



An argument for parsimony in differentiable hydrologic models

Sandeep Poudel¹, Scott Steinschneider²

¹Department of Civil and Environmental Engineering, Cornell University, United States

²Department of Biological and Environmental Engineering, Cornell University, United States

5 *Correspondence to:* Sandeep Poudel (sp2596@cornell.edu)

Abstract. Differentiable hydrologic models that use machine learning to infer parameters for process-based models show promise for both prediction and inference. However, these models are often developed with time-varying parameters, despite evidence that such flexibility can undermine physical consistency and yield only marginal predictive improvements over simpler static approaches. In this study, we revisit the comparison between static and dynamic differentiable models across 531 CAMELS-US basins, evaluating key architectural choices: (1) neural network type (multi-layer perceptron (MLP) vs. long short-term memory network (LSTM)); (2) process model configuration (single- versus ensemble-parameter estimation); and (3) comprehensive versus alternative input feature sets. Using the Hydrologiska Byråns Vattenbalansavdelning (HBV) conceptual model, we find that ensemble parameterizations consistently outperform single-parameter configurations, and that static, MLP-based ensembles achieve performance comparable to dynamic, LSTM-based ensembles despite their simpler structure. Additionally, we find that LSTM-estimated parameters rarely exhibit meaningful temporal variability despite their time-varying inputs, and when they do, this temporal variability may reflect hydrologic model equifinality rather than process dynamics. We further show that models using only latitude and longitude as static inputs achieve spatial generalization comparable to models using comprehensive feature sets describing climate, topography, geology, soils, and land cover. Similarly, temporal generalization is retained even when comprehensive features are replaced with physically meaningless values. This indicates static inputs are primarily used as spatial proxies to generalize in space and for site memorization when generalizing in time, rather than representations of physical basin processes. Overall, our results support reduced complexity in differentiable hydrologic modeling to provide greater transparency while retaining predictive performance.

1 Introduction

Hydrological models that simulate water fluxes and stores across the landscape are central to water resources management, infrastructure design, flood forecasting, and reservoir operations (Todini, 2007; Wagener et al., 2001). Process-based models have traditionally underpinned these applications, but machine learning models have recently surpassed them in predictive performance (Nearing et al., 2021). In particular, Long Short-Term Memory (LSTM) networks (Hochreiter & Schmidhuber, 1997) have proven highly effective due to their ability to represent long temporal dependencies and are often considered state-of-the-art for catchment-scale streamflow simulation (Feng et al., 2020; Kratzert et al., 2018). Despite their predictive



skill, the black-box nature of LSTMs and their lack of physical constraints raise concerns regarding their reliability for extrapolation tasks, such as extreme event prediction and climate change projections (Natel de Moura et al., 2022; Poudel et al., 2026; Razavi, 2021; Reichert et al., 2024; Wi & Steinschneider, 2022, 2024). These concerns are further compounded by saturation limits inherent to such models (Acuña Espinoza et al., 2025; Baste et al., 2025; Kratzert et al., 2024).
35 Consequently, hybrid approaches that integrate machine learning with process-based structure offer a promising pathway to retain predictive gains while preserving physical consistency (Bennett & Nijssen, 2021; Kraft et al., 2022; Reichstein et al., 2019; Shen et al., 2023).

Differentiable models represent a prominent hybrid approach in which neural networks learn parameters of process-based models within a unified, end-to-end training framework. These models can learn regionally coherent process-based model
40 parameters (Shen et al., 2023; Tsai et al., 2021) and have achieved predictive performance comparable to LSTMs in both gauged and ungauged basins (Feng et al., 2022, 2023). The most common differentiable architecture couples an LSTM with a lumped conceptual model, using the network to estimate either a single parameter set or an ensemble of parameter sets (Acuña Espinoza et al., 2025; Feng et al., 2022, 2023; Mangukiya & Sharma, 2025; Zhong et al., 2023). In the ensemble configuration, a single neural network produces multiple parameter sets, each of which is processed in parallel through the
45 process-based model to generate ensemble predictions; the mean of these predictions is commonly used as the final output. Depending on the network design and input features, these parameters may be inferred as static in time or allowed to vary dynamically at each simulation step.

Dynamic parameterizations provide additional flexibility and can yield modest performance improvements in many basins, but their advantages over simpler static formulations remain unclear. Empirical evidence suggests that gains are often
50 marginal (median NSE of 0.732 versus 0.714 in CAMELS-US; (Feng et al., 2022)) and that dynamic models can underperform static alternatives in certain hydroclimatic regimes, such as arid regions (Feng et al., 2024). From an interpretability perspective, static parameterizations are preferable because they yield a single, basin-specific parameter set that is directly comparable to conventional process-based models. Moreover, several studies caution that dynamically parameterized differentiable models may effectively function as LSTM variants despite their apparent physical structure,
55 thereby inheriting many of the same limitations (Acuña Espinoza et al., 2024; Feng et al., 2022; Acuña Espinoza et al., 2025).

In earlier work, Feng et al. (2022) hypothesized that overuse of dynamic parameterization risks compromising physical consistency for incremental performance gains. More recently, Acuña Espinoza et al. (2024) rigorously tested this by comparing three conceptual model structures coupled with LSTM-based dynamic parameterization: (a) a standard multi-
60 reservoir structure with realistic processes, (b) a single-bucket structure conserving only mass, and (c) a nonsensical structure with implausible processes. They found that all three hybrid configurations achieved performance comparable to a pure LSTM, suggesting that LSTM-based dynamic parameterization can compensate for missing or even implausible physical structural representation, providing the right answer for the wrong reasons (Kirchner, 2006). Nevertheless, dynamic



differentiable models dominate the current literature (Bohl et al., 2025; Ji et al., 2025; Mangukiya & Sharma, 2025; Zhong et al., 2023, 2024), while static approaches remain comparatively underexplored.

Interpretations of architectural trade-offs in differentiable modeling are further confounded by conceptual ambiguity in how the literature characterizes LSTM-based estimation of static and dynamic process-model parameters. Previous work has argued that LSTMs can provide both static and dynamic parameterization to process-based models. However, LSTMs are fundamentally dynamic models that operate in two modes: (a) sequence-to-sequence, processing an input sequence to produce an output at each time step, and (b) sequence-to-one, processing an input sequence to produce a single output from the final time step. In the differentiable modeling literature, these modes have been applied in three distinct ways. First, fully dynamic parameterization uses sequence-to-sequence LSTMs to vary all parameters at each simulation time step of the process-based model. Second, ‘static’ parameterization uses sequence-to-one LSTMs to produce parameters that vary between sequences during training, while during testing, the full input for a basin is supplied as a single sequence to get a single parameter set. Third, hybrid parameterization uses sequence-to-sequence LSTMs to vary selected parameters at each time step while treating others as ‘static’, similar to the second approach.

Critically, although the literature labels parameters as static in the second and third approaches, the LSTMs are actually learning parameters that vary across input sequences. A more appropriate architecture for truly static parameterization would be a Multi-Layer Perceptron (MLP), which takes static basin attributes as input and produces a single basin-specific parameter set, analogous to conventional process-based hydrological models. Despite this distinction, differentiable model comparisons have focused exclusively on LSTM formulations, excluding models that produce truly static parameterizations. Similarly, while sequence-to-sequence LSTMs have been criticized for violating physical constraints of process-based models (Acuña Espinoza et al., 2024, 2025), similar scrutiny has not been applied to sequence-to-one formulations.

Another argument for LSTM-based differentiable models is that they process time-varying inputs including precipitation and temperature to produce time-varying parameters, with parameter variability presumably indicative of seasonally varying hydrological processes. However, varying parameter sets are not necessarily representative of process dynamics but can merely arise due to equifinality (Beven, 2006), where multiple parameter sets produce comparable model performance. Thus, an alternative interpretation is that LSTMs are capturing equifinality manifesting as cyclical parameter changes through time, rather than meaningful seasonal dynamics. There is some evidence that supports the seasonal dynamics interpretation for sequence-to-sequence LSTMs. Feng et al. (2022) found that a time-varying runoff factor parameter followed a seasonal cycle mimicking basin water storage, which they hypothesized improved streamflow predictions by linking deep groundwater storage to runoff generation. However, whether temporal parameter variability in sequence-to-one LSTMs similarly represents seasonal process dynamics remains unclear.

