consuming memory. AD typically requires very little effort on the modeler's side besides writing the model in a forward mode (no need to provide gradient functions) on a machine learning platform like PyTorch, Tensorflow, JAX, or Julia and is the obvious tool to calculate gradients for explicit models. However, if we want to improve a model's accuracy and stability using implicit solvers that require iterative steps, AD could run into the issue of having high overhead and excessive memory usage. Memory use is a significant issue for GPUs, which are crucial to modern machine learning. Furthermore, tracking the gradients of the many iterative steps with AD may lead to the dreaded vanishing gradient problem (Hochreiter and Schmidhuber, 1997) facing recurrent neural network training, where the gradients become exceedingly small and prevent the NN weights from being effectively updated. These issues, unfortunately, make it challenging for differentiable models to employ implicit solvers that encompass a wealth of powerful and essential tools for solving some equations, e.g., those containing elliptic operators (groundwater equations, Todd and Mays, 2004; Richards' equation, Richards, 1931; shallow water equations, Sadourny, 1975; heat equation, Bergman, 2011) or systems of nonlinear equations (Aboelyazeed et al., 2023).

The adjoint method has been widely used for equation-constrained optimization (Cao et al., 2002) in various fields such as meteorology, oceanography, and geophysics, but only in recent years has it been applied for neural network training with differential equations (Rackauckas et al., 2021). Hydrological modelers have also used the adjoint for data assimilation (Fisher and Andersson, 2001; White et al., 2003; Neupauer and Wilson, 2001; Liu and Gupta, 2007; Castaings et al., 2009; Jay-Allemand et al., 2020; Bandai, 2022). Instead of automatically working through the elementary operations as AD does, the adjoint solves another accompanying equation (derived by the modeler based on the chain rule and the associative property of matrix multiplication) to rapidly produce the gradients of outputs with respect to inputs – more precisely, it computes the vector-Jacobian product. However, the adjoint method has not yet been extensively explored in the context of large-scale, regionalized hydrological simulations (White et al., 2003; Colleoni et al., 2022), which require mini batch processing, high data throughput, and a long time for integration. It is unclear whether the adjoint method is applicable in this scenario.

Adjoint methods can be defined at different levels. Simply put, the adjoint state method is defined at the differential equation level (called "optimize-then-discretize"), involving the solution of a separate differential equation for the adjoint (Chen et al., 2018). However, we can also define it at lower functional levels, e.g., solving an adjoint equation for a specific operator inside a discretized numerical model (called "discretize-then-optimize") (Onken and Ruthotto, 2020). The latter is more naturally implemented along with the numerical algorithms to solve the forward problem.

In both machine learning and process-based modeling, an important point for consideration is whether the basic architecture has the expressive power to represent the phenomenon of interest. Deep networks can approximate extremely complex functions, due to the enormous number of weights that can be trained, along with their generic architecture (Hornik et al., 1989). For process-based models (or hybrid differentiable models), structural deficiencies may lead to problematic behaviors that cannot be remediated (even with the help of highly flexible NNs for parameterization). It is unclear whether the difference between implicit and explicit solvers can lead to differences in representation and whether differentiable models can help us identify structural deficiencies.

In this work, we proposed the application of the discretize-then-optimize adjoint method to implement implicit numerical schemes in differentiable hydrological models (referred to as implicit adjoint-based models or "adjoint models" for brevity, also denoted as δHBV.adj). We then compared them to the existing differentiable models with explicit Euler time stepping and sequential operations (referred to as explicit sequential models or "sequential models" for brevity and denoted as δHBV). We investigated the impacts of these methods on hydrological model performance and parameter distributions. Furthermore, we examined the potential for these adjoint-based methods to enhance the performance of differentiable models, bringing them closer to or surpassing the performance of state-of-the-art LSTM models. We sought answers for the following questions.

1. Can we support implicit numerical solvers in large-scale differentiable hydrological modeling, and what are their implications for performance and computational efficiency?
2. Do implicit and sequential models have different representation power? That is, does the sequential-calculation approach result in errors that prevent it from accurately representing certain aspects of hydrological dynamics?
3. Do we get very different parameter distributions with the implicit (adjoint) model than with the sequential model at the regional and local scales?

The full version name of the adjoint model is δHBV.adj-CAMELS-hydroDL, where "δ" indicates differentiable modeling, "adj" represents the adjoint, "CAMELS" represents the training dataset, and "hydroDL" represents the software implementation.

2 Data and methods

As a high-level summary, the differentiable model couples an LSTM network to a conceptual hydrological model, Hydrologiska Byråns Vattenbalansavdelning (HBV), and trains them together in an end-to-end fashion (from the input of LSTM to the output of HBV) on daily discharge data from 671 basins in the conterminous United States (CONUS). The LSTM, which provides static or dynamic parameters for HBV, is not trained directly with pre-calibrated parameters but is jointly trained with HBV using the discharge observations. This joint training, where the LSTM weights are updated, is supported by either AD for the explicit HBV model or the adjoint method for the implicit HBV model. The conceptual frameworks of the two models are similar, and the differences between AD and the adjoint only pertain to how the gradients are obtained during backpropagation through the solving of HBV's equations. However, due to gradient vanishing and memory issues arising from the required numerical iterations, the implicit model simply cannot be supported by AD for backpropagation. In addition to the implementation of the implicit scheme, this work also evaluated a structural change (incorporating capillary rise) to HBV, based on insights acquired during joint training. We compare the explicit (sequential), implicit, improved explicit (sequential), and improved implicit differentiable models and a direct LSTM simulation of streamflow in terms of various streamflow metrics, and the differentiable models are also benchmarked in terms of ET and baseflow fraction simulations. In the following, we describe the different parts of the framework and discuss the differences between AD and the adjoint method in more detail.

2.1 Datasets

We utilized the Catchment Attributes and Meteorology for Large-sample Studies (CAMELS) dataset (Addor et al., 2017; Newman et al., 2014) for this study. This dataset comprises basin-averaged hydrometeorological time series, catchment attributes, and streamflow observations from the United States Geological Survey (USGS) for 671 catchments across the CONUS. The majority of its daily streamflow observations span from 1980 to 2014. For our study, the meteorological forcing data were sourced from Daily Surface Weather Data on a 1 km Grid for North America (Daymet) Version 4 (Thornton et al., 2020). From the CAMELS dataset, we incorporated catchment attributes such as topography, climate patterns, land cover, soil, and geological characteristics as inputs to our models (Table A1).

We compared our model simulations with the streamflow observations as well as the streamflow simulations from a traditional process-based model, SAC-SMA (Sacramento Soil Moisture Accounting), which was calibrated by the National Weather Service. The simulation results of SAC-SMA are provided in CAMELS. To assess the accuracy of predicted intermediate variables, we also employed the Baseflow Index (BFI) from the CAMELS dataset and ET data from the MOD16A2 dataset (Running et al., 2017). The BFI is obtained by applying Lyne and Hollick filters with warm-up periods to streamflow hydrographs (Ladson et al., 2013). The MOD16A2 Version 6 Evapotranspiration/Latent Heat Flux product provides 8 d composites at a 500 m pixel resolution and is aggregated to the average values at the basin levels. The algorithm used for the MOD16A2 dataset relies on the Penman–Monteith equation logic, incorporating daily meteorological reanalysis data and Moderate Resolution Imaging Spectroradiometer (MODIS) remote sensing inputs like vegetation dynamics, albedo, and land cover (Monteith, 1965; Mu et al., 2011; Running et al., 2017).

2.2 Models

2.2.1 Differentiable, learnable, regionalized process-based model

This study utilized the differentiable regionalized process-based model framework presented in Feng et al. (2022), which employs the HBV model as its backbone and utilizes LSTM for parameter regionalization. The motivation for making the model differentiable is so that it can train connected NNs in an end-to-end manner to learn robust and complex relationships from big data, which can provide indirect supervision to the NNs. As stated above, the model was already made programmatically differentiable because it was implemented on the PyTorch ML platform, so gradient information can backpropagate through it. However, it was previously only integrated in time using the explicit Euler method with no error control. The HBV model is a conceptual model that uses a set of linked storage components representing processes like snow accumulation and melt, soil moisture dynamics, and river routing to forecast river discharge. To summarize the NN–HBV coupling succinctly, the differentiable model based on HBV can be written as

$$\theta = g_{\mathrm{w}}(x, A), \tag{1}$$

where θ represents HBV physical parameters; A contains 35 static attributes such as topography, climate, soil texture, land cover, and geology (Table A1); x represents the meteorological forcings; and g_{w} is a parameterization neural network that seeks to capture the prevalent relationship between the input data and the HBV parameters (w represents the weights of the neural network). θ can be formulated as being either static-in-time or time-dependent, where new values are obtained for every day of the simulation. More details about the data can be found in Feng et al. (2022). The HBV forward simulation is succinctly written as

$$Q = \mathrm{HBV}(x, \theta), \tag{2}$$

where Q is the simulated streamflow. The HBV model utilizes three primary forcing variables: precipitation (P), temperature (T), and potential evapotranspiration (E_{P}). The Hargreaves (1994) method, which considers mean, maximum, and minimum temperatures along with latitudes, is employed to estimate E_{P}, representing the total evaporative demand. The same forcings, x (including P, T, and E_{p}), were used in g_{w}. It should be noted that HBV only serves as an example, and other hydrological models (Knoben et al., 2019) can be similarly employed.

2.2.2 Hydrologiska Byråns Vattenbalansavdelning model

The HBV model employs a framework that includes five water storages and associated fluxes to encapsulate the primary hydrological processes within a catchment. It can simulate hydrological variables, including soil moisture, groundwater storage, evapotranspiration, quick flow, baseflow, and streamflow. It consists of four main modules to classify all storages and fluxes as shown in Fig. 1.

Snow accumulation and melt: this module uses a temperature-index method to distinguish between rainfall and snowfall and to simulate the snow accumulation and melt processes:

$$\frac{\mathrm{d}S_{\mathrm{p}}}{\mathrm{d}t} = P_{\mathrm{s}} + R_{\mathrm{fz}} - S_{\mathrm{smelt}}, \tag{3}$$

$$P_{\mathrm{s}} = P \text{ if } T < \theta_{\mathrm{TT}}, \text{ otherwise } 0, \tag{4}$$

$$R_{\mathrm{fz}} = (\theta_{\mathrm{TT}} - T)\,\theta_{\mathrm{DD}}\theta_{\mathrm{rfz}}, \tag{5}$$

$$s_{\mathrm{melt}} = (T - \theta_{\mathrm{TT}})\,\theta_{\mathrm{DD}}, \tag{6}$$

where t is time; S_{p} is the current snow storage [mm]; P_{s} is the precipitation as snow [mm d^{-1}]; R_{fz} is the refreezing of liquid snow [mm d^{-1}]; S_{smelt} is the snowmelt as water equivalent [mm d^{-1}]; and θ_{TT}, θ_{DD}, and θ_{rfz} are the threshold temperature for snowfall [°C], degree-day factor [mm °C^{-1} d^{-1}], and refreezing coefficient [–], respectively.

$$\frac{\mathrm{d}S_{\mathrm{liq}}}{\mathrm{d}t} = S_{\mathrm{smelt}} - R_{\mathrm{fz}} - I_{\mathrm{snow}}, \tag{7}$$

$$I_{\mathrm{snow}} = s_{\mathrm{liq}} - \theta_{\mathrm{CWH}} S_{\mathrm{p}}, \tag{8}$$

where I_{snow} is the snowmelt infiltration to soil moisture [mm d^{-1}]; S_{liq} is the liquid water content in the snowpack [mm]; and θ_{CWH} is the water-holding capacity as a fraction of the current snowpack [–].

Soil moisture and evapotranspiration: the model features a simple soil moisture accounting scheme where precipitation and snowmelt infiltration can contribute to either evapotranspiration or runoff. Potential evapotranspiration, typically calculated externally (e.g., using the Hargreaves method; Hargreaves, 1994), limits the actual evapotranspiration from the soil storage.

$$\frac{\mathrm{d}S_{\mathrm{S}}}{\mathrm{d}t} = I_{\mathrm{snow}} + P_{\mathrm{r}} - P_{\mathrm{eff}} - E_x - E_{\mathrm{T}}\,(+C_{\mathrm{r}}), \tag{9}$$

$$P_{\mathrm{r}} = P \text{ if } T > \theta_{\mathrm{TT}}, \text{ otherwise } 0, \tag{10}$$

$$P_{\mathrm{eff}} = \min\left(\left(\frac{S_{\mathrm{S}}}{\theta_{\mathrm{FC}}}\right)^{\beta}, 1\right)(P_{\mathrm{r}} + I_{\mathrm{snow}}), \tag{11}$$

$$E_x = (S_{\mathrm{S}} - \theta_{\mathrm{FC}})\,/\mathrm{d}t, \tag{12}$$

$$E_\mathrm{T}=\min\left(\left(\frac{S_\mathrm{S}}{\theta_\mathrm{FC}\theta_\mathrm{LP}}\right)^{\gamma},1\right)E_\mathrm{P},\tag{13}$$

$$C_\mathrm{r}=\theta_\mathrm{C}\cdot S_\mathrm{LZ}\cdot\left(1-\frac{S_\mathrm{S}}{\theta_\mathrm{NDC}\cdot\theta_\mathrm{FC}}\right),\tag{14}$$

where S_S is the current storage in soil moisture [mm]; S_LZ is the current storage in the lower subsurface zone [mm d^{-1}]; P_r is the precipitation as rain [mm d^{-1}]; P_eff is the effective flow to the upper subsurface zone [mm d^{-1}]; E_x is the rainfall excess [mm d^{-1}]; E_T is the actual evapotranspiration [mm d^{-1}]; C_r is the capillary rise from the lower subsurface zone; θ_C is a time parameter [d^{-1}]; θ_FC is the maximum soil moisture (field capacity) [mm]; θ_NDC is the fraction ratio of the field capacity [–] and is set to a constant value of 1 in the discrete model to prevent gradient explosion, while in the adjoint model it is treated as a static parameter to be learned; θ_LP is the vegetation wilting point [–]; β is a parameter influencing the shape of the soil moisture function [–]; and γ is a parameter influencing the shape of the evapotranspiration function [–].

Runoff generation: runoff in the HBV model is represented by three components – two quick flows (near-surface flow and interflow) and delayed runoff (or baseflow).

$$\frac{\mathrm{d}S_\mathrm{UZ}}{\mathrm{d}t}=P_\mathrm{eff}+E_x-\mathrm{Perc}-Q_0-Q_1,\tag{15}$$

$$\mathrm{Perc}=\min\left(\theta_\mathrm{perc}S_\mathrm{UZ}/\mathrm{d}t\right),\tag{16}$$

$$Q_0=\theta_{K_0}\left(S_\mathrm{UZ}-\theta_\mathrm{UZL}\right),\tag{17}$$

$$Q_1=\theta_{K_1}S_\mathrm{UZ},\tag{18}$$

$$\frac{\mathrm{d}S_\mathrm{LZ}}{\mathrm{d}t}=\mathrm{Perc}-Q_2\left(-C_\mathrm{r}\right),\tag{19}$$

$$Q_2=\theta_{K_0}S_\mathrm{LZ},\tag{20}$$

where S_UZ is the current storage in the upper subsurface zone [mm]; Perc is the percolation to the lower subsurface zone [mm d^{-1}]; Q_0, Q_1, and Q_2 are the near-surface flow [mm d^{-1}], interflow [mm d^{-1}], and baseflow [mm d^{-1}], respectively; θ_perc is the percolation flow rate [mm d^{-1}]; and θ_{K_0}, θ_{K_1}, and θ_{K_2} are the recession coefficients [d^{-1}].

Basin-scale routing: we employ a gamma function to simulate the flow routing through rivers and lakes within the catchment, leading to the simulated discharge at the catchment outlet.

$$Q(t)=\int_0^t\xi(s)Q'(t-s)\mathrm{d}s,\tag{21}$$

$$\xi(s)=\frac{1}{\Gamma\left(\theta_a\right)\theta_b^{\theta_a}}s^{\theta_a-1}e^{-\frac{1}{\theta_b}},\tag{22}$$

where $Q'=Q_0+Q_1+Q_2$; Q is the simulated streamflow at the catchment outlet; and θ_a [–] and θ_b [–] are two routing parameters.

In this work, we investigated two distinct HBV structures. The first structure replicates the structure employed in Feng et al. (2022), adapted from the HBV structure used in Beck et al. (2020). A primary limitation of this structure is its inability to represent the depletion of storages and the occurrence of zero flow. In scenarios where precipitation events are minimal and soil moisture is not entirely depleted, there always exists a recharge flow directed into the groundwater compartments, consequently producing a baseflow. To alleviate this limitation, according to previous experiences (Knoben et al., 2019), various strategies can be employed: (1) applying a threshold-based function to Q_1 and Q_2, (2) constraining the effective rainfall and excess volume, (3) adapting the ET functions, (4) introducing a sink flux into the lower subsurface zone to directly remove water, or (5) incorporating a capillary flux to redistribute water among the soil zones. Among these strategies, we chose to incorporate a capillary flux from the lower subsurface zone to the surface soil, as illustrated by the dashed arrow in Fig. 1, in order to avoid introducing threshold-like functions, which would complicate gradient calculations. Notably, this is not the typical capillary flux from the upper subsurface zone. Similarly, the reciprocal flux between different soil zones is profoundly influenced by the operation order, making it more susceptible to numerical errors in the sequential models. The capillary flux can increase the evapotranspiration, especially when surface soil moisture is at a diminished level, thereby moderating the baseflow. This flux (Eq. 14) is mathematically accounted for by additional terms in the brackets of Eqs. (9) and (19), whereby it is subtracted from the lower subsurface soil zone and added to the upper soil zone. The sequential and adjoint models incorporating the capillary flux are referred to as "the sequential improved model (δHBV improved model)" and "the adjoint improved model (δHBV.adj improved model)", respectively.

2.2.3 Long short-term memory network

LSTM is a recurrent neural network (RNN) designed for identifying patterns in long time series. While traditional RNNs face challenges like vanishing or exploding gradients (Hochreiter and Schmidhuber, 1997), LSTM alleviates these issues using a distinct cell architecture with input, forget, and output gates. These gates modulate information flow, rendering LSTM especially effective for tasks like time series forecasting and sequence-to-sequence modeling. Given LSTM's proficiency in capturing temporal dynamics, it serves two primary functions in this study: direct streamflow prediction and regionalized parameterization.

Direct streamflow prediction using LSTM:

$$Q=\mathrm{LSTM}(x,A,w).\tag{23}$$

This model serves as a benchmark. The meteorological forcings, x, used in the pure LSTM include precipitation, solar radiation, maximum and minimum temperature, and vapor pressure. Attributes, A, include topography, climate, soil texture, land cover, and geology. w are the LSTM weights to be trained for streamflow prediction. More details about the pure LSTM streamflow model can be found in Kratzert et al. (2019), and it is referred to here as the LSTM model. We used the code from Kratzert et al. (2019) and ran the model in the same training and testing periods as the HBV models. Although we have a separate implementation that generated a similar performance (Feng et al., 2020), here we include just one LSTM model from a third party for cross-comparability. While LSTM offers exceptional accuracy in streamflow prediction, its application in hydrological modeling presents interpretability challenges, and it does not produce intermediate physical states or fluxes.

Regionalized parameterization using LSTM: LSTM can also serve as a parameter learning function, referred to as g_w in the DM framework:

$$\theta \text{ or } \theta^t = \mathrm{LSTM}(x, A, w), \tag{24}$$

where the parameters learned can be either static (θ) or time-dynamic (θ^t). The forcings, x, only include the precipitation, temperature, and potential evapotranspiration used in HBV (same as Feng et al., 2022). A includes the same 35 static attributes used in the pure LSTM model and listed in Table A1. w represents the LSTM weights to be trained for HBV parameter estimation.

When employed for regionalized parameterization (in this context, this means that all available training sites are employed to train one network), LSTM establishes a correlation between input data and HBV parameters. By learning from a dataset with 671 basins, it tries to learn the implications of basin characteristics, making it applicable to ungauged basins. The HBV parameters used in Eqs. (3)–(22) can be treated as static (θ) or dynamic parameters (θ^t). When treated as static, the same parameter value is used throughout the HBV simulation, whereas dynamic parameterization (DP) provides a time series of parameters that differ for each basin and each day. In Feng et al. (2022), the DP approach was adopted for parameter γ and β. In the adjoint model, besides γ and β, the field capacity θ_FC is also treated as dynamic to enhance the adaptability of the model.

2.2.4 Backpropagation with a coupled NN and a process-based model

Within the framework of the differentiable model, the HBV model's parameterization is achieved by optimizing the weights, w_i, of the LSTM. This process learns the relationship between the input big data and the optimal parameterization using gradient descent:

$$w^{n+1} = w^n - \alpha \frac{\mathrm{d}L}{\mathrm{d}w^n}, \tag{25}$$

where α is the learning rate (a model hyperparameter) and can be updated automatically by the optimizer, AdaDelta (Zeiler, 2012); L is the loss function that evaluates the discrepancy between the simulated and observed streamflow based on the root-mean-square error (RMSE), conducted on a mini batch of basins during the training process; and n is the iteration number.

The gradient $\frac{\mathrm{d}L}{\mathrm{d}w}$ can be decomposed into multiple terms using the chain rule:

$$\frac{\mathrm{d}L}{\mathrm{d}w} = \frac{\partial L}{\partial w} + \frac{\partial L}{\partial \theta}\frac{\mathrm{d}\theta}{\mathrm{d}w} = \frac{\partial L}{\partial w} + \left(\frac{\partial L}{\partial Q}\frac{\partial Q}{\partial S}\frac{\mathrm{d}S}{\mathrm{d}\theta}\right)\frac{\mathrm{d}\theta}{\mathrm{d}w}, \tag{26}$$

where $\frac{\partial L}{\partial w}$ represents the gradient of regularization terms applied to the weights in the loss function and $\frac{\partial L}{\partial \theta}$ represents the gradient of the loss function with respect to the HBV parameters (θ), which encompasses the backpropagation steps through the loss function associated with the streamflow $\left(\frac{\partial L}{\partial Q}\right)$ and HBV functions $\left(\frac{\partial Q}{\partial S}\frac{\mathrm{d}S}{\mathrm{d}\theta}\right)$. $\frac{\mathrm{d}\theta}{\mathrm{d}w}$ represents the gradient of the HBV parameters with respect to the LSTM weights. During backpropagation, we automatically obtain the gradient $\frac{\mathrm{d}L}{\mathrm{d}w}$ as the program tracks through each function and resolves the gradients from left to right of the chain rule terms in Eq. (26).

2.2.5 Adjoint-based implicit scheme

The HBV model, like many hydrological models, relies on a set of ordinary differential equations (ODEs) to simulate processes. Following many previous hydrological modeling studies, Feng et al. (2022) solved the HBV model by describing each process in a sequential manner with a daily time step (sequential model). They managed the fluxes by sequentially adding to or subtracting from the water storages, limiting the depletion or saturation of storages with threshold functions and ensuring that the storages were updated following each individual process. It remains unclear how the explicit and sequential approach to solving ODEs influences the parameter estimation and internal fluxes, subsequently altering the prediction of streamflow.

To enhance numerical accuracy, this study utilized an adjoint-based implicit numerical scheme to solve the ODEs simultaneously and implicitly. The gradient components, $\frac{\partial L}{\partial Q}$, $\frac{\partial Q}{\partial S}$, and $\frac{\mathrm{d}\theta}{\mathrm{d}w}$ in Eq. (26), can be easily handled by AD. The other gradient component, $\frac{\mathrm{d}S}{\mathrm{d}\theta}$, is challenging for an implicit solver due to the numerous iterations and matrix solving required (Eq. 2).

Consistent with the approach adopted in MARRMoT (Knoben et al., 2019), in one time step, the time derivatives of all ODEs are discretized using a first-order backward Euler implicit scheme:

$$\frac{S^t - S^{t-1}}{\Delta t} = f\left(S^t, \theta^t\right). \tag{27}$$

This can be reformulated as a nonlinear equation:

$$F\left(S^t, \theta^t\right) = f\left(S^t, \theta^t\right) - \frac{S^t - S^{t-1}}{\Delta t} = 0. \tag{28}$$

The Newton–Raphson method is used to solve Eq. (28):

$$S^{t,n} = S^{t,n-1} - \frac{F\left(S^{t,n-1}, \theta^{t,n-1}\right)}{F_{\mathrm{S}}'\left(S^{t,n-1}, \theta^{t,n-1}\right)}, \tag{29}$$

where n is the iteration number and S simply refers to a generic storage. Typically, Eq. (29) is computed for many iterations until convergence, but the number of iterations poses a challenge to AD (due to gradient vanishing) and GPU memory as discussed above. To avoid applying AD throughout the iterations, we differentiate $F(S, \theta) = 0$ with respect to θ:

$$\frac{\partial F}{\partial \theta} + \frac{\partial F}{\partial S}\frac{\mathrm{d}S}{\mathrm{d}\theta} = 0. \tag{30}$$

Thus, we obtain the following equation:

$$\frac{\mathrm{d}S}{\mathrm{d}\theta} = -\left(\frac{\partial F}{\partial S}\right)^{-1}\frac{\partial F}{\partial \theta}. \tag{31}$$

Substituting Eq. (31) into Eq. (26), the gradient of weights of LSTM becomes

$$\frac{\mathrm{d}L}{\mathrm{d}w} = \frac{\partial L}{\partial w} + \frac{\mathrm{d}L}{\mathrm{d}\theta}\frac{\mathrm{d}\theta}{\mathrm{d}w} = \frac{\partial L}{\partial w} - \left(\frac{\partial L}{\partial S}\left(\frac{\partial F}{\partial S}\right)^{-1}\frac{\partial F}{\partial \theta}\right)\frac{\mathrm{d}\theta}{\mathrm{d}w}. \tag{32}$$

We seek to solve for the adjoint, λ, which satisfies

$$\left(\frac{\partial F}{\partial S}\right)^T \lambda = -\left(\frac{\partial L}{\partial S}\right)^T. \tag{33}$$

Here the adjoint is a so-called "vector-Jacobian product", where $\left(\frac{\partial L}{\partial S}\right)^T$ is the vector and $\left(\left(\frac{\partial F}{\partial S}\right)^T\right)^{-1}$ is the Jacobian matrix.

Upon obtaining the adjoint λ, we substitute it into Eq. (32):

$$\frac{\mathrm{d}L}{\mathrm{d}w} = \frac{\partial L}{\partial w} + \frac{\mathrm{d}L}{\mathrm{d}\theta}\frac{\mathrm{d}\theta}{\mathrm{d}w} = \frac{\partial L}{\partial w} + \left(\lambda^T \frac{\partial F}{\partial \theta}\right)\frac{\mathrm{d}\theta}{\mathrm{d}w}. \tag{34}$$

While solving for the adjoint requires solving a matrix, the adjoint method bypasses the need for direct AD through all Newton iterations. Only after the Newton iteration converges and the solution is obtained do we need to compute the vector and the Jacobian and solve the Jacobian matrix, thus greatly reducing the amount of information that requires AD. Furthermore, in our implementation, the adjoints for all the basins in the mini batch are computed in parallel to permit rapid training on the GPU. The forward processes and the backward processes via backpropagation are summarized in Fig. 2. In the backward mode, we only need to customize the backward function of the Newton solver and pass $\frac{\mathrm{d}L}{\mathrm{d}\theta}$ for backpropagation to the subsequent steps. The other parts are automatically supported by PyTorch with AD.