Input feature selection represents another, closely related challenge in differentiable model development. The neural networks used for parameter estimation are flexible and can ingest many features. As a result, differentiable hydrological



models commonly adopt extensive sets of basin attributes - spanning meteorology, topography, geology, land cover, and soils - largely inherited from pure LSTM modeling frameworks (Feng et al., 2022; Kratzert et al., 2018). Recent work, however, questions whether neural networks meaningfully exploit such physical attributes for generalization (Heudorfer et al., 2024, 2025). Heudorfer et al. (2025) compared LSTMs provided with dynamic forcings and static attributes to ablated variants without attributes, finding that while static features improve temporal generalization, they contribute little to spatial generalization, which is primarily driven by meteorological forcings. Should similar limitations apply to parameter learning in differentiable models, extensive static inputs may only promote overfitting to training data without improving generalization.

Motivated by these knowledge gaps, this study contributes a systematic comparison of static and dynamic differentiable models, with focus on sequence-to-one LSTM formulations that have received less scrutiny than their sequence-to-sequence counterparts. We develop differentiable models for CAMELS-US (Addor et al., 2017) basins and evaluate their performance in gauged and ungauged settings based on choices in neural network architecture (MLP versus LSTM), process-based model configuration (single versus ensemble parameter sets), and input requirements (comprehensive versus alternative). We further investigate the temporal variability of LSTM-produced parameters, evaluate their value for prediction, and assess the contribution of dynamic inputs towards parameter learning. Collectively, these analyses are designed to address the following research questions:

1. How do neural network architecture (MLP versus LSTM) and process model configuration (single versus ensemble parameter sets) affect differentiable model performance in out-of-sample evaluations for both gauged and ungauged basins?
2. What static basin attributes are necessary for robust and transferable parameter learning in differentiable models, and how does performance change when using alternative versus comprehensive static inputs?
3. To what extent do dynamic inputs meaningfully contribute to parameter learning or predictive skill in differentiable hydrologic models?

2 Data and Methods

This work compares multiple differentiable model architectures, investigating single and ensemble configurations combined with either MLP or LSTM networks for parameter learning. We also evaluate performance when the networks use comprehensive and alternative static feature sets to determine the necessary inputs for effective parameter learning. Finally, for the LSTM-based differentiable models, we examine the nature of learned process-model parameters and assess how dynamic input features contribute to their estimation. These experiments are described in more detail below.



2.1 Study Area and Data

This study examines 531 basins from the CAMELS-US dataset (Fig. A1), filtered by Newman et al. (2017) based on basin size and catchment area reliability. The dataset contains a large set of static input features, spanning meteorology, topography, geology, land cover, and soils data (see Table 1). Daily dynamic inputs are taken from the Daymet meteorological product between 1980-2014, and daily USGS streamflow for each basin are available for the same period.

For all experiments, we randomly hold out 20% of basins for spatial (ungauged basin) evaluation (see Fig. A1). For the remaining 80% of basins, we use 16 years (1990-2005) for model training, 8 years (2006-2014) for validation, and 10 years (1980-1989) for testing.

Table 1: Full set of input features used by MLP and LSTM models in differentiable architecture. Dynamic features are only used by the LSTM.

Category	Name	Description
Dynamic	Precipitation	Daily total precipitation
	Temperature	Daily average temperature
	Daylength	Length of day
Static	P_mean	Mean daily precipitation
	PET_mean	Mean daily PET
	Aridity	Ratio of PET to precipitation
	P_seasonality	Seasonality and timing of precipitation
	Frac_snow	Fraction of precipitation falling as snow
	High_prec_freq	Frequency of high precipitation days
	High_prec_dur	Average duration of high precipitation events
	Low_prec_freq	Frequency of dry days
	Low_prec_dur	Average duration of dry days
	Elev_mean	Mean elevation of basin
	Slope_mean	Mean slope of basin
	Area_gages2	Basin area from GAGESII
	Frac_forest	Fraction of forest
	Lai_max	Maximum monthly mean of the leaf area index
	Lai_diff	Difference between the maximum and minimum monthly mean of the leaf area index
	Gvf_max	Maximum monthly mean of the green vegetation
	Gvf_diff	Difference between maximum and minimum monthly mean of the green vegetation



		fraction
	Dom_land_cover_frac	Fraction of the basin area with dominant land cover
	Soil_depth_pelletier	Depth to bedrock
	Soil_depth_statsgo	Soil depth
	Soil_porosity	Volumetric porosity of soil
	Soil_conductivity	Saturated hydraulic conductivity
	Max_water_content	Maximum water content
	Sand_frac	Fraction of sand
	Silt_frac	Fraction of silt
	Clay_frac	Fraction of clay
	Glim_1st_class_frac	Fraction of basin area with first most common geologic class
	Glim_2nd_class_frac	Fraction of basin area with second most common geologic class
	Carbonate_rocks_frac	Fraction of basin area as carbonate sedimentary rocks
	Geol_p permeability	Permeability of subsurface

2.2 Process-based hydrological model

We use the Hydrologiska Byråns Vattenbalansavdelning (HBV) model (Bergström, 1995; Seibert, 2005) as the process-based hydrological model throughout this work. HBV represents watershed processes through 20 parameters distributed across three interconnected modules: snow accumulation and melt, soil moisture and evapotranspiration, and streamflow generation and routing. The model first partitions precipitation into rainfall and snowfall based on temperature thresholds, with snowmelt calculated using a degree-day method. In the soil and evapotranspiration module, actual evapotranspiration is calculated from potential evapotranspiration and available soil moisture. Excess water beyond the soil's storage capacity enters linear reservoirs representing quick flow, delayed subsurface flow, and groundwater discharge. These flow components are aggregated and routed to the catchment outlet through a gamma unit hydrograph function. HBV takes daily series of total precipitation and average temperature as inputs and uses daylength to calculate potential evapotranspiration based on the Hamon method (Hamon, 1961). Additional details on model parameters and process representations are provided in Appendix B (Table B1).

2.3 Differentiable model architectures

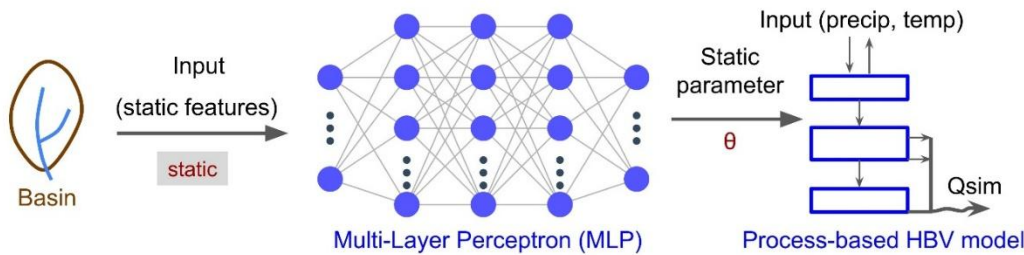
Differentiable hydrologic models are developed in a unified, end-to-end framework. A neural network processes static and/or dynamic features to infer parameters of the HBV model, which then uses these parameters together with



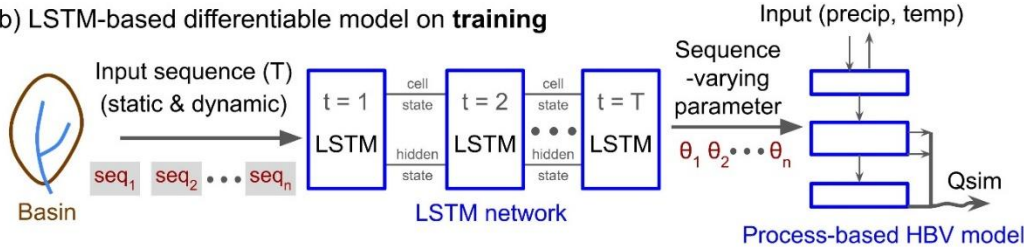
meteorological forcings to simulate streamflow. Neural network inputs are standardized to zero mean and unit variance, whereas HBV inputs are provided in their original units (mm/day for precipitation, °C for temperature).