2.2.6 Metrics, model training, and hyperparameters

The training phase employed data spanning 15 years, from 1 October 1980 to 30 September 1995, while the performance evaluation was conducted on data spanning another 15 years, from 1 October 1995 to 30 September 2010. In all the cases, one neural network was trained on all the training basins with all the training data. The hyperparameters of the LSTM unit were inherited from Feng et al. (2022). A hidden state of 256, a mini batch size of 100, and a time series length of 365 d were used to train the models. The model was trained to minimize an objective function (loss function) based on the RMSE across all basins in a mini batch:

$$\mathrm{Loss} = (1.0 - \alpha_l)\sqrt{\frac{\sum_{b=1}^{B}\sum_{t=1}^{T}(Q - Q^*)^2}{B \cdot T}} + \alpha_l\sqrt{\frac{\sum_{b=1}^{B}\sum_{t=1}^{T}\left(\hat{Q} - \hat{Q}^*\right)^2}{B \cdot T}}, \tag{35}$$

$$\hat{Q} = \log_{10}\left(\sqrt{Q + \epsilon} + 0.1\right), \tag{36}$$

where B is the number of basins (mini batch size), T is the number of days involved in the training (time series length), and $\hat{Q}$ is the log-transformed streamflow (transformation done to better represent the low flows in the training data). ϵ is a small value (1×10^{-6}) to stabilize the gradient calculation. α_l is a weight parameter to balance the model's performance between high flow and low flow, where a large value of α_l intends to improve the low-flow performance. Here we set α_l to 0.25, which was manually tuned in Feng et al. (2022).

To evaluate model performance, the NSE (Nash and Sutcliffe, 1970), the Kling–Gupta model efficiency coefficient (KGE; Gupta et al., 2009), and the low-flow- and peak-flow-related hydrological signatures were computed for streamflow as well as for other hydrological variables such as ET and baseflow. The metrics used to evaluate all model performances were the following.

- NSE: the NSE metric was derived from the ratio of the error variance of the modeled time series to the variance of the observed time series, with a value of 1 indicating a perfect model and 0 indicating a performance equivalent to using the long-term mean value as the prediction.
- KGE: the KGE metric considers correlation, bias, and flow variability error, with a perfect simulation having a value of 1.
- Low-flow RMSE: the low-flow RMSE represents the RMSE of the bottom 30 % of the streamflow range.
- Peak-flow RMSE: the peak-flow RMSE represents the RMSE of the top 2 % of the streamflow range.

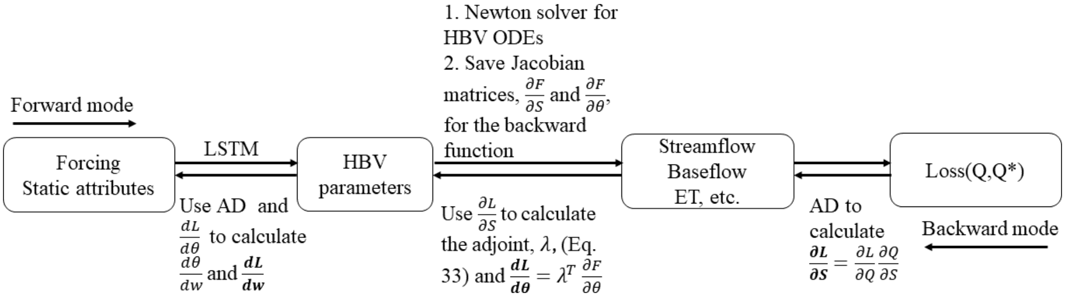

1. Newton solver for HBV ODEs
2. Save Jacobian matrices, $\frac{\partial F}{\partial s}$ and $\frac{\partial F}{\partial \theta}$, for the backward function
Forward mode
Forcing
Static attributes
LSTM
HBV parameters
Streamflow
Baseflow
ET, etc.
Loss(Q,Q*)
Use AD and $\frac{dL}{d\theta}$ to calculate $\frac{d\theta}{dw}$ and $\boldsymbol{\frac{dL}{dw}}$
Use $\frac{\partial L}{\partial s}$ to calculate the adjoint, λ, (Eq. 33) and $\boldsymbol{\frac{dL}{d\theta}} = \lambda^T \frac{\partial F}{\partial \theta}$
AD to calculate $\boldsymbol{\frac{\partial L}{\partial s}} = \frac{\partial L}{\partial Q}\frac{\partial Q}{\partial s}$
Backward mode

Figure 2. Schematic view of the forward code for streamflow prediction and backward code for gradient backpropagation. In the backpropagation process, the steps utilizing automatic differentiation (AD) can be automatically executed by PyTorch. We only need to address the adjoint calculation and pass $\frac{dL}{d\theta}$ for backpropagation to the subsequent step. The bolded derivatives represent the outputs of each step in the backward mode.

- Absolute FLV: the percent of the absolute bias of the bottom 30 % ("low") flow range, i.e., the sum of the absolute bias of the low flow divided by the sum of low-flow values.
- Absolute FHV: the percent of the absolute bias of the top 2 % ("peak") flow range, i.e., the sum of the absolute bias of the peak flow divided by the sum of peak-flow values.
- Baseflow Index spatial correlation: the correlation between the simulated BFI (Q_2/Q) and the BFI from the CAMELS derived from Ladson et al. (2013) across all basins in a spatial context.
- Temporal ET simulation NSE: NSE of the ET time series from the models compared against ET data from the MODIS satellite mission.

3 Results and discussion

In this section, we first examine the overall performance of the adjoint model in comparison with the sequential model and direct LSTM simulation. Then, we examine the impact of a structural change (adding capillary rise to improve baseflow performance) on the sequential and adjoint models. Finally, we examine how using an explicit sequential solution or implicit solution impacts the spatial distribution of parameters produced by the regionalized parameterization network.

3.1 Adjoint model

Before making any structural changes, the adjoint model already demonstrated a highly competitive streamflow prediction performance overall – its KGE, high-flow, and low-flow metrics are all modestly better than those of the sequential model (Table 1). For KGE, the adjoint model (0.75) was higher than the sequential model (0.73) but lower than LSTM (0.77). In terms of peak-flow RMSE (lower is better), the adjoint model scored 2.47 mm d^{-1}, lower (and thus better) than the 2.56 mm d^{-1} scored by both the sequential model and LSTM, which was in turn noticeably lower than SAC-SMA's 3.19 mm d^{-1}. In terms of low-flow RMSE, the adjoint model's performance (0.048 mm d^{-1}) was lower than that of the sequential model (0.074 mm d^{-1}) and LSTM (0.055 mm d^{-1}).

Surprisingly, the adjoint model (δHBV.adj) even slightly surpassed the LSTM in peak-flow accuracy, reducing the median high-flow RMSE by 0.1 mm d^{-1} and the median FHV by 0.2 %. While this difference is small, we remind the readers that the metrics are extremely difficult to improve at this level, and we have not noticed better high-flow metrics elsewhere for this benchmark. The advance may be attributable to δHBV.adj's mass balance preservation (achieved by avoiding the thresholds for negative state values) and reduction of numerical errors, which forces the model to more accurately represent extreme values. It is worthwhile mentioning that, while δHBV.adj is competitive with LSTM at the median on CAMELS basins, as illustrated earlier, it was outperformed by LSTM for the low-KGE basins (lower part in Fig. 3, where NSE or KGE is in [0, 0.5]). One possibility is that rainfall data have significant predictable bias and errors for these basins (see more discussion in Sect. 4). On a side note, this comparison also highlights that a single metric like the median KGE or NSE may not tell the full story.

The adjoint model improved predictions for other hydrological variables not employed as training targets (and thus cannot be directly simulated by LSTM), including baseflow and ET. The spatial correlation of the simulated BFI (Q_2/Q) was enhanced to 0.83, in comparison with the sequential model's correlation of 0.76 (Table 2). This improvement is consistent with the adjoint model's superior ability to capture low flows. The correlation of simulated ET with the MODIS product was increased from 0.59 with the sequential model to 0.61 with the implicit (adjoint) model. Both the BFI and the MODIS products are only alternative estimates, but they are derived using different methods and MODIS utilizes independent information, and thus a better agreement is nonetheless an indication of better model behavior.

Table 1. Summary of the statistical streamflow metrics for all the models in the testing period using Daymet meteorological forcing data. We used the code from Kratzert et al. (2019) for cross-group comparability. Bold values denote the best metrics in each column.

Model	Median NSE	Median KGE	Median absolute (non-absolute) FLV (%)	Median absolute (non-absolute) FHV (%)	Median low-flow RMSE (mm d^{-1})	Median peak-flow RMSE (mm d^{-1})	Dynamic parameters
LSTM	**0.73**	**0.77**	40.59 (29.70)	13.46 (−4.19)	0.055	2.56	
SAC-SMA	0.66	0.73	59.40 (46.96)	17.55 (−9.79)	0.081	3.19	–
δHBV	**0.73**	0.73	56.53 (50.93)	15.29 (−8.89)	0.074	2.56	γ, β
δHBV.adj	0.72	0.75	43.29 (37.61)	**13.25** (**–4.33**)	0.048	**2.47**	$\gamma, \beta, \theta_{FC}$
δHBV improved	**0.73**	0.75	**35.69** (**21.09**)	15.45 (−10.61)	0.049	2.72	γ, β
δHBV.adj improved	**0.73**	0.76	37.63 (28.63)	14.36 (−6.04)	**0.047**	2.59	$\gamma, \beta, \theta_{FC}$

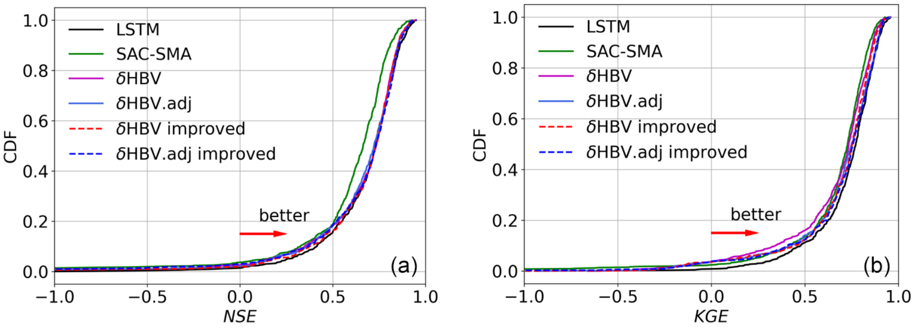


Figure 3. Empirical cumulative distribution function of the test performance metrics for all the models: Nash–Sutcliffe efficiency (NSE, **a**) and Kling–Gupta efficiency (KGE, **b**). LSTM represents a fully data-driven deep learning model previously used in Kratzert et al. (2019), and SAC-SMA is a purely process-based model for which the simulation results are provided in CAMELS. δHBV represents the original differentiable explicit "sequential" HBV model, and δHBV.adj is the implicit adjoint-based HBV model. "Improved" indicates models where a capillary flux was added from the lower subsurface zone to the surface soil to mitigate issues with zero and low flows.

As the implicit model comprehensively improved the streamflow simulation (high and low flows, uncalibrated variables), we conclude that the numerical errors of the sequential model, introduced by the dependence on the order of calculations, have a negative impact on the model's ability to represent hydrological dynamics and fit observations. The differences are admittedly small, but one should not expect major gaps here because the sequential model was already highly competitive and did not leave too much room for improvement (Feng et al., 2022, 2023). It is also worthwhile mentioning that the small differences in the median metrics could manifest as larger differences in capturing some peak events (Fig. 6).

Probing the low-flow issue further, the sequential model seemed to have significant structural deficiencies in representing low flows, which were remediated by using the implicit solver. The sequential model's FLV values were much larger than LSTM's (Fig. 4a and c). With the implicit solver, the adjoint model reduced FLV for a number of regions: (i) on the Great Plains; (ii) in Indiana/Ohio (south of Lake Michigan); and (iii) in some basins in the southeast, including Florida (Fig. 4e). Therefore, the numerical errors with the sequential model exerted a substantial negative impact on the model's ability to accurately represent baseflow. The adjoint model mitigated the overestimation of zero and near-zero flows in arid areas (as seen in Fig. 6, site i) and also corrected the underestimation of the recession limb (refer to Fig. 6, site ii).