We consider four differentiable architectures: MLP with a single HBV unit (MLP+HBV), MLP with a 16-unit HBV ensemble (MLP+16HBV), LSTM with a single HBV unit (LSTM+HBV), and LSTM with a 16-unit HBV ensemble (LSTM+16HBV). The ensemble size of 16 follows recommendation from prior work (Feng et al., 2022). These configurations are described below.

a) MLP-based differentiable model on **training & inference**



b) LSTM-based differentiable model on **training**



c) LSTM-based differentiable model on **inference**

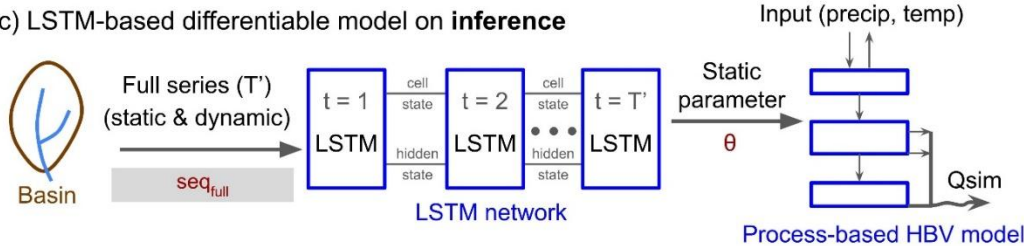


Figure 1: Schematic of differentiable model architecture. (a) MLP+HBV: Static basin attributes are passed through an MLP network to generate a static HBV parameter (θ), which are then used by the HBV model to simulate streamflow. This process is identical for training and inference/testing. (b) LSTM+HBV during training: Sequences of concatenated static and dynamic features are processed through the LSTM, with the final hidden state producing HBV parameters for streamflow simulation. Multiple input sequences ($seq_1, seq_2, \dots, seq_n$) per basin result in sequence-varying parameters ($\theta_1, \theta_2, \dots, \theta_n$) during training. (c) LSTM+HBV during inference: The entire available time series for a basin (seq_{full}) is processed as one continuous sequence, with the final hidden state producing a single set of static HBV parameters (θ) for streamflow simulation. In all cases, the HBV model receives precipitation and temperature as forcing inputs. The schematic illustrates for a single basin, though models are trained regionally across all basins.



2.3.1 MLP+HBV and MLP+16HBV

The first network architecture considered is a Multi-Layer Perceptron (MLP), a feedforward neural network composed of multiple fully connected layers that transform inputs through successive nonlinear activation functions. In the MLP+HBV architecture (Fig. 1a), the MLP takes 30 static basin attributes as input - including measures of climate, topography, geology, land cover, and soil type (see Table 1) - and outputs a single vector of HBV parameters per basin. The MLP uses ReLU activations in all hidden layers and a linear activation in the output layer. The resulting outputs are then normalized between [0,1] using sigmoid activation and scaled to parameter-specific minimum and maximum ranges for the HBV model (Table B1).

Each training instance consists of a 3-year sequence of daily precipitation and temperature forcings ($\mathbb{R}^{1095 \times 2}$) for the HBV model, along with a standardized vector of static basin attributes ($\mathbb{R}^{30}$) for the MLP. The static attributes are first passed through the MLP to produce a single set of 20 HBV parameters ($\mathbb{R}^{20}$). These parameters, together with the 3-year forcing sequence, are then passed to the HBV model to simulate streamflow over the full 3-year period ($\mathbb{R}^{1095}$). The first 2 years of streamflow simulation are discarded for spin-up, and the loss is computed using only the final year. Gradients of the loss are then backpropagated through the HBV model to update MLP weights.

During inference, each basin's static attribute set ($\mathbb{R}^{30}$) and the complete precipitation and temperature forcing record are provided to the hybrid model in a single forward pass. For example, given 15 years (5475 days) of forcing data for a basin, the static attributes are first passed through the MLP to obtain a fixed set of HBV parameters ($\mathbb{R}^{20}$), which are then used together with the entire forcing sequence ($\mathbb{R}^{5475 \times 2}$) in the HBV model to simulate streamflow for the entire period.

We also consider an ensemble architecture called MLP+16HBV, which is identical to MLP+HBV except that the MLP outputs an ensemble of 16 parameter sets ($\mathbb{R}^{320}$). Each parameter set is run through independent HBV units that receive the same meteorological forcings, producing 16 parallel streamflow predictions at each timestep. These ensemble members are averaged to produce the final streamflow prediction (Fig. 2).

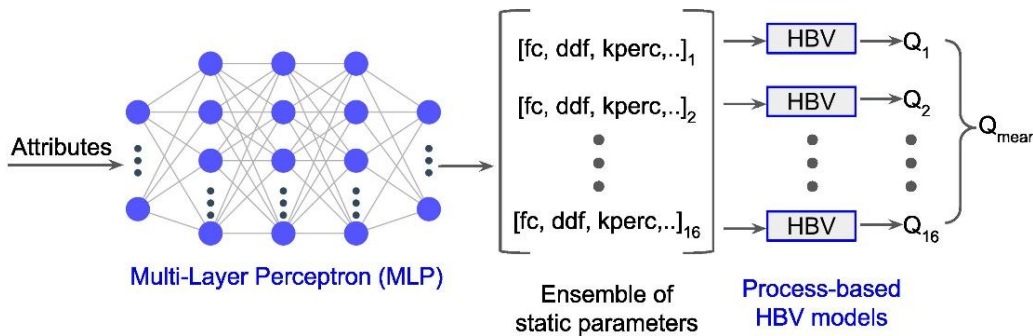


Figure 2: Schematic of differentiable MLP with ensemble HBV architecture with 16 independent modules.



2.3.2 LSTM+HBV and LSTM+16HBV

The second network architecture uses an LSTM to estimate HBV parameters. In the LSTM+HBV architecture (Fig. 1b-c), the LSTM takes as input the same 30 static features used in the MLP, along with 3 dynamic features - precipitation, temperature, and daylength (see Table 1). Static features are repeated at every timestep within the LSTM lookback period, which is fixed at 365 days in this study. The LSTM is developed in a sequence-to-one configuration and uses linear activations in the output layer. Again, the resulting outputs are normalized between [0,1] using sigmoid activation and scaled to parameter-specific minimum and maximum ranges for the HBV model (Table B1).

During training (Fig. 1b), each instance consists of: (1) a 3-year sequence of precipitation and temperature forcings for the HBV model ($\mathbb{R}^{1095 \times 2}$); and (2) a 1-year concatenated sequence of static and dynamic input features from the most recent year of the 3-year period ($\mathbb{R}^{365 \times 33}$). This 1-year input sequence is passed through the LSTM, which transforms the inputs over the lookback period into a sequence of hidden states of equal length. In the sequence-to-one configuration adopted here, only the hidden state at the final timestep is extracted to produce static parameters for the HBV model ($\mathbb{R}^{20}$). As in the MLP+HBV model, these parameters, together with the 3-year forcing sequence, are then passed to the HBV model to simulate streamflow, and output from only the last of the three years is used to calculate the loss and update the LSTM weights.

During training of the LSTM+HBV model, HBV parameters are static for each input sequence, but they vary across different input sequences in the training period (denoted $\text{seq}_1, \dots, \text{seq}_n$ in Fig. 1b). To obtain a static HBV parameter set during inference (i.e., the test set; Fig. 1c), the entire input sequence for a basin (denoted seq_{full} in Fig. 1c) is passed through the LSTM in a single forward pass, and only the final hidden state is used to estimate the HBV parameters. For example, with 15 years of data, the full 15-year concatenated input sequence ($\mathbb{R}^{5475 \times 33}$) is processed by the LSTM, and the final hidden state (at the end of year 15) is mapped to a single parameter vector ($\mathbb{R}^{20}$), ensuring one static parameter set per basin. These parameters are then used with the full forcing series ($\mathbb{R}^{5475 \times 2}$) in the HBV model to generate streamflow simulations for the entire period. This approach is taken to enforce a single, basin-specific static parameter set for the entire test period.

The training and testing approach described above is similar to what the literature calls ‘static’ parameterization (Feng et al., 2022, 2023, 2024), even though sequence-to-one LSTMs produce parameters that vary across sequences during training. Another common approach – though not one taken here – is hybrid parameterization, where most parameters are treated as ‘static’ (i.e., estimated based on the last time step of each sequence), while a subset is allowed to vary dynamically (i.e., estimated for all time steps in the LSTM’s lookback period based on a sequence-to-sequence formulation) (Acuña Espinoza et al., 2025; Feng et al., 2022, 2023, 2024). However, even under this hybrid parameterization scheme, parameters labeled as ‘static’ still vary across training sequences.



Analogous to the ensemble MLP configuration (Fig. 2), the LSTM+16HBV architecture extends the LSTM+HBV model by producing an ensemble of 16 HBV parameter sets ($\mathbb{R}^{320}$). Each parameter set is passed to an independent HBV model, and the resulting ensemble of streamflow simulations is averaged to produce the final prediction.

2.3.3 Model training and hyperparameter tuning

For all model configurations, loss is calculated between predicted and observed streamflow (in mm/day) using mean square error (MSE), and neural network weights are updated via backpropagation using the ADAM optimizer (Kingma & Ba, 2017). We initialize training with a learning rate of 0.001, reduced by 90% if validation loss does not improve for 5 consecutive epochs. Training is continued until validation loss shows no improvement for 10 consecutive epochs. A dropout rate of 40% is applied to MLP hidden layers and to the final linear layer of the LSTM. Models are trained with a batch size of 128 sequences, and input sequences are created with a stride length of 60 days. For example, if one sequence consists of a 3-year time series input, the subsequent sequence is generated by shifting the start of the previous sequence forward by 60 days. This approach results in overlapping data sequences and showed a modest improvement in validation set performance.

Our choices for hyperparameters are similar to previous studies (Feng et. al, 2022; Acuña Espinoza et al., 2024), with additional control over the total number of epochs and learning rate through early stopping based on validation performance. We performed grid search to optimize neural network sizes for each configuration, selecting the architecture that achieves the lowest validation set error. Details of the hyperparameter grid search are provided in Table C1, Appendix C.

2.4 Impacts of comprehensive versus alternative static input features

In differentiable models, the static inputs to the neural networks often consist of the same diverse set of catchment attributes - encompassing climate, topography, geology, land cover, and soil type - commonly used by pure LSTM models. The implicit assumption is that these basin attributes provide informative context for neural network-based parameter learning. To test this assumption, we evaluate the role of static features through two experiments that use alternative static feature sets relative to the reference case where models use the full set.

In the first experiment, the neural network receives only basin latitude and longitude as static input. This represents a scenario where models are deprived of all physical basin attributes and provided only with direct positional information.

In the second experiment, we use the same full static feature set as in the reference case, but we randomly shuffle the attribute vectors across all basins without replacement. This experiment represents a scenario where the neural network still receives comprehensive attribute features, but the values carry no physical information.

We then compare differentiable model simulations for out-of-sample evaluations in both time and space using these alternative static feature sets against the reference case trained with the full feature set. Note that while the MLP architecture



uses only static features, the LSTM architecture also incorporates three dynamic features (precipitation, temperature, and daylength), which are held constant across all experiments.

2.5 Assessing temporal variability in LSTM-estimated parameters and the role of dynamic features

255 As discussed previously, the MLP architecture relies exclusively on static inputs and therefore produces a single, time-invariant set of parameters for each basin, analogous to conventional process-based models. In contrast, the sequence-to-one LSTM architecture generates sequence-dependent parameter sets during training and produces a single static parameter only during inference. This distinction motivates a closer examination of the extent to which the LSTM-based architecture introduces meaningful temporal variability into process-model parameters, and whether such variability is beneficial for
260 streamflow prediction.

To quantify this behavior, we compute the coefficient of variation (standard deviation divided by mean) for each HBV parameter across training input sequences for each study basin. Two temperature-related parameters are excluded from this analysis because their mean values are close to zero, which would artificially inflate their coefficients of variation. For each basin, we then calculate the mean coefficient of variation across the remaining HBV parameters, yielding a single summary
265 metric that characterizes the overall degree of temporal variability in LSTM-estimated parameters. We select representative basins exhibiting high, medium, and low variability and examine the temporal variation of individual HBV parameters in detail.

To understanding how dynamic inputs contribute to HBV parameter learning for these basins, we use an explainable AI technique known as Integrated Gradients (IG) (Sundararajan et al., 2017). We apply IG to the LSTM-based differentiable
270 model, compute the mean of the absolute IG values for each input feature, and normalize the results such that the contributions across features sum to 100%, providing an interpretable measure of the relative importance of static and dynamic inputs in HBV parameter learning. More details are provided in Appendix D.

3. Result

3.1 Differentiable model performance in out-of-sample in time and space evaluation

275 Hyperparameter tuning detailed in Appendix C (Table C1), resulted in the following optimal neural network sizes:

1. MLP+HBV: 3 layers, each with 2048 nodes
2. MLP+16HBV: 3 layers, each with 2048 nodes
3. LSTM+HBV: 1 layer with 512 nodes
4. LSTM+16HBV: 1 layer with 1024 nodes

280

We used these optimized architectures for all four differentiable models and evaluated their temporal and spatial out-of-sample performance.

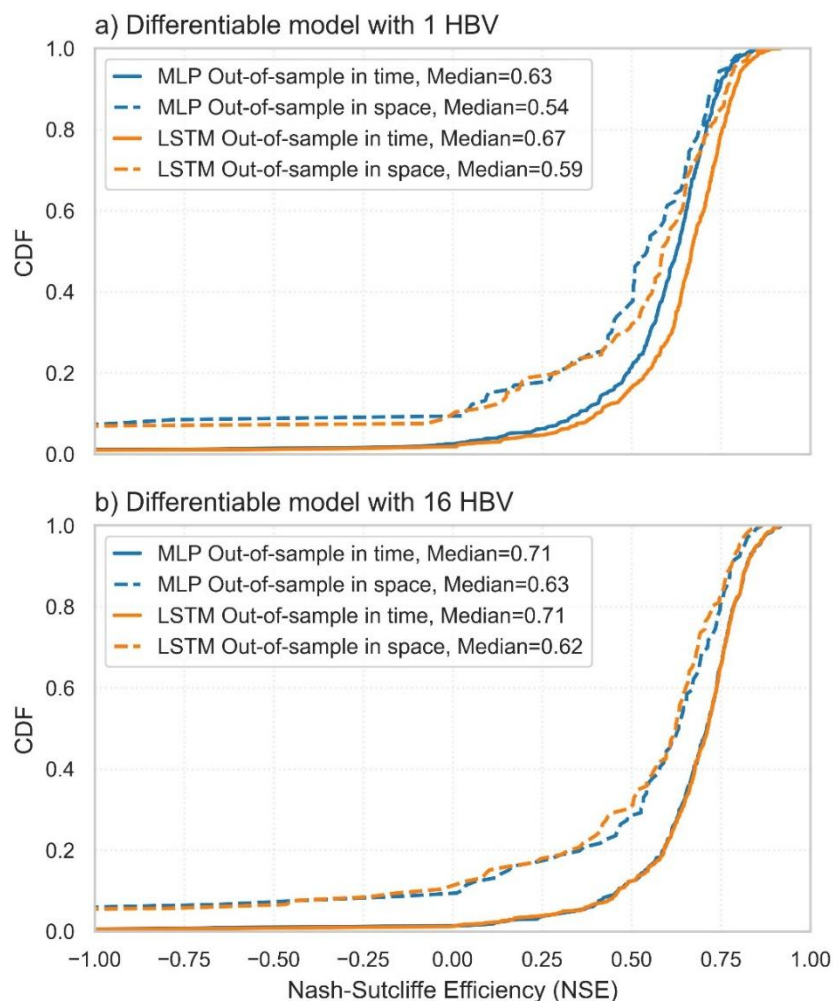


Figure 3: Cumulative distributions of Nash-Sutcliffe Efficiency (NSE) for out-of-sample in time and out-of-sample in space evaluations of MLP and LSTM-based differentiable model architecture using: (a) single-member HBV and (b) ensemble (16-member) HBV.

Figure 3 shows the distribution of Nash-Sutcliffe Efficiency (NSE) across in-sample gauges (solid lines) and out-of-sample gauges (dashed lines) for both single-unit (Fig. 3a) and ensemble-unit (Fig. 3b) model versions coupled with MLP (blue) and LSTM (orange) architectures. Several key findings emerge from this figure.

For single-unit HBV configurations (Fig. 3a), the LSTM-based architecture outperforms the MLP-based architecture in both out-of-sample in time and out-of-sample in space evaluations. Median NSE values increase from 0.63 to 0.67 for out-of-sample in time testing and from 0.54 to 0.59 for out-of-sample in space testing when using the LSTM instead of the MLP.

Introducing ensemble configurations (Fig. 3b) leads to substantial performance gains relative to single-unit models (Fig. 3a) for both neural network architectures and both evaluation settings. However, under the ensemble configuration, the



performance advantage of the LSTM over the MLP disappears. The cumulative NSE distributions for the two architectures are nearly indistinguishable, with identical median NSE values of 0.71 for out-of-sample in time evaluation and median values of 0.62 and 0.63 for LSTM- and MLP-based models, respectively, in out-of-sample in space evaluation.

Finally, we note that both ensemble-based architectures achieve the same benchmark performance of the ensemble, differentiable model applied to CAMELS-US in Feng et al. (2022), which was reported as having a median NSE of 0.714 in out-of-sample in time testing.

We also evaluated the ensemble-based differentiable models across additional performance metrics, including the Kling-Gupta Efficiency (KGE), root mean square error (RMSE), percent bias, and flow-regime specific biases for high flows (top 10% of observed flows), medium flows (30th to 90th percentiles), and low flows (bottom 30% of observed flows) (Fig. 4). Across all metrics and flow regimes, the LSTM- and MLP-based architectures performed similarly in an out-of-sample in space evaluation. Taken together, these results suggest that ensemble-based parameter estimation provides clear benefits for differentiable hydrological modelling, while the added complexity of LSTM-based networks offers limited advantage over the simpler MLP-based approach.