We suspect the above-highlighted regions are where effective flow strongly competes with runoff and ET, and thus

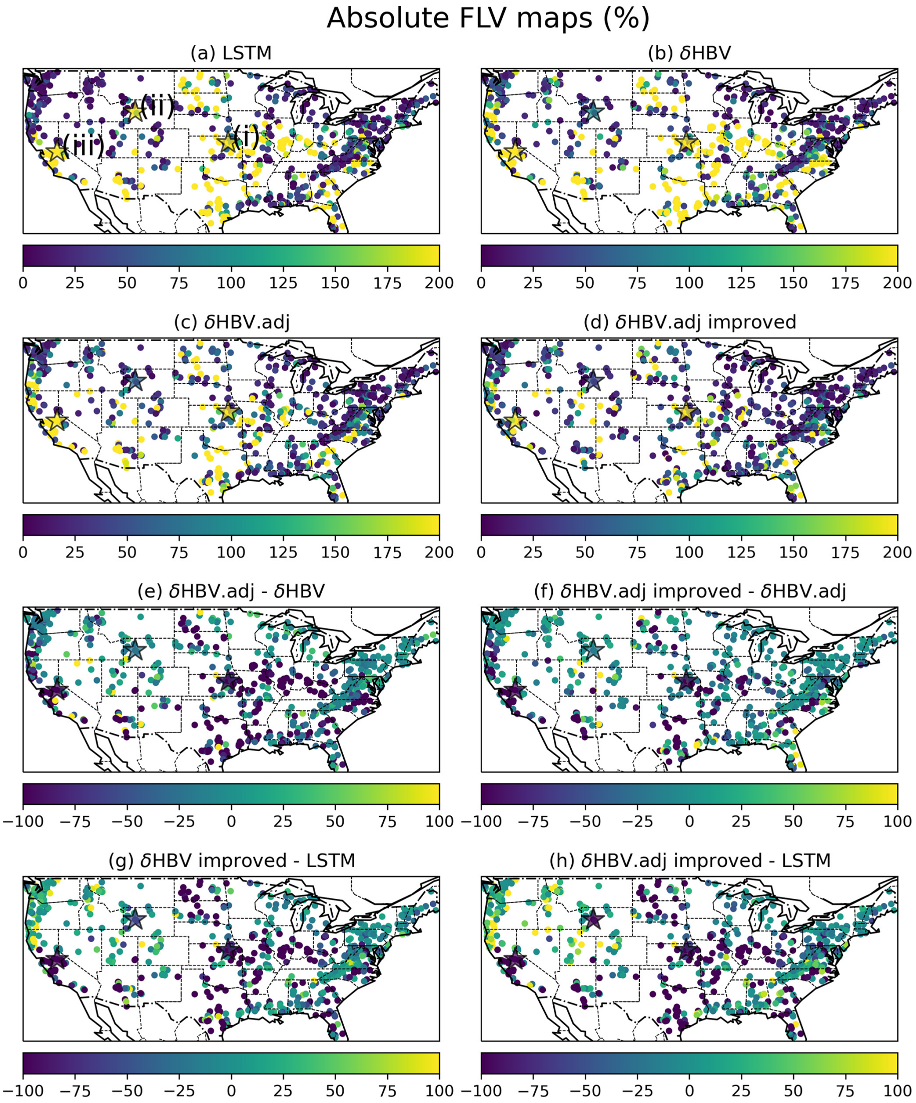


Figure 4. Maps of the percent of the absolute bias of the bottom 30 % ("low") flow range (absolute FLV, %) of **(a)** the LSTM model, **(b)** the sequential model, **(c)** the adjoint model, and **(d)** the adjoint improved model, together with maps of differences in FLV between **(e)** the adjoint model and the sequential model, **(f)** the adjoint improved model and the adjoint model, **(g)** the sequential improved model and the LSTM model, and **(h)** the adjoint improved model and the LSTM model. In panels **(e)**–**(h)**, dark color indicates an improvement in the baseflow representation. The sites annotated in the maps and represented by star-shaped points (and labeled with i, ii, and iii in panel **a**) represent the locations for the plots in Fig. 6.

the order of calculations has large impacts on the separation of fluxes. On the Great Plains, the precipitation tended to be in sync with potential ET, and both were high in summer months (Fang and Shen, 2017) – when the sequential HBV calculates ET before effective flow, there can be significantly less effective flow than if effective flow was computed first. The belt of Indiana/Ohio has compacted soil with high bulk density and difficulty with drainage and thus a shallow water table, which also exists for Florida due to the low relief. In all of these cases, there is competition between effective flow and other processes (ET or excess rainfall): calculating excess first could generate more excess volume than there would be if ET is calculated first. In the arid southwest, the competition between effective flow and ET is also important. The implicit scheme mitigates this problem by solving two operators simultaneously while avoiding overshooting fluxes or stability issues, which enables a better fit to the data.

3.2 The impact of structural changes in HBV

Although the implicit scheme improved the simulations, all the models, including the LSTM, exhibited significant underperformance within the geographical expanse of the Great Plains (Fig. 5), which was particularly pronounced in areas

Table 2. Summary of the statistical hydrological signatures of all the models in the testing period. Bold values denote the best metrics in each column.

Methods	Baseflow Index spatial correlation	Median NSE of the temporal ET simulation
LSTM	–	–
SAC-SMA	–	–
δHBV	0.76	0.59
δHBV.adj	0.83	**0.61**
δHBV improved	0.80	0.54
δHBV.adj improved	**0.86**	0.6

marked by low, or even zero, baseflow conditions (Fig. A1d). In particular, the original HBV model encountered challenges in accurately simulating instances of zero flow due to its structural limitations, resulting in high FLV values (Fig. 4b and c). Specifically, even a minimal precipitation event leads to the creation of a recharge flow in HBV from the soil moisture zone to the subsurface soil zone, which subsequently contributes to the baseflow.

As we enhanced the HBV model by adding a capillary rise from the lower subsurface zone to the soil zone, the baseflow simulations were improved for both the sequential and adjoint models (Table 1 and Fig. 4c and d), with improvements over the default adjoint model in both the absolute FLV (from 43.29 to 37.63 mm d^{-1}) and Baseflow Index (from 0.83 to 0.86). This means that a decent description of baseflow in arid regions needs a mechanism to help the model produce zero or near-zero baseflow, such as returning water from the lower zones to the upper zone (though multiple other structural changes may have similar effects – see Sect. 2.2.2). The structural change mostly continued to improve FLV, substantially reducing FLV within the western coastal regions, the southwestern region, the Great Plains, and the Gulf Coastal Plain, characterized by relatively flat topography and where groundwater-driven flow contributes proportionally less to the overall streamflow dynamics, as evident from the associated low Baseflow Index (Figs. 4f and A1d).

The improvements with baseflows come with a penalty in the high flows. The best adjoint improved model has a low-flow RMSE of 0.047 mm d^{-1}, better than the sequential model, the adjoint model without capillary rise, or LSTM (Table 1). However, the high-flow RMSE did see a slight increase from 2.47 to 2.59 mm d^{-1}, but this was still better than the sequential models (Table 1 and Fig. 6, site iii). A critical divergence in the absolute FLV of δHBV improved models and LSTM appears in the central US (Fig. 4g and h), where accurate flow predictions depend significantly on the fidelity of actual ET simulations.

3.3 Analysis of impacts on parameterization

The adjoint and sequential models achieved optimal performance with different dynamic parameterizations in this temporal test (trained and tested on the same basins but in different periods). The sequential model used γ and β as dynamic parameters, while the adjoint model used dynamic γ, β, and field capacity, θ_{FC}. θ_{FC} plays a significant role in computing effective rainfall, excess, ET, and capillary rise, exerting a substantial influence on the infiltration and recharge mechanisms. Implicit schemes, involving more intricate computations like solving nonlinear equations, enable greater adaptability to data. Adapting θ_{FC} dynamically can improve the model's ability to represent real-world hydrological behavior, such as soil shrink and swell, frozen ground, soil surface sealing, and expansion of the saturation excess areas (or variable source area) (Schneiderman et al., 2007), which are not directly considered in the default HBV.

The spatial patterns of the regionalized HBV parameter values demonstrate moderate differences but still a significant level of consistency between the sequential and adjoint models, suggesting that our regionalization is robust. These patterns also exhibit similarities to the parameter values estimated by Beck et al. (2016) (Fig. A2, reprinted with permission) and also conform well to large-scale climate patterns across the CONUS (Fig. 7a and b). It was known previously that explicit and implicit schemes arrive at very different optimal parameters (Kavetski and Clark, 2010), so we had expected larger discrepancies, but the results show only moderate shifts. Such consistency is likely due to the strong implicit constraints imposed by parameter regionalization using data from the whole CONUS. Since all the basins are served by the same neural network for the mapping, $\theta = g_{\mathrm{w}}(xA)$, it ensures autocorrelation in the parameter fields due to autocorrelation in the used predictors and thus suppresses overfitting to local noise and numerical errors. As a result, the existence of numerical errors alone did not lead to a noisy metric surface for even the explicit model (as shown in Kavetski and Clark, 2010). Previously, it was difficult to efficiently impose such strong constraints and nearly optimally learn the parameters, but the differentiable modeling framework can enable regionalization at low cost and high parallel efficiency.

Delving deeper into the parameter field changes due to employing the implicit solver, we found that the adjoint model seems to have reduced some large (close-to-bound) parameter values, which suggests that parameter compensation for the numerical error is mitigated. The shape coefficient, β, exhibits larger values (> 4) within warm climate regions, while lower values (< 3) characterize cold and mountainous regions (Figs. 7 and A1a). In North Dakota, the Gulf Coastal Plain, and Florida, the adjoint model predicts a reduced β compared to the sequential model. β can influence the flashiness of the peaks, and a larger β tends to cause more threshold-like behaviors. Since the sequential model calculates ET after effective runoff and excess, the available wa-

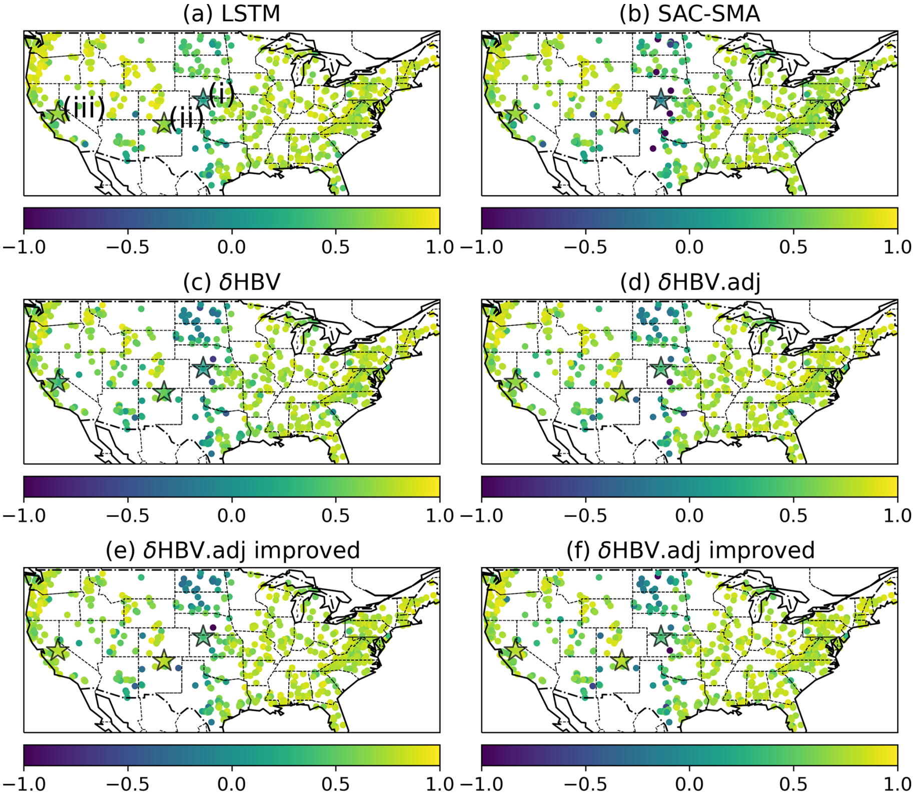


Figure 5. Maps of the KGE for **(a)** LSTM, **(b)** SAC-SMA, **(c)** the sequential model, **(d)** the adjoint model, **(e)** the sequential improved model, and **(f)** the adjoint improved model. The sites annotated in the maps represented by the star-shaped points labeled with (i), (ii), and (iii) represent the locations for the plots in Fig. 6.

ter for runoff is more than that for the adjoint model, which solves the equation implicitly and thus needs such a large β to generate the same amount of effective flow. A similar pattern is observed for the field capacity, θ_{FC}, of the sequential model. However, the adjoint model's field capacity estimation for the northeastern US is notably reduced compared to the sequential model, attributed to a smaller clay fraction and forest fraction (Fig. A1b and c), crudely aligning with the estimated θ_{FC} reported by Beck et al. (2016).

The adjoint model provided a reasonable estimation for other key parameters, including the recession coefficient of the lower subsurface zone (θ_{K_2}) and the wilting point (θ_{LP}). θ_{K_2} usually exhibits a pattern correlated with the BFI (Fig. A1d). A higher BFI indicates greater groundwater-based baseflow, corresponding to a lower θ_{K_2} value that leads to diminished groundwater discharge during low-flow periods. Both the sequential and adjoint models exhibited a consistent θ_{K_2} pattern that contrasts the BFI pattern (Figs. 7 and A1d). Overall, the estimated wilting point, θ_{LP}, of the sequential model is lower than that of the adjoint model, leading to increased ET. As mentioned, ET being underestimated arises from the sequential solving approach of the sequential model, and a smaller θ_{LP} compensates for such numerical errors.

The sequential and adjoint models exhibit similar sensitivity patterns across varying HBV parameters in high-performance basins (as illustrated in Fig. 8a and b and the geographical locations in Fig. 7). This consistency again highlights the stability of the regionalization scheme, in contrast to the noisy metric surface shown in Kavetski and Clark (2010) for single basins. For these two basins, we see a smooth contour where the NN-predicted values (annotated by the symbol "o") were not too far from the optimal value. However, in basins exhibiting poor performance (as depicted in Fig. 8c and d), their differences enlarged and apparently numerical errors shifted the parameter distributions. Although the overall contour patterns remain similar, the values of the contours have changed quite significantly, due to the dependence on the calculation order.

The process of parameter regionalization introduces a tradeoff between performance and spatial coherence, leading to parameters that might not be optimal for each specific basin; when this gap is too large, it suggests that there might be some structural issues or missing information. Take basin (D) (Fig. 8d) as an example, where the optimal values for θ_{FC} and β fall within the ranges of 100–200 mm and 1.0–4.0, respectively, but the NN-predicted parameter values (centers of the KGE contours in Fig. 8) deviate signif-

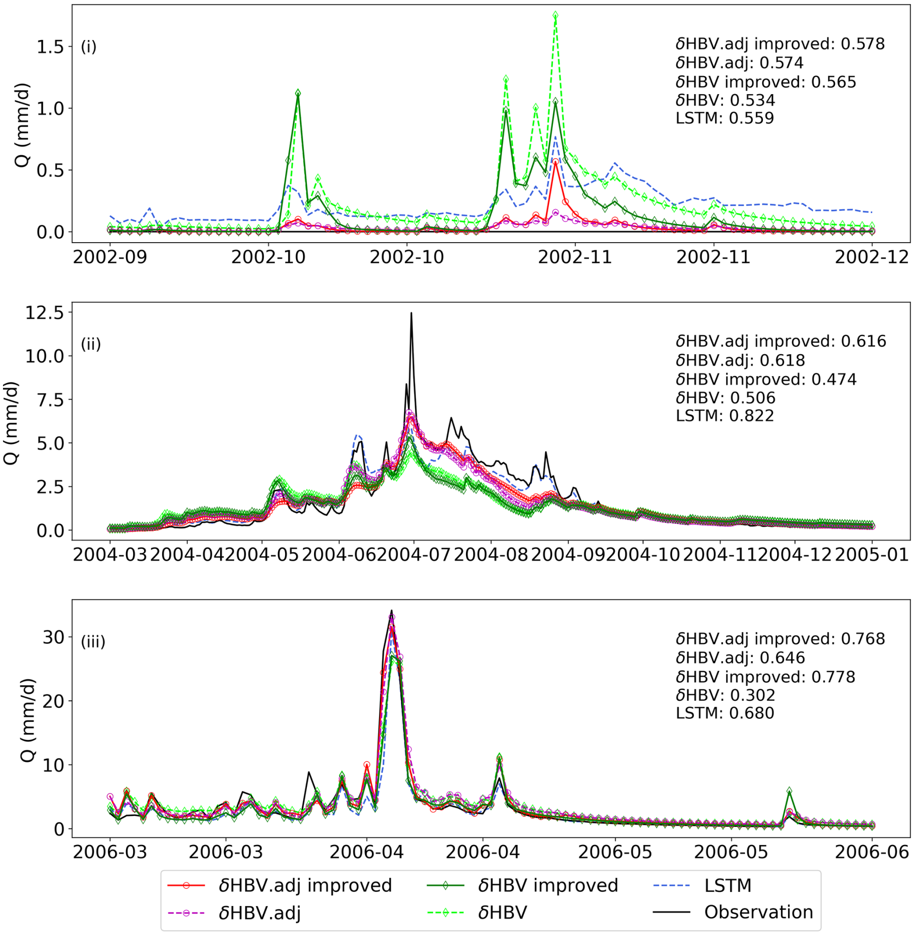


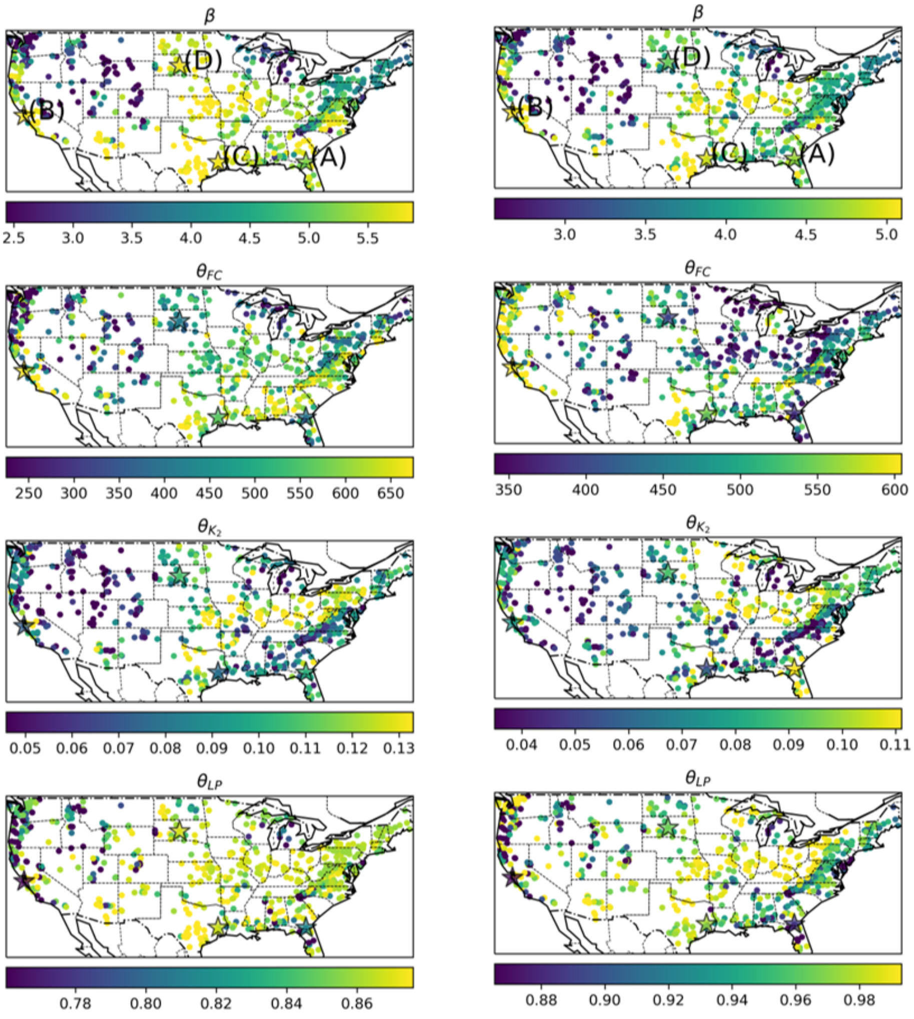
Figure 6. The time series of streamflow simulations from the models and observations at the locations annotated in Figs. 4 and 5. The KGE values of the models in the whole testing period are listed in the subfigures.

icantly from these optimal ranges. The regionalized parameters thus produced a rather low KGE of −0.3. This tradeoff could mean that some key processes are not well represented and the parameters could potentially compensate for these processes, but the compensation was prevented by regionalization. A notable example is the absence of topographic information and subbasin-scale spatial heterogeneity, which are crucial for modeling arid basins but were not fully considered by the present parameterization network. These parameter gaps give us hints for the next stage of model improvements.

4 Further discussion

While this work focuses on enabling implicit solvers in differentiable modeling, we do not suggest that explicit solvers are to be discouraged. It has long been explored in the numerical algorithm literature that each type of solver has advantages and disadvantages and is suitable for different problems. For example, implicit solvers are not only preferred, but are also necessary for stiff ODEs, especially those with dynamics on vastly different timescales and those resulting from the discretization of elliptic PDEs. Using explicit solvers for them could necessitate very small time steps.

Further complications of using explicit schemes with small time steps include computational expenses, parallel efficiency, and matching forcing functions. Even though hourly data are now publicly available, directly training an hourly model with ML techniques remains computationally expensive and may also cause the problem of gradient vanishing if the training time steps are too numerous (Gauch et al., 2021; Greff et al., 2017). The numerical schemes employed in the physical models within the differentiable modeling framework need to maintain stability for simultaneous large-scale simulations in each mini batch while also allowing for gradient tracking. Batched learning and parallel efficiency may prefer uniform operations across basins and challenge the application of adaptive time-stepping algorithms. We conducted tests on the differentiable HBV model with various numerical schemes and fixed smaller time steps (Table A2 and Fig. A3). The sequential model and implicit adjoint



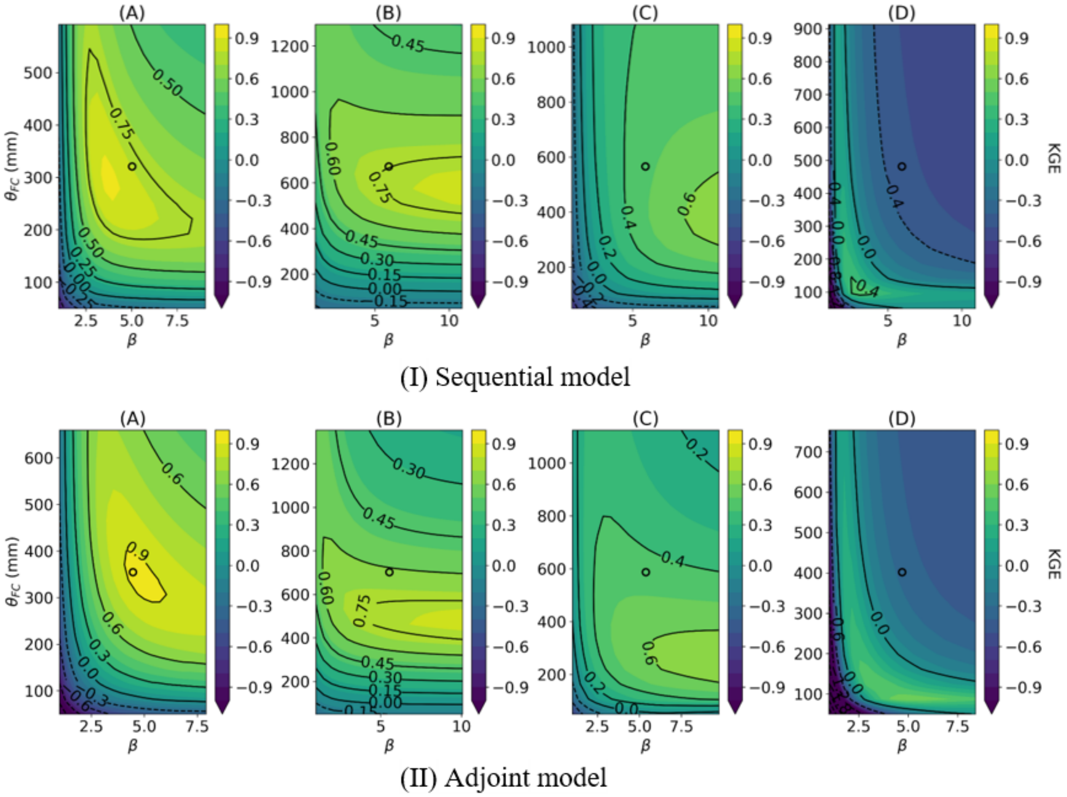
(a) Sequential model

(b) Adjoint model

Figure 7. Map of the optimized parameter β, field capacity θ_{FC}, recession coefficient of the lower subsurface zone θ_{K_2}, and wilting point θ_{LP} from **(a)** the sequential model and **(b)** the adjoint model. The sites annotated on the maps as star-shaped points labeled with the letters A–D represent the locations for the plots in Fig. 8.

model with a 1 d time step presented higher performance than the explicit Euler schemes with smaller time steps or the fourth-order Runge–Kutta scheme. The main reason may be that the daily forcing inputs and daily physical parameters from the neural network do not match the smaller time steps within a day. Thus, explicit schemes with smaller time steps may be complicated by the need for matching forcing functions as well. Some multi-timescale ML techniques have been used to predict hourly flood hydrographs using daily flow data to avoid gradient-vanishing issues in the direct hourly training (Gauch et al., 2021; Sarıgöl and Katipoğlu, 2023). These approaches present possible solutions for future investigations.

The parameterization function (the neural network) embedded in the differentiable models demonstrates robustness, as evidenced by the similarity of parameter patterns and metric surfaces derived from various numerical schemes in Figs. 7 and A3. We did not observe a notable macroscale roughness in the metric surface (Fig. A3) as shown in Kavetski and Clark (2010) when using explicit schemes. Moderate distortions and roughness were present on the KGE surface in models employing the RK scheme (sites A and D). As we reduced the time steps and transitioned to implicit schemes, these distortions seem to have alleviated and converged toward the metric surface, consistent with the better numerical solution. That is, the 4-hourly and hourly patterns are more similar to the implicit results than that of the RK scheme. The convergence toward the implicit scheme suggests that the implicit scheme results are more reliable.



(I) Sequential model

(II) Adjoint model

Figure 8. Impact of numerical schemes on the KGE surface of the HBV model: the contour of KGE calculated from the (I) sequential model and (II) adjoint model on the 2D slice of field capacity (θ_{FC}) and parameter β. The predicted parameter values are positioned at the central point of the contours delineated by circles. The locations of the selected sites are annotated in Fig. 7.

The adjoint method was used in this work to support the implicit numerical scheme in differentiable models, which allows for the efficient joint training of the neural networks with physical models using gradient descent. Such joint and "online" training on big data is not possible without the process-based model being differentiable, because the only presently known way to train such a large number of weights is via backpropagation. While we utilized a coupled neural network in this work for regional parameterization, it can also be used to replace any component of the physical model for knowledge discovery. For example, the runoff module in the HBV model could be replaced with a neural network. Thanks to their ability to train on big data, differentiable models can be trained on all basins and constrain a common neural network to learn a universal relationship between the inputs and the physical parameters. Such relationships can be used for interpolation and extrapolation at sites lacking observations.