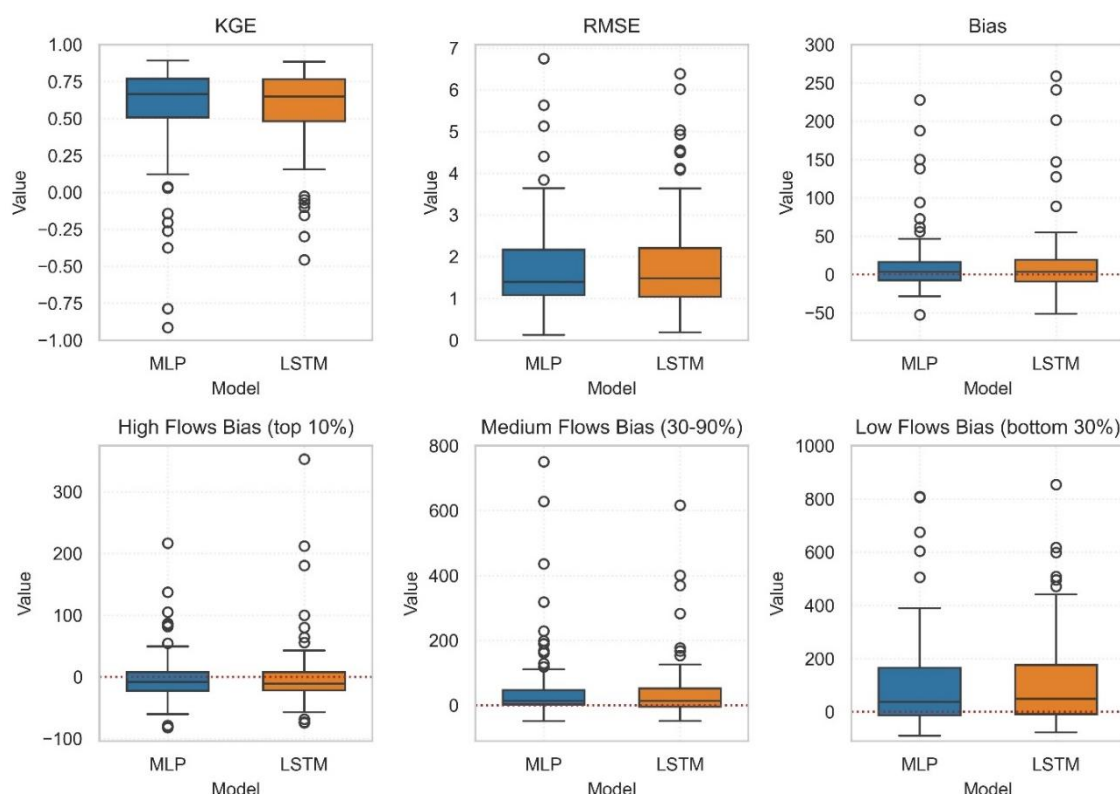


Figure 4: Performance of differentiable, ensemble-unit models with MLP- and LSTM-based architecture across multiple performance metrics for out-of-sample in space evaluation.



3.2 Impacts of alternative static input features

Results in Figs. 3 and 4 are based on models trained with the full set of reference static input features. Figure 5 shows performance for ensemble-based models trained with two alternative static feature sets, the first of which only includes basin latitude and longitude and the second including the same comprehensive reference features but randomly mixed across
315 basins. This experiment was performed only with ensemble-based models because of their demonstrated outperformance compared to single-unit models.

For the first experiment where only latitude and longitude were used as static input (Fig. 5a-b), we find similar NSE distributions compared to the reference input set for out-of-sample in space evaluations, both for MLP- (median NSE of 0.62 versus 0.63) and LSTM- (median NSE of 0.62 for both) based differentiable models. However, for out-of-sample in time
320 evaluations, both differentiable models using only latitude and longitude showed substantial performance reductions, with a median NSE of 0.64 for the MLP-based model and 0.60 for LSTM-based model, as compared to a reference of 0.71 for both.

For the second experiment using full but nonsensical static inputs (Fig. 5c-d), we find little difference in the NSE distributions compared to the reference for out-of-sample in time evaluation, both for the MLP- (median NSE of 0.68 versus 0.71) and LSTM- (median NSE of 0.69 versus 0.71) based models. However, for out-of-sample in space evaluations, both
325 differentiable models showed large degradation in performance, with a median NSE value of 0.47 against a reference of 0.63 for the MLP-based model, and a median value of 0.42 against a reference of 0.62 for the LSTM-based model.

Overall, we find that the spatial generalization of differentiable models is preserved even when only latitude and longitude are used as static inputs, suggesting that neural networks use comprehensive basin attributes primarily as spatial proxies rather than learning meaningful physical relationships. However, temporal generalization degrades in this case because the
330 2-dimensional input vector of latitude and longitude lacks sufficient complexity for the model to ‘memorize’ unique behavior to individual training basins. This interpretation is further supported by the shuffled feature experiment, where providing comprehensive but nonsensical feature sets enable near-reference performance at the training basins but precludes the ability to generalize spatially.

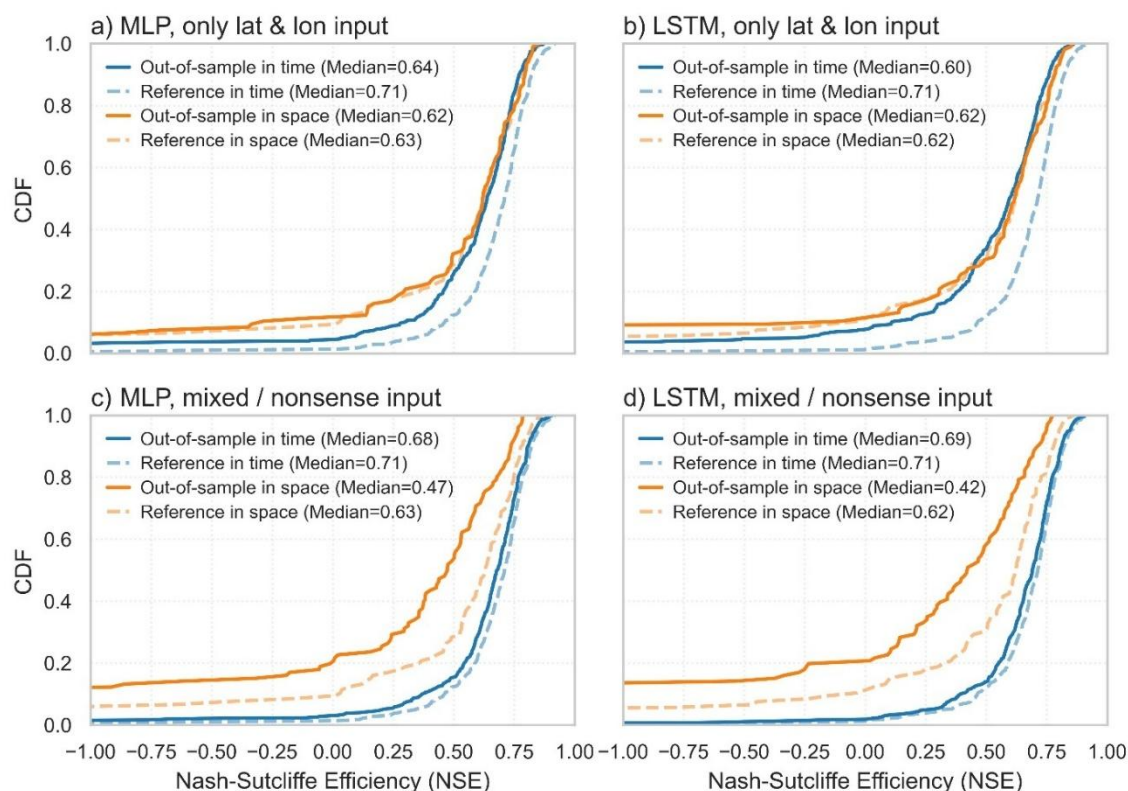


Figure 5: Cumulative distributions of NSE for out-of-sample in time and space evaluations of ensemble HBV differentiable models with (a) MLP using only latitude and longitude as static inputs, (b) LSTM using only latitude and longitude as static inputs, (c) MLP using comprehensive but randomly perturbed static features, and (d) LSTM using comprehensive but randomly perturbed static features. Dotted lines indicate reference performance using the comprehensive static feature set.