While the differentiable implicit model outperforms the sequential one and offers state-of-the-art performance, it incurs a substantially larger computational cost. Newton's method to solve the implicit equation requires a number of iterations (approximately three to four iterations to solve Eq. 29) and, for the sake of the adjoint, we need to solve a matrix (Eq. 33). The calculation of the Jacobian matrix for multiple basins, depending on the batch size, also consumes time. In addition, the more complicated computational instructions of the adjoint procedure may encounter higher CPU overhead and thus a lower GPU utilization rate compared to the forward simulations. For all these reasons, the computational cost is almost 5–10 times that of the sequential model. For context, compared to running traditional models on CPUs, our implementation is already orders of magnitude more efficient due to parallel efficiency. Nonetheless, the increased computational demand for the implicit solver still creates challenges for training at large scales, as it requires thousands of forward simulations. As a potential solution, based on the parameter consistency between the sequential and implicit models, it seems we can use the inexpensive sequential model as an "explorer model" for model structure identification, hyperparameter tuning, and neural network pretraining. Then we can fine-tune the network using the adjoint model. Hence, both models offer utility for global-scale applications. Although the adjoint model already outperformed LSTM in terms of low-flow and high-flow RMSE values, which is an astounding result, they still fell slightly behind LSTM at certain basins, with poor performance in the central and western US (arid regions). In the future, we can assess multiple hypotheses that may explain why the adjoint model's performance in these regions is not as good as that of LSTM. (A) The differentiable model is more hampered by precipitation bias in these regions than is LSTM, which can internally account for predictable bias. (B) HBV's baseflow inadequacy arises from not fully utilizing forcing information. LSTM can fully utilize information in the inputs, e.g., solar radiation and vapor pressure, while the present HBV model only uses temperature in determining PET, apart from precipitation. The difference in solar radiation and vapor pressure could have impacted long-term water balance and baseflow. Future work could use the Penman–Monteith equation for PET, which considers vapor pressure, or learn better PET equations from data (Zhao et al., 2019). (C) HBV faces a larger tradeoff between matching the high- and low-flow portions of the observed hydrographs, and the adjoint models sacrificed low-flow performance to some extent in favor of better overall performance. (D) The model backbone, HBV, is unable to represent some groundwater dynamics, e.g., lateral redistribution of moisture from hillslope to valley (Clark et al., 2015; Fan et al., 2019). While LSTM could internally form a cascade of neurons that transfers mass akin to lateral groundwater movements, the two-layer groundwater structure in HBV is too simple to represent such impacts. These hypotheses will lead us to improved model structures with the help of data. An unprecedented advantage with differentiable modeling is that we can simultaneously learn robust relationships from large data and find nearly optimal parameterization schemes. This reduces the complex, iterative testing between different model structures and parameter optimization.

In future research, it would be valuable to evaluate the advantages and disadvantages of optimize-then-discretize versus discretize-then-optimize adjoint methods in hydrologically differentiable models. Generally, the optimize-then-discretize approach tends to be less accurate because of gradient inaccuracies when the adjoint state differential equations are not sufficiently resolved. On the other hand, the adjoint in the discretize-then-optimize method is solved from the Jacobian matrix using automatic differentiation, offering greater efficiency and accuracy compared to numerically solving the adjoint differential equation (Onken and Ruthotto, 2020). Nevertheless, more work is required to thoroughly compare the two options.

The adjoint used in this work is derived for the gradient of the Newton solver such that it can theoretically support any model that can be solved with the Newton solver – not only bucket models governed by ODEs, but also distributed models governed by PDEs. However, the challenge may still exist in calculating the Jacobian matrix for a batch of basins for PDEs, as the distributed parameters in PDEs are significantly greater in number than those in ODEs and can slow down the efficiency of the model in both forward and backward modes.

AD is a tool that we seldom used prior to the prevalence of deep learning, nor did we have modern GPUs or the software to maximize its utilization. The past few years saw substantial software and hardware investments from the artificial intelligence community that have made these tools orders of magnitude more efficient. Utilizing these tools and running hydrological models on such platforms means the water community can leverage these investments and can grow with the AI community at little cost. For example, the model can automatically become even faster with slight effort to embrace just-in-time compilation of Torch 2.0 (Wu, 2023).

5 Conclusions

Our comparisons show that the numerical errors associated with the sequential model, and especially its dependence on the order of computation, had a detrimental impact on its representation power – it cannot provide high-quality low-flow, high-flow, and groundwater simulations and can introduce parameter compensations. The adjoint method for gradient calculation enables the use of implicit solvers in differentiable modeling, partially mitigating the numerical errors. While not explicitly demonstrated, other hydrological problems that require implicit solvers can similarly benefit from the adjoint method. With the implicit solver and with a structural change (capillary rise), our model comprehensively improved the simulations of low flow and an uncalibrated variable, baseflow fraction. While some of the differences in metrics may not seem large, they are already significant and could result in flood peaks being more accurately predicted. The comparison of baseflow simulations also implies that the same numerical issue may hamper other models, so that, in order to achieve top-of-the-line performance with differentiable models, numerical errors have to be examined, and the implicit model and adjoint will be needed.

The capacity of differentiable models to outperform LSTM in low-flow and high-flow metrics at the median of CAMELS basins proves that structural priors (and physical interpretability) and state-of-the-art performance are not mutually exclusive, and pure deep networks are not necessarily the performance ceiling of environmental models (although we do expect them to be close to optimal). In fact, for rarely observed events or spatial extrapolation, structural priors may potentially overcome data limitations. The fact that modifying the structure can result in better physical representations hints that we can make further improvements to baseflow and peak flow and identify better structures from data.

The regionalization scheme produced overall stable parameter fields that, on first look, have similar patterns between sequential and implicit models, but a deeper investigation shows that the implicit scheme reduced large, near-bound parameter values where competition between fluxes is likely to occur. This is visual evidence that parameter compensation occurs more strongly with the sequential model and can be mitigated. Since it is preferable to remove the interference of numerical errors prior to interpreting the parameter fields, the implicit model would be favored when the interest is in the intermediate parameters or internal fluxes. The ancient demon of numerical errors remains relevant in the new era of big data, but may be mitigated by the adaptive capability of deep networks.

Appendix A

Table A1 shows the forcing and attribute variables used in the LSTM models.

We provided the maps with four attributes (aridity, forest fraction, clay fraction, and Baseflow Index) used in the differential models to support our analysis (Fig. A1).

We reprinted Fig. 4 of Beck et al. (2016) to facilitate a comprehensive parameter comparison (Fig. A2).

We conducted tests on the differentiable HBV model with various numerical schemes and time steps (Table A2, Fig. A3). The forcing and physical parameters estimated by the neural network (LSTM) remain constant within a day. In theory, with smaller time steps, we were supposed to configure the forcing function and LSTM to provide hourly inputs and physical parameters that match the progression of time within a day; i.e., the inputs should reflect diurnal changes in forcing. However, this configuration would greatly increase the memory usage and complexity and is thus beyond the scope of this study (Gauch et al., 2021). Here we distinguish between an explicit scheme and a sequential scheme with operator splitting: the explicit scheme solves the right-hand side of the ODE simultaneously, while the sequential scheme applies an order to the operations, generally from the surface to the subsurface, as directed in the original HBV. The fixed-step explicit Euler scheme with a 1 d time step caused divergence in the large-scale simulation due to its instability. However, with 4 and 1 h time steps, the explicit Euler schemes exhibited better performance than the fourth-order Runge–Kutta (RK) explicit scheme but still lagged behind the daily sequential scheme that employed ad hoc operation splitting or the implicit adjoint scheme. The regional parameters learned through various numerical schemes exhibit similarity, indicating the robustness of the parameterization function (the neural network) embedded within the differentiable models (Fig. A3). However, parameter distortion and surface roughness in the KGE models employing explicit schemes, particularly the RK scheme, were still observed.

Table A1. Summary of the forcing and attribute variables used in all the models.

	Variable	Name	Unit
Forcings	PRCP	Precipitation	$\mathrm{mm\,d^{-1}}$
	E_p	Potential evapotranspiration	$\mathrm{mm\,d^{-1}}$
	T	Temperature	°C
Attributes	p_mean	Mean daily precipitation	$\mathrm{mm\,d^{-1}}$
	pet_mean	Mean daily PET	$\mathrm{mm\,d^{-1}}$
	p_seasonality	Seasonality and timing of precipitation	–
	frac_snow	Fraction of precipitation falling as snow	–
	Aridity	PET/P	–
	high_prec_freq	Frequency of high-precipitation days	$\mathrm{d\,yr^{-1}}$
	high_prec_dur	Average duration of high-precipitation events	d
	low_prec_freq	Frequency of dry days	$\mathrm{d\,yr^{-1}}$
	low_prec_dur	Average duration of dry periods	d
	elev_mean	Catchment mean elevation	m
	slope_mean	Catchment mean slope	$\mathrm{m\,km^{-1}}$
	area_gages2	Catchment area (GAGESII estimate)	$\mathrm{km^2}$
	frac_forest	Forest fraction	–
	lai_max	Maximum monthly mean of the leaf area index	–
	lai_diff	Difference between the maximum and minimum monthly means of the leaf area index	–
	gvf_max	Maximum monthly mean of the green vegetation	–
	gvf_diff	Difference between the maximum and minimum monthly mean of the green vegetation fraction	–
	dom_land_cover_frac	Fraction of the catchment area associated with the dominant land cover	–
	dom_land_cover	Dominant land cover type	–

Table A1. Continued.

	Variable	Name	Unit
Attributes	root_depth_50	Root depth at the 50th percentile, extracted from a root depth distribution based on the International Geosphere-Biosphere Programme (IGBP) land cover	m
	soil_depth_pelletier	Depth to bedrock	–
	soil_depth_statgso	Soil depth	m
	soil_porosity	Volumetric soil porosity soil_conductivity	–
	soil_conductivity	Saturated hydraulic conductivity	$\mathrm{cm\,h^{-1}}$
	max_water_content	Maximum water content	m
	sand_frac	Sand fraction	–
	silt_frac	Silt fraction	–
	clay_frac	Clay fraction	–
	geol_class_1st	Most common geologic class in the catchment basin	–
	geol_class_1st_frac	Fraction of the catchment area associated with its most common geologic class	–
	geol_class_2nd	Second most common geologic class in the catchment basin	–
	geol_class_2nd_frac	Fraction of the catchment area associated with its second most common geologic class	–
	carbonate_rocks_frac	Fraction of the catchment area as carbonate sedimentary rocks	–
	geol_porosity	Subsurface porosity	–
	geol_permeability	Subsurface permeability	$\mathrm{m^2}$

Table A2. Summary of streamflow metrics for models using different numerical schemes and time steps. Timing was obtained on an Nvidia Tesla V100 GPU.

Model	Numerical scheme	Time step	Memory usage per batch	Computational time per batch	Median NSE RMSE (mm d^{-1})	Median KGE flow RMSE (mm d^{-1})	Median low flow	Median peak
δHBV	Fixed-step explicit	1 day	2274 M	1.6 s	–	–	–	–
δHBV	Fourth-order Runge–Kutta explicit	1 d	2532 M	3.9 s	0.69	0.70	0.061	3.25
δHBV	Fixed-step explicit	4 h	2706 M	6.3 s	0.72	0.71	0.09	2.50
δHBV	Fixed-step explicit	1 h	4146 M	18.1 s	0.72	0.71	0.08	2.63
δHBV	Sequential	1 d	2266 M	1.8 s	0.73	0.73	0.074	2.56
δHBV.adj	Implicit adjoint	1 d	2788 M	19.5 s	0.72	0.75	0.048	2.47

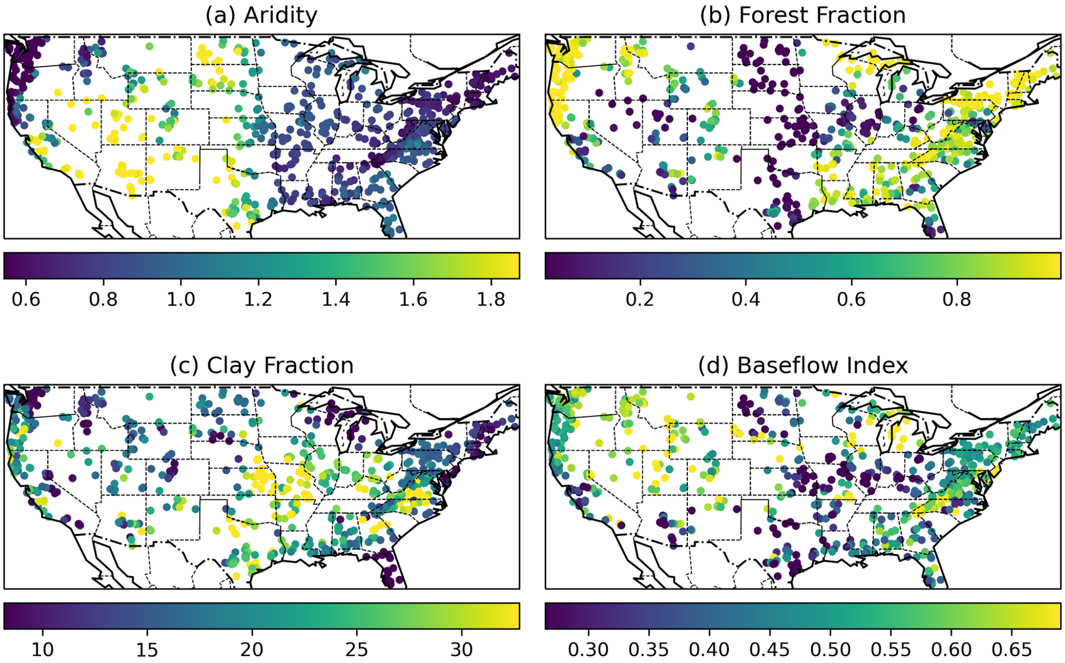


Figure A1. Map of the static attributes: **(a)** aridity, **(b)** forest fraction, **(c)** clay fraction, and **(d)** Baseflow Index from the CAMELS dataset.