3.3 Assessing temporal variability in LSTM-estimated parameters and the role of dynamic features

Here we examine the nature of parameter estimates by the LSTM+HBV differentiable model, with particular emphasis on their temporal stability and the influence of dynamic inputs. To simplify interpretation, we focus on the single-unit LSTM+HBV configuration rather than the ensemble-based LSTM+HBV16 model.

Figure 6 summarizes the average coefficient of variation of HBV parameters across input sequences for each study basin. Most basins exhibit low variability, suggesting that the LSTM-based architecture does not utilize dynamic features to vary parameter estimates across sequences. There are some basins with a higher degree of parameter variability across training sequences, particularly in the central US. However, the average coefficient of variation across parameters never exceeds 0.2 for any site.

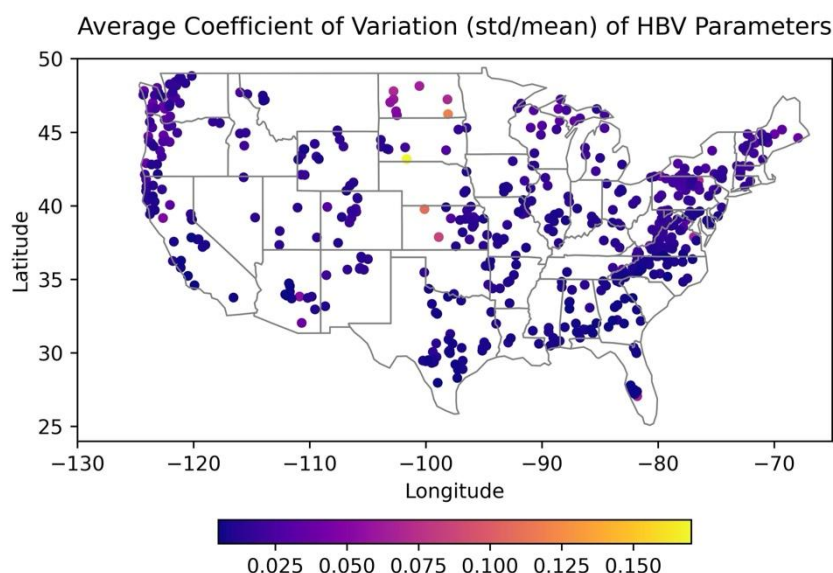


Figure 6: Spatial distribution of the average coefficient of variation of HBV parameters between input sequences estimated by the LSTM+HBV differentiable model. Each dot represents a study basin, with color intensity indicating the magnitude of the average temporal variability of the parameter set.

To further examine this behavior, we focus on three representative basins with high, medium, and low parameter variability (USGS IDs 01532000, 10336660, 02465493). Although the HBV model has 20 parameters, we present results for four illustrative parameters: field capacity (FC), coefficient of potential evapotranspiration (Coeff PET), coefficient of percolation (Kperc), and degree-day factor (DDF). Figure 7 shows the temporal variability of these parameters across input sequences for basins with (a) high, (b) medium, and (c) low variability.

In the low-variability basin, parameter estimates show negligible temporal variability across sequences, indicating that the LSTM effectively produces static parameters. In contrast, the medium- and high-variability basins exhibit progressively more parameter variability, although to varying degrees for different parameters. For example, in the high-variability basin, DDF exhibits comparatively limited temporal variation relative to the other three parameters. Additionally, several parameters tend to co-vary: FC, Coeff PET, and Kperc follow correlated seasonal cycles, reaching maxima and minima simultaneously, whereas DDF tends to vary out of phase, peaking when the other parameters are at their minima. Similar, though less pronounced, patterns are observed in the medium-variability basin. Feature importance based on Integrated Gradients shows that the differentiable model consistently relies more on static features than dynamic features (precipitation, temperature, and daylength) for HBV parameter estimation, even in the basin with high parameter variability (Fig. D1 in Appendix D), consistent with the limited variability in parameter values across sequences.

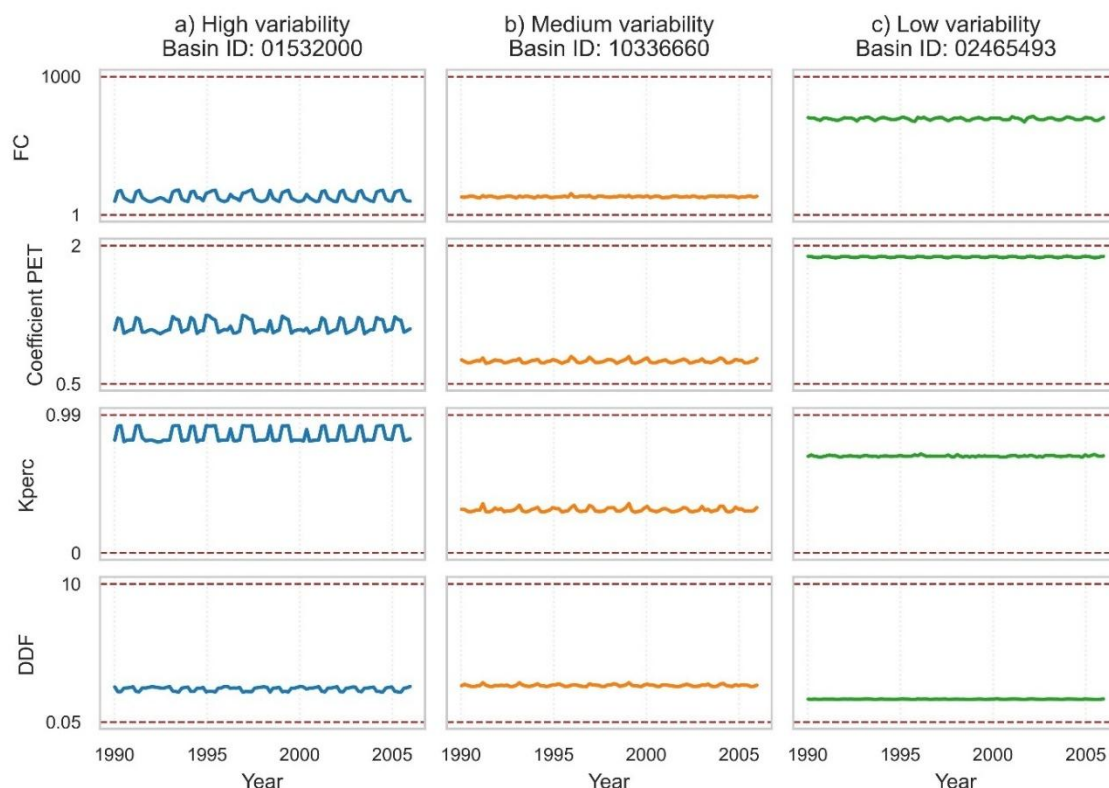


Figure 7: Temporal variability of four selected HBV parameters inferred by the LSTM+HBV differentiable model for basins exhibiting (a) high, (b) medium, and (c) low average coefficients of variation. The red dashed line shows the lower and upper bound for each parameter.

It is worthwhile to ask how the temporal variability in LSTM-based parameter estimates, where present, influences streamflow simulation. To investigate this, we compared two approaches to HBV simulation. In the first approach, model simulations were based on a single, static parameter set extracted from the final hidden state of the LSTM after processing the full input sequence for the test period. This is the reference case and is the same as shown in Fig. 1c. In the second approach, simulations are based on sequence-varying parameters, i.e., running the LSTM on rolling sequences during the test period. With a stride of 60 days, the input sequence shifts forward by 60 days at each step, generating new parameters every 60 days throughout the test period. This approach mirrors the training procedure as shown in Fig. 1b and represents a more natural inference method for sequence-to-one LSTMs. Thus, the first approach runs HBV with one parameter set for the entire test period (dashed lines in Fig. 8), while the second uses time-varying parameters that update every 60 days (solid lines in Fig. 8).

Figure 8a presents the distribution of NSE values across all basins for these two cases, while Fig. 8b focuses exclusively on the 20 basins exhibiting the highest degree of temporal parameter variability. In both cases, results show little difference in performance across basins. There are some small differences in the out-of-sample in space evaluation for basins with high



temporal parameter variability (Fig. 8b), but it is the reference case (a single static parameter set) that outperforms the time-varying parameter models in this situation.

Together, these findings suggest that although the LSTM produces sequence-varying parameter estimates, exploiting this variability during inference does not yield meaningful performance improvements relative to using a single static parameter set derived at the end of the full forcing test sequence. Consequently, the observed temporal variability in LSTM-predicted parameters may be better interpreted as a manifestation of equifinality rather than evidence of dynamically evolving process parameters.