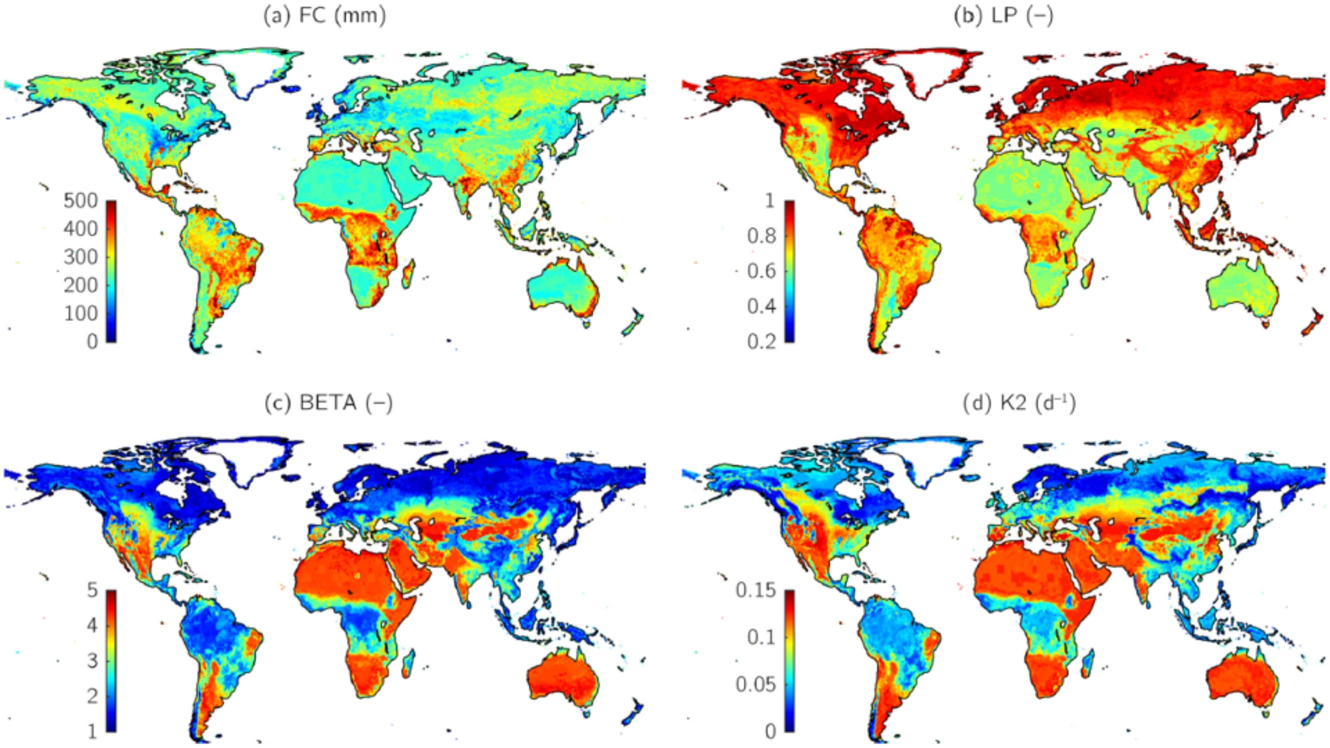
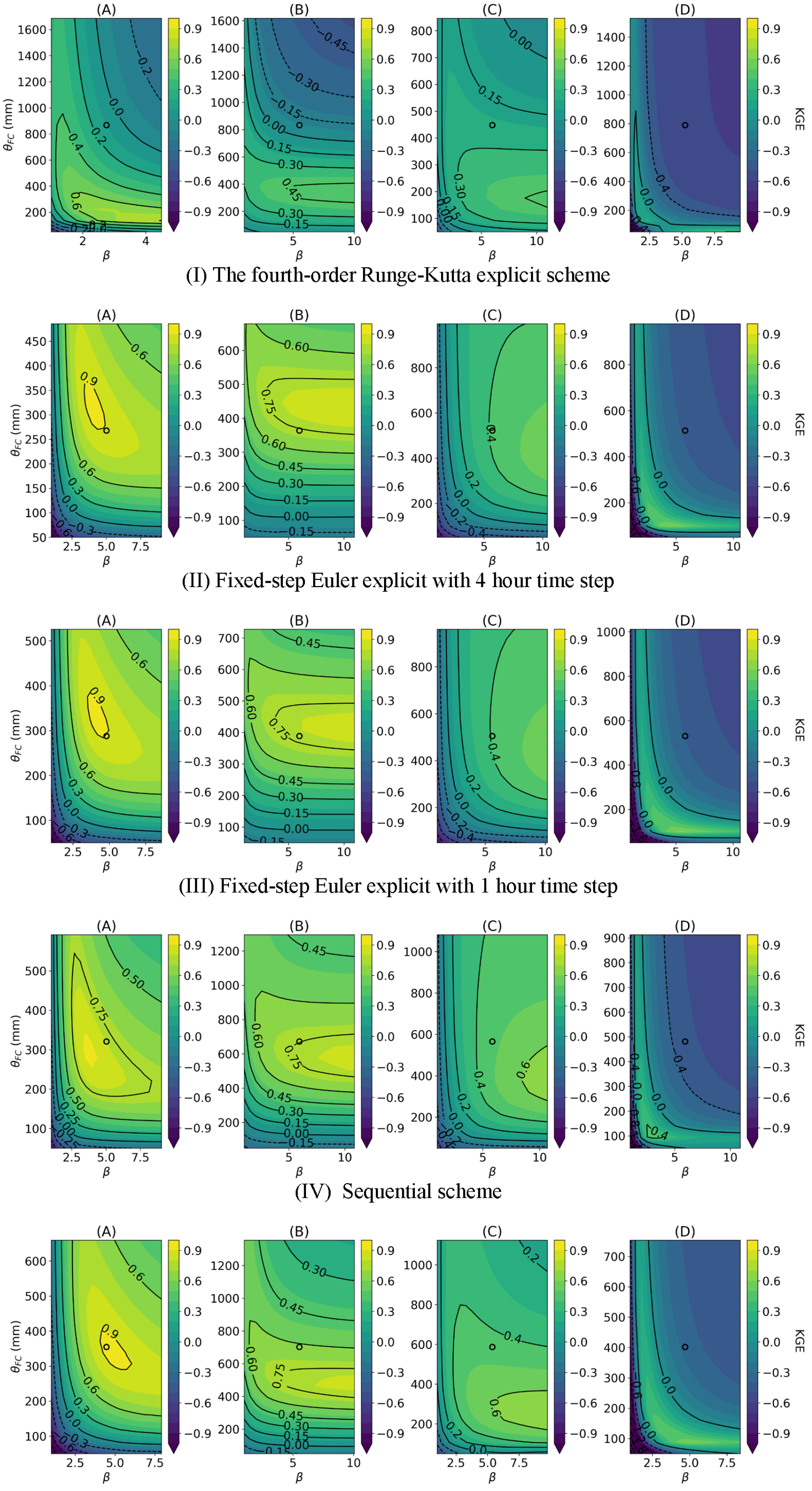


Figure A2. For HBV, mean values of the regionalized parameters based on the 10 most similar donor catchments for **(a)** the maximum water storage in the unsaturated-zone store (FC), **(b)** the soil moisture value above which actual evaporation reaches potential evaporation (LP), **(c)** the shape coefficient of the recharge function (β), and **(d)** the recession coefficient of the lower groundwater store (K_2). Reprinted with permission from Beck et al. (2016).



(I) The fourth-order Runge-Kutta explicit scheme

(II) Fixed-step Euler explicit with 4 hour time step

(III) Fixed-step Euler explicit with 1 hour time step

(IV) Sequential scheme

(V) Implicit adjoint scheme

Figure A3. Impact of numerical schemes on the KGE surface of the HBV model: the contour of the KGE calculated from the (I) fourth order Runge–Kutta explicit scheme, (II) fixed-step Euler explicit scheme with a 4 h time step, (III) fixed-step Euler explicit scheme with a 1 h time step, (IV) sequential scheme, and (V) implicit adjoint scheme, on the 2D slice of the FC and parameter. The predicted parameter values are positioned at the central point of the contours delineated by circles. The locations of the selected sites are annotated in Fig. 7.

Data availability. Our code for the adjoint differentiable models can be downloaded at https://doi.org/10.5281/zenodo.11205309 (Song, 2024a) or https://github.com/mhpi/HydroDLAdj (Song, 2024b). The CAMELS dataset can be downloaded at https://doi.org/10.5065/D6MW2F4D (Newman et al., 2014; Addor et al., 2017). MODIS ET data can be downloaded at https://doi.org/10.5067/MODIS/MOD16A2.006 (Running et al., 2017).

Author contributions. All the authors made significant contributions to this work. YS contributed to the conceptualization, formal analysis, methodology, and writing. WK contributed to the formal analysis, methodology, and writing. MC contributed to the methodology and writing. DF contributed to the software. KL contributed to the writing. KS contributed to the validation. CS contributed to the conceptualization, formal analysis, methodology, and writing.

Competing interests. Kathryn Lawson and Chaopeng Shen have financial interests in HydroSapient, Inc., a company which could potentially benefit from the results of this research. This interest has been reviewed by The Pennsylvania State University in accordance with its individual conflict of interest policy for the purpose of maintaining the objectivity and the integrity of research. The other authors have no competing interests to declare.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Acknowledgements. Yalan Song and Chaopeng Shen were supported primarily by sub-award A22-0307-S003 from the Cooperative Institute for Research to Operations in Hydrology (CIROH) through an NOAA Cooperative Agreement (grant no. NA22NWS4320003) and partially by the US Department of Energy under award DE-SC0016605. Wouter J. M. Knoben and Martyn P. Clark were also supported by CIROH. Dapeng Feng was supported by US National Science Foundation (NSF) EAR-2221880. Computational resources were partially provided by NSF OAC-1940190.

Financial support. This research has been supported by the National Oceanic and Atmospheric Administration (grant no. NA22NWS4320003), the US Department of Energy (grant no. DE-SC0016605), and the National Science Foundation (grant nos. EAR-2221880 and OAC-1940190).

Review statement. This paper was edited by Ralf Loritz and reviewed by Uwe Ehret and Ilhan Özgen-Xian.

References

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

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P.: Catchment Attributes and MEteorology for Large-Sample studies (CAMELS) version 2.0, NCAR, https://doi.org/10.5065/D6G73C3Q, 2017.

Aghakouchak, A. and Habib, E.: Application of a Conceptual Hydrologic Model in Teaching Hydrologic Processes, Int. J. Eng. Educ., 26, 963–973, 2010.

Bandai, T.: Inverse Modeling of Soil Moisture Dynamics: Estimation of Soil Hydraulic Properties and Surface Water Flux, PhD thesis, University of California, Merced, California, USA, 172 pp., https://escholarship.org/uc/item/8gb9m1gm#article_main (last access: 11 July 2024), 2022.

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

Beck, H. E., Pan, M., Lin, P., Seibert, J., van Dijk, A. I. J. M., and Wood, E. F.: Global fully distributed parameter regionalization based on observed streamflow from 4,229 headwater catchments, J. Geophys. Res.-Atmos., 125, e2019JD031485, https://doi.org/10.1029/2019JD031485, 2020.

Bennett, A. and Nijssen, B.: Deep learned process parameterizations provide better representations of turbulent heat fluxes in hydrologic models, Water Resour. Res., 57, e2020WR029328, https://doi.org/10.1029/2020WR029328, 2021.

Bergman, T. L. (Ed.): Introduction to heat transfer, 6th Edn., Wiley, Hoboken, NJ, 960 pp., https://books.google.com/books?hl=en&lr=&id=YBaNaLurTD4C&oi=fnd&pg=PR21&dq=Introduction+to+heat+transfer&ots=tRNGbglJeB&sig=MLCrQzH7a0CeETw72S0iUGEqS0g (last access: 11 July 2024), 2011.

Bergström, S.: Development and application of a conceptual runoff model for Scandinavian catchments, PhD thesis, SMHI – Swedish Meteoro-logical and Hydrological Institute, http://urn.kb.se/resolve?urn=urn:nbn:se:smhi:diva-5738 (last access: 11 July 2024), 1976.

Bergström, S.: The HBV model – Its structure and applications, RH No. 4; SMHI Reports), SMHI – Swedish Meteorological and HydrologicalInstitute, https://www.smhi.se/en/publications/the-hbv-model-its-structure-and-applications-1.83591 (last access: 11 July 2024), 1992.

Bindas, T., Tsai, W.-P., Liu, J., Rahmani, F., Feng, D., Bian, Y., Lawson, K., and Shen, C.: Improving river routing using a differentiable Muskingum–Cunge model and physics-informed machine learning, Water Resour. Res., 60, e2023WR035337, https://doi.org/10.1029/2023WR035337, 2024.

Cao, Y., Li, S., and Petzold, L.: Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software, J. Comput. Appl. Math., 149, 171–191, https://doi.org/10.1016/S0377-0427(02)00528-9, 2002.

Castaings, W., Dartus, D., Le Dimet, F.-X., and Saulnier, G.-M.: Sensitivity analysis and parameter estimation for distributed hydrological modeling: potential of variational methods, Hydrol. Earth Syst. Sci., 13, 503–517, https://doi.org/10.5194/hess-13-503-2009, 2009.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D.: Neural ordinary differential equations, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montreal, Canada, 6572–6583, arXiv [preprint], https://doi.org/10.48550/arXiv.1806.07366, 2018.