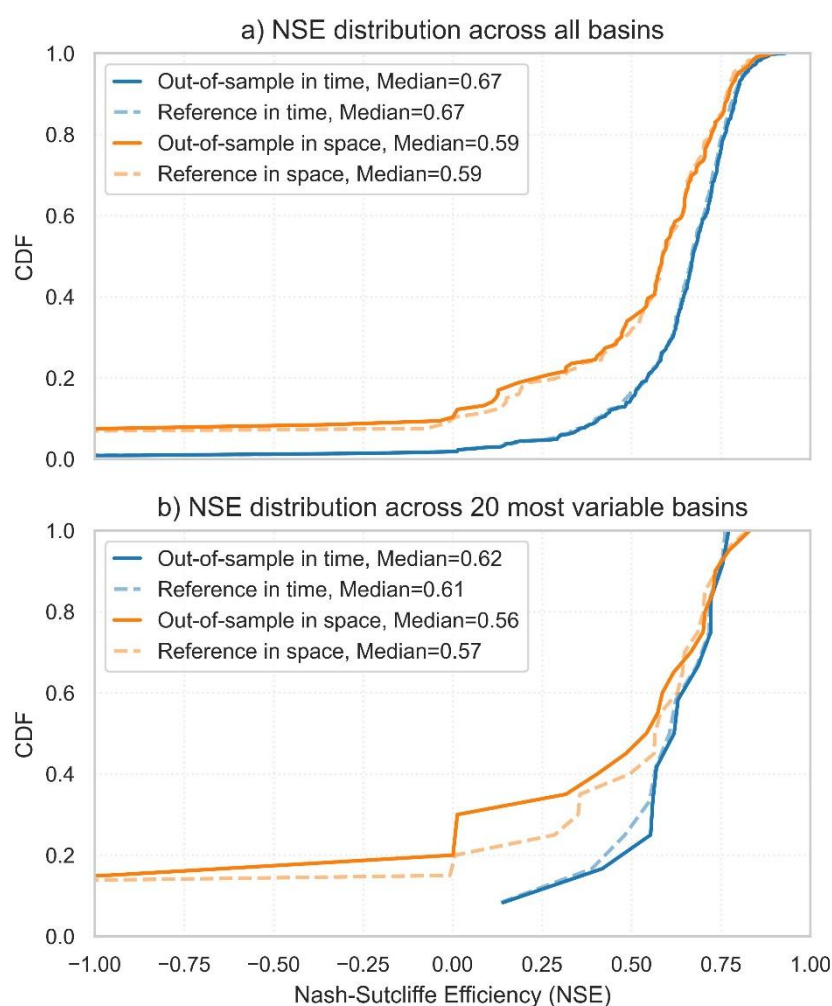


Figure 8: Cumulative distributions of NSE for out-of-sample in time and out-of-sample in space evaluations for the differentiable LSTM+HBV model using sequence-varying parameters for (a) all study basins, and (b) the 20 basins with highest parameter variability. Dotted lines indicate reference performance using a single static parameter from the final LSTM time step.



4. Discussion and Conclusion

400 Differentiable hybrid models that use machine learning to estimate parameters for process-based models show strong potential for achieving state-of-the-art performance while maintaining transparency of internal hydrological processes (Feng et al., 2022, 2023). Their rapid adoption reflects this promise. However, several aspects of differentiable model development haven't received sufficient scrutiny, risking a shift toward prediction-driven modeling that undermines transparency and contradicts the philosophy of physics-based machine learning (Karpatne et al., 2017). This study addresses several such concerns by revisiting static and dynamic differentiable models and investigating key architectural decisions including neural network choices, process-based model configuration, and input requirements. We evaluate these models based on out-of-sample performance as well as interpretability. Our main conclusions are as follows:

1. Ensemble models outperform single-parameter configurations, and a simpler MLP-based static parameterization matches the performance of a dynamic LSTM-based parametrization.

410 Differentiable models with ensemble configurations consistently outperformed single-unit models across both temporal and spatial test cases, consistent with previous findings (Feng et al., 2022, 2023). This result is not unexpected and aligns with the multi-parameter ensemble literature that shows improved prediction accuracy by representing parametric uncertainty through ensemble aggregation (Beven & Binley, 1992; Blasone et al., 2008; Chiang et al., 2017; Shin et al., 2023).

415 More importantly, simpler MLP-based differentiable models matched sequence-to-one LSTM-based model performance. MLP-based models achieved a median NSE of 0.71 in temporal testing for CAMELS-US basins, comparable to sequence-to-one LSTMs (0.714) and sequence-to-sequence LSTMs (0.732) reported by Feng et al. (2022). Similar results are found for spatial generalization. In addition, MLPs offer greater transparency by producing single, basin-specific process model parameters, which is not available with LSTM-based formulations. This provides a strong argument for the use of more parsimonious, MLP-based architectures in future differentiable hydrological modeling.

2. Differentiable models use static basin attributes primarily as spatial proxies and for memorizing unique behavior at training sites, rather than learning physically meaningful processes.

Recent work by Heudorfer et al. (2025) suggested that pure LSTM models may not be as entity aware as previously assumed, indicating that basin attributes provide limited information beyond meteorological forcings. We extend this insight to differentiable hydrological models through two experiments that directly examine how neural networks use static basin attributes.

In the first experiment, we compared models using comprehensive basin attributes (climate, topography, geology, land cover, and soils) with models using only basin latitude and longitude as input. Spatial generalization did not degrade



when only positional inputs were used, indicating that comprehensive attributes are primarily used to encode basin location and serve as spatial proxies. This is somewhat expected in regional settings where similar basins are geographically clustered. When models learn this spatial structure, they can generalize in space by placing new basins within an internally learned spatial representation. Consequently, for regions with dense and representative basin coverage such as CAMELS-US, latitude and longitude alone are sufficient for spatial generalization. Importantly, positional information (like latitude and longitude) cannot support transferability beyond the spatial domain represented in the training data, thus limiting transfer learning for new regions (e.g., other continents). Given the results presented here and in past work (Heudorfer et al., 2025), it remains unclear the degree to which comprehensive static basin attributes truly support generalized mappings between basin attributes and their internal representations, which is required for robust transfer learning across regions. Additional research is needed in this area.

In contrast, temporal generalization degraded when only latitude and longitude were used. We hypothesize that in this case, using minimal (low-dimensional) inputs reduces the neural network's ability to differentiate individual basins. Because regional models are trained across basins simultaneously, distinguishing basins is necessary to learn basin-specific input-output relationships. To test this, we conducted a second experiment using the same comprehensive attribute vectors but with physically meaningless values, preserving input dimensionality without providing any physical information. Temporal generalization in this case matched performance obtained using physically meaningful attributes, supporting our hypothesis.

These results indicate that differentiable models may not be learning meaningful relationships with basin static attributes, or at least, they are only using them as spatial proxies to generalize in space and for site memorization to generalize in time. This finding underscores the need for methods that better leverage physiographic data to improve higher-order model generalization, particularly for extrapolating to conditions involving structural changes such as climate change and land-use modification (Heudorfer et al., 2025; Gupta, 2026).

3. LSTMs do not support true static parameterization. In sequence-to-one formulations, dynamic inputs contribute minimally, and time-varying parameters reflect equifinality rather than seasonal dynamics.

LSTMs operate in either sequence-to-sequence or sequence-to-one modes, generating parameters that vary at each timestep or across input sequences, respectively. In sequence-to-one implementations, the full input record for a basin can be used as a single sequence during testing, producing a single basin-specific parameter set. This is commonly labeled as a static parameterization in the literature (Feng et al., 2022, 2023), despite being trained with parameters that vary between input sequences. We evaluated how sequence-varying parameters influence streamflow simulations by comparing them with simulations generated using a single parameter set derived from the final timestep of the full input sequence. We didn't find any meaningful differences between the two cases, suggesting that temporal variability in parameters with sequence-to-one implementation likely reflects cyclic manifestations of equifinality (Beven, 2006),



rather than physically meaningful seasonal dynamics. This is further supported by the integrated gradients analysis, where dynamic inputs showed limited contribution in the parameter learning process. These findings contrast with a common assumption that time-varying parameters necessarily reflect changing physical systems dynamics. While such relationships have been reported in sequence-to-sequence LSTM implementations (Feng et al., 2022), they do not appear to be the case for the sequence-to-one implementation presented in this work.