Clark, M. P. and Kavetski, D.: Ancient numerical daemons of conceptual hydrological modeling: 1. Fidelity and efficiency of time stepping schemes, Water Resour. Res., 46, W10510, https://doi.org/10.1029/2009WR008894, 2010.

Clark, M. P., Fan, Y., Lawrence, D. M., Adam, J. C., Bolster, D., Gochis, D. J., Hooper, R. P., Kumar, M., Leung, L. R., Mackay, D. S., Maxwell, R. M., Shen, C., Swenson, S. C., and Zeng, X.: Improving the representation of hydrologic processes in Earth System Models, Water Resour. Res., 51, 5929–5956, https://doi.org/10.1002/2015WR017096, 2015.

Colleoni, F., Garambois, P.-A., Javelle, P., Jay-Allemand, M., and Arnaud, P.: Adjoint-based spatially distributed calibration of a grid GR-based parsimonious hydrological model over 312 French catchments with SMASH platform, EGUsphere [preprint], https://doi.org/10.5194/egusphere-2022-506, 2022.

Fan, Y., Clark, M., Lawrence, D. M., Swenson, S., Band, L. E., Brantley, S. L., Brooks, P. D., Dietrich, W. E., Flores, A., Grant, G., Kirchner, J. W., Mackay, D. S., McDonnell, J. J., Milly, P. C. D., Sullivan, P. L., Tague, C., Ajami, H., Chaney, N., Hartmann, A., Hazenberg, P., McNamara, J., Pelletier, J., Perket, J., Rouholahnejad-Freund, E., Wagener, T., Zeng, X., Beighley, E., Buzan, J., Huang, M., Livneh, B., Mohanty, B. P., Nijssen, B., Safeeq, M., Shen, C., Verseveld, W., Volk, J., and Yamazaki, D.: Hillslope hydrology in global change research and earth system modeling, Water Resour. Res., 55, 1737–1772, https://doi.org/10.1029/2018WR023903, 2019.

Fang, K. and Shen, C.: Full-flow-regime storage-streamflow correlation patterns provide insights into hydrologic functioning over the continental US, Water Resour. Res., 53, 8064–8083, https://doi.org/10.1002/2016WR020283, 2017.

Fang, K., Shen, C., Kifer, D., and Yang, X.: Prolongation of SMAP to spatiotemporally seamless coverage of continental U.S. using a deep learning neural network, Geophys. Res. Lett., 44, 11030–11039, https://doi.org/10.1002/2017gl075619, 2017.

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

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

Feng, D., Beck, H., Lawson, K., and Shen, C.: The suitability of differentiable, physics-informed machine learning hydrologic models for ungauged regions and climate change impact assessment, Hydrol. Earth Syst. Sci., 27, 2357–2373, https://doi.org/10.5194/hess-27-2357-2023, 2023.

Fisher, M. and Andersson, E.: Developments in 4D-Var and Kalman Filtering, European Centre for Medium Range Weather Forecasts, Shinfield Park, Reading, Berkshire, England, https://www.ecmwf.int/sites/default/files/elibrary/2001/9409-developments-4d-var-and-kalmanfiltering.pdf (last access: 11 July 2024), 2001.

Frame, J. M., Kratzert, F., Raney II, A., Rahman, M., Salas, F. R., and Nearing, G. S.: Post-Processing the National Water Model with Long Short-Term Memory Networks for Streamflow Predictions and Model Diagnostics, J. Am. Water Resour. Assoc., 57, 885–905, https://doi.org/10.1111/1752-1688.12964, 2021.

Gauch, M., Kratzert, F., Klotz, D., Nearing, G., Lin, J., and Hochreiter, S.: Rainfall–runoff prediction at multiple timescales with a single Long Short-Term Memory network, Hydrol. Earth Syst. Sci., 25, 2045–2062, https://doi.org/10.5194/hess-25-2045-2021, 2021.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J.: LSTM: A Search Space Odyssey, IEEE Trans. Neural Netw. Learn. Syst., 28, 2222–2232, https://doi.org/10.1109/TNNLS.2016.2582924, 2017.

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

Hannah, D. M., Demuth, S., van Lanen, H. A. J., Looser, U., Prudhomme, C., Rees, G., Stahl, K., and Tallaksen, L. M.: Large-scale river flow archives: importance, current status and future needs, Hydrol. Process., 25, 1191–1200, https://doi.org/10.1002/hyp.7794, 2011.

Hargreaves, G. H.: Defining and using reference evapotranspiration, J. Irrig. Drain. Eng., 120, 1132–1139, https://doi.org/10.1061/(ASCE)0733-9437(1994)120:6(1132), 1994.

Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, Neural Comput., 9, 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735, 1997.

Hornik, K., Stinchcombe, M., and White, H.: Multilayer feedforward networks are universal approximators, Neural Netw., 2, 359–366, https://doi.org/10.1016/0893-6080(89)90020-8, 1989.

Jay-Allemand, M., Javelle, P., Gejadze, I., Arnaud, P., Malaterre, P.-O., Fine, J.-A., and Organde, D.: On the potential of variational calibration for a fully distributed hydrological model: application on a Mediterranean catchment, Hydrol. Earth Syst. Sci., 24, 5519–5538, https://doi.org/10.5194/hess-24-5519-2020, 2020.

Kavetski, D. and Clark, M. P.: Ancient numerical daemons of conceptual hydrological modeling: 2. Impact of time stepping schemes on model analysis and prediction, Water Resour. Res., 46, W10511, https://doi.org/10.1029/2009WR008896, 2010.

Knoben, W. J. M., Freer, J. E., Fowler, K. J. A., Peel, M. C., and Woods, R. A.: Modular Assessment of Rainfall–Runoff Models Toolbox (MARRMoT) v1.2: an open-source, extendable framework providing implementations of 46 conceptual hydrologic models as continuous state-space formulations, Geosci. Model Dev., 12, 2463–2480, https://doi.org/10.5194/gmd-12-2463-2019, 2019.

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., and Nearing, G.: Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets, Hydrol. Earth Syst. Sci., 23, 5089–5110, https://doi.org/10.5194/hess-23-5089-2019, 2019.

Ladson, A. R., Brown, R., Neal, B., and Nathan, R.: A standard approach to baseflow separation using the Lyne and Hollick filter, Australas, J. Water Resour., 17, 25–34, 2013.

Liu, Y. and Gupta, H. V.: Uncertainty in hydrologic modeling: Toward an integrated data assimilation framework, Water Resour. Res., 43, W07401, https://doi.org/10.1029/2006WR005756, 2007.

Monteith, J. L.: Evaporation and environment, Symp. Soc. Exp. Biol., 19, 205–234, 1965.

Mu, Q., Zhao, M., and Running, S. W.: Improvements to a MODIS global terrestrial evapotranspiration algorithm, Remote Sens. Environ., 115, 1781–1800, https://doi.org/10.1016/j.rse.2011.02.019, 2011.

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

Neupauer, R. M. and Wilson, J. L.: Adjoint-derived location and travel time probabilities for a multidimensional groundwater system, Water Resour. Res., 37, 1657–1668, https://doi.org/10.1029/2000WR900388, 2001.

Newman, A., Sampson, K., Clark, M. P., Bock, A., Viger, R. J., and Blodgett, D.: A large-sample watershed-scale hydrometeorological dataset for the contiguous USA, UCAR/NCAR [data set], https://doi.org/10.5065/D6MW2F4D, 2014.

Onken, D. and Ruthotto, L.: Discretize-Optimize vs. Optimize-Discretize for Time-Series Regression and Continuous Normalizing Flows, arXiv [preprint], https://doi.org/10.48550/arXiv.2005.13420, 2020.

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

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., and Edelman, A.: Universal Differential Equations for Scientific Machine Learning, arXiv [preprint], https://doi.org/10.48550/arXiv.2001.04385, 2021.

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

Rahmani, F., Lawson, K., Ouyang, W., Appling, A., Oliver, S., and Shen, C.: Exploring the exceptional performance of a deep learning stream temperature model and the value of streamflow data, Environ. Res. Lett., 16, 024025, https://doi.org/10.1088/1748-9326/abd501, 2021b.

Richards, L. A.: Capillary conduction of liquids through porous mediums, Physics, 1, 318–333, 1931.

Running, S., Mu, Q., and Zhao, M.: MOD16A2 MODIS/Terra Net Evapotranspiration 8-Day L4 Global 500 m SIN Grid V006, USGS [data set], https://doi.org/10.5067/MODIS/MOD16A2.006, 2017.

Sadourny, R.: The dynamics of finite-difference models of the shallow-water equations, J. Atmos. Sci., 32, 680–689, https://doi.org/10.1175/1520-0469(1975)032<0680:TDOFDM>2.0.CO;2, 1975.

Sarıgöl, M. and Katipoğlu, O. M.: Estimation of hourly flood hydrograph from daily flows using machine learning techniques in the Büyük Menderes River, Nat. Hazards, 119, 1461–1477, https://doi.org/10.1007/s11069-023-06156-x, 2023.

Schmidhuber, J.: Deep learning in neural networks: An overview, Neural Netw., 61, 85–117, https://doi.org/10.1016/j.neunet.2014.09.003, 2015.

Schneiderman, E. M., Steenhuis, T. S., Thongs, D. J., Easton, Z. M., Zion, M. S., Neal, A. L., Mendoza, G. F., and Todd Walter, M.: Incorporating variable source area hydrology into a curve-number-based watershed model, Hydrol. Process., 21, 3420–3430, https://doi.org/10.1002/hyp.6556, 2007.

Seibert, J. and Vis, M. J. P.: Teaching hydrological modeling with a user-friendly catchment-runoff-model software package, Hydrol. Earth Syst. Sci., 16, 3315–3325, https://doi.org/10.5194/hess-16-3315-2012, 2012.

Shen, C., Appling, A. P., Gentine, P., Bandai, T., Gupta, H., Tartakovsky, A., Baity-Jesi, M., Fenicia, F., Kifer, D., Li, L., Liu, X., Ren, W., Zheng, Y., Harman, C. J., Clark, M., Farthing, M., Feng, D., Kumar, P., Aboelyazeed, D., Rahmani, F., Song, Y., Beck, H. E., Bindas, T., Dwivedi, D., Fang, K., Höge, M., Rackauckas, C., Mohanty, B., Roy, T., Xu, C., and Lawson, K.: Differentiable modelling to unify machine learning and physical models for geosciences, Nat. Rev. Earth Environ., 4, 552–567, https://doi.org/10.1038/s43017-023-00450-9, 2023.

Sivapalan, M., Takeuchi, K., Franks, S. W., Gupta, V. K., Karambiri, H., Lakshmi, V., Liang, X., McDonnell, J. J., Mendiondo, E. M., O'Connell, P. E., Oki, T., Pomeroy, J. W., Schertzer, D., Uhlenbrook, S., and Zehe, E.: IAHS Decade on Predictions in Ungauged Basins (PUB), 2003–2012: Shaping an exciting future for the hydrological sciences, Hydrolog. Sci. J., 48, 857–880, https://doi.org/10.1623/hysj.48.6.857.51421, 2003.

Song, Y.: mhpi/HydroDLAdj: v1.0 (v1.0), Zenodo [code], https://doi.org/10.5281/zenodo.11205309, 2024a.

Song, Y.: HydroDLAdj, GitHub [code], https://github.com/mhpi/HydroDLAdj (last access: 11 July 2024), 2024b.

Thornton, M. M., Shrestha, R., Wei, Y., Thornton, P. E., Kao, S.-C., and Wilson, B. E.: Daymet: Daily Surface Weather Data on a 1-km Grid for North America, Version 4, ORNL DAAC, https://doi.org/10.3334/ORNLDAAC/1840, 2020.

Todd, D. K. and Mays, L. W.: Groundwater Hydrology, John Wiley & Sons, 663 pp., ISBN 10:0471059374, ISBN 13:978-0471059370, 2004.

Tsai, W.-P., Feng, D., Pan, M., Beck, H., Lawson, K., Yang, Y., Liu, J., and Shen, C.: From calibration to parameter learning: Harnessing the scaling effects of big data in geoscientific modeling, Nat. Commun., 12, 5988, https://doi.org/10.1038/s41467-021-26107-z, 2021.

Wang, R., Kim, J.-H., and Li, M.-H.: Predicting stream water quality under different urban development pattern scenarios with an interpretable machine learning approach, Sci. Total Environ., 761, 144057, https://doi.org/10.1016/j.scitotenv.2020.144057, 2021.

White, L. W., Vieux, B., Armand, D., and LeDimet, F. X.: Estimation of optimal parameters for a surface hydrology model, Adv. Water Resour., 26, 337–348, https://doi.org/10.1016/S0309-1708(02)00189-6, 2003.

Wu, P.: PyTorch 2.0: The Journey to Bringing Compiler Technologies to the Core of PyTorch (Keynote), in: Proceedings of the 21st ACM/IEEE International Symposium on Code Generation and Optimization, February 2023, New York, NY, USA, https://doi.org/10.1145/3579990.3583093, 2023.

Zeiler, M. D.: ADADELTA: An adaptive learning rate method, ArXiv [preprint], 1–6, https://doi.org/10.48550/arXiv.1212.5701, 2012.

Zhao, W. L., Gentine, P., Reichstein, M., Zhang, Y., Zhou, S., Wen, Y., Lin, C., Li, X., and Qiu, G. Y.: Physics-constrained machine learning of evapotranspiration, Geophys. Res. Lett., 46, 14496–14507, https://doi.org/10.1029/2019gl085291, 2019.

Zhi, W., Ouyang, W., Shen, C., and Li, L.: Temperature outweighs light and flow as the predominant driver of dissolved oxygen in US rivers, Nat. Water, 1, 249–260, https://doi.org/10.1038/s44221-023-00038-z, 2023.