There are several limitations of this work that warrant further discussion. First, this study focuses exclusively on sequence-to-one LSTM formulations, which have received less scrutiny than sequence-to-sequence approaches. Therefore, our conclusions may not generalize to sequence-to-sequence models, although others have already highlighted interpretability challenges with those formulations (Acuña Espinoza et al., 2024, 2025). Second, while integrated gradients provide useful insight into feature contributions, they do not capture interaction effects; future work should therefore consider alternative explainable AI approaches (e.g., Janizek et al., 2020) and ensure insights are robust across methods. Additionally, our generalization analysis is limited to out-of-sample temporal and spatial cases within a single region (CONUS). Evaluating transferability across regions and under climate non-stationarity remains an important next step. Finally, while MLP-based models yield interpretable, basin-specific parameters and match LSTM-based hybrid model performance, they remain black-box function approximators. Replacing MLPs with more interpretable architectures, such as Kolmogorov-Arnold Networks (KAN; Liu et al., 2025) that use learnable and visualizable activation functions, may enable deeper insight into internal model processes (Jing et al., 2025). Such advances could support hybrid modeling frameworks that offer transparency not only in parameter estimation, but also in the representation of hydrological processes, which is critical for inference and decision-making under non-stationary conditions.

In conclusion, this study highlights the value of parsimony in differentiable hydrological modeling. MLP-based differentiable models match the predictive performance of LSTM-based models while offering greater interpretability through static, basin-specific parameters. Comprehensive basin attributes appear largely redundant, serving primarily as spatial proxies or mechanisms for memorization, rather than enabling the learning of physically meaningful relationships. Together, these findings suggest that increasing model complexity does not necessarily yield improved insights or generalization, and that caution is required when advancing differentiable hydrological models to promote improvements in both predictive skill and transparency.

Appendix A: Study area

The study area includes 531 basins (Fig. A1) from the CAMELS-US dataset. We used 80% ($n = 425$) basins for training and out-of-sample temporal testing, and 20% ($n = 106$) for out-of-sample spatial testing. The dataset includes basin physiographic attributes, meteorological forcings from multiple products (we used Daymet), and observed USGS streamflow



data. Physiographic attributes are static, while meteorological and streamflow records are available at daily resolution from 1980 to 2014. Table 1 shows the attributes and meteorological forcings used in this study.

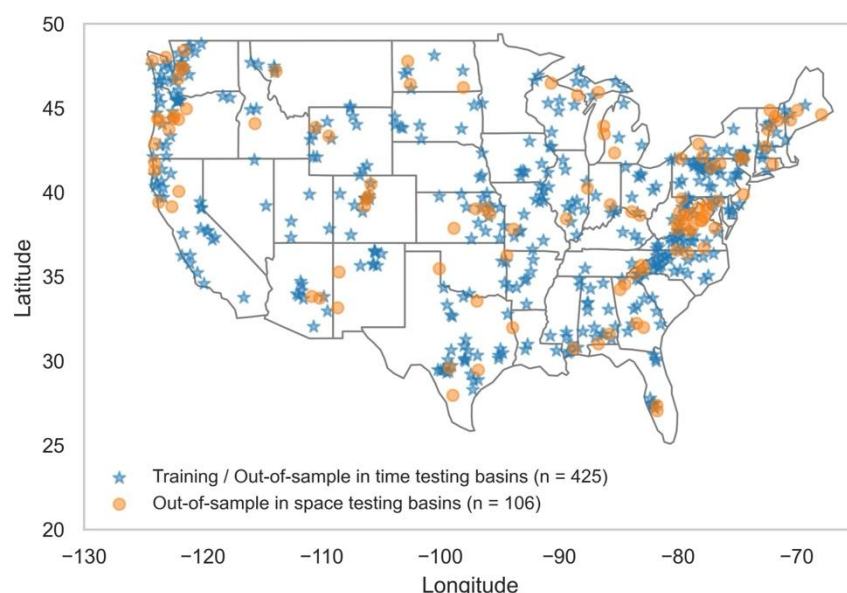


Figure A1: Study area showing the 531 CAMELS-US basins used in this study. Blue stars indicate basins used for training and out-of-sample in time testing (via temporal split). Orange dots indicate basins withheld entirely from training and used exclusively for out-of-sample in space testing.

Appendix B: Process-based HBV model

Our implementation of the process-based HBV model has 20 parameters across three interconnected modules; see Seibert (2005) for a detailed description of process representation. We follow the approach in Seibert (2005) with one key modification, where instead of single-parameter triangular routing, we use separate two-parameter gamma routing for direct flow and base flow. Table B1 shows the process representation, parameters, parameter descriptions, and bounds.

Table B1: HBV model parameters and feasible ranges.

Process	Parameter	Description	Lower bound	Upper bound
Rain and Snow Module	ts	Threshold temperature for snowfall	-4	4
	tm	Threshold temperature for snowmelt	-4	4
	tti	Temperature interval for a mixture of snow and rain	0.1	4
	scf	Snowfall correction factor	0.5	2



	ddf	Degree-day factor for snowmelt	0.05	10
	whc	Water holding capacity of the snowpack	0.05	0.2
	crf	Refreezing coefficient	0.1	1
Soil Moisture and Evapo- transpiration Module	coeff_pet	Coefficient of potential evapotranspiration	0.5	2
	fc	Maximum soil moisture storage; field capacity	1	1000
	pwp	Soil permanent wilting point	0.01	0.99
	beta	Shape coefficient governing the fate of water input to soil moisture storage	0.5	5
Runoff Separation and Routing Module	l	Threshold parameter for upper storage; if water level is higher than the threshold, shallow flow occurs	1	999
	ks	Recession constant for near surface storage	0.01	0.99
	ki	Recession constant for upper storage	0.01	0.99
	kperc	Percolation constant, max flow rate from upper to lower storage	0.0001	0.99
	kb	Recession constant for lower storage	0.001	0.99
	d_shape	Gamma function shape parameter for direct flow	1	4
	d_scale	Gamma function scale parameter for direct flow	0.5	4
	b_shape	Gamma function shape parameter for base flow	1	6
	b_scale	Gamma function scale parameter for base flow	1	7

Appendix C: Hyper-parameter tuning of differentiable models

For all four differentiable architectures, we tuned hyperparameters to determine the neural network size. The optimal architecture was selected using an early stopping criterion, defined as no decrease in validation loss for 10 consecutive training epochs. The hyperparameter tuning results are presented in Table C1, with the top-performing configuration italicized. Overall, the selected hyperparameter sets are justified not only based on validation performance but also in terms of reduced training time.

Table C1: Hyperparameter tuning results for four differentiable architectures. All models use a stride length of 60.

MLP (3 layers)

HBV Units	Hidden Size	Val. Loss	Time (min)	Epochs
1	256	4.13	232.48	19
1	512	4.04	224.49	18
1	1024	3.97	260.33	21



<i>1</i>	<i>2048</i>	<i>3.95</i>	<i>168.01</i>	<i>12</i>
1	4096	3.97	151.51	10
16	256	3.24	576.68	52
16	512	3.25	335.93	28
16	1024	3.2	292.84	25
<i>16</i>	<i>2048</i>	<i>3.18</i>	<i>246.99</i>	<i>20</i>
16	4096	3.28	217.73	17

LSTM (1 layer)

HBV Units	Hidden Size	Val. Loss	Time (min)	Epochs
1	128	4.14	317.34	27
1	256	4.01	252.76	20
1	384	4.02	172.9	12
<i>1</i>	<i>512</i>	<i>3.52</i>	<i>310.15</i>	<i>24</i>
1	1024	3.98	260.53	20
16	128	3.4	535.23	50
16	256	3.33	638.21	56
16	384	3.31	379.23	33
16	512	3.25	323.67	26
<i>16</i>	<i>1024</i>	<i>3.19</i>	<i>377.83</i>	<i>30</i>
16	2048	3.26	542.23	42

Appendix D: Integrated Gradients analysis of LSTM-based differentiable model

This study employs an explainable AI technique known as Integrated Gradients (IG) (Sundararajan et al., 2017), which quantifies the contribution of each input feature to model predictions by integrating gradients along the path from a baseline input to the observed input. We applied IG to the LSTM-based, single-HBV unit differentiable model to assess how both static and dynamic inputs influence parameter estimation. For each daily HBV parameter prediction, the LSTM uses a one-year lookback period comprising concatenated static (30) and dynamic (3) input features ($\mathbb{R}^{365 \times 33}$). We compute IG attributions for each of the 20 HBV parameters, yielding a three-dimensional attribution tensor over lookback period, input features, and HBV parameters ($\mathbb{R}^{365 \times 33 \times 20}$) for each prediction. This analysis is repeated for all timesteps in the testing period for each basin, and finally, we compute the average of the absolute IG values for each input feature. The results for basins with high, medium, and low parameter variability are shown in Fig. D1.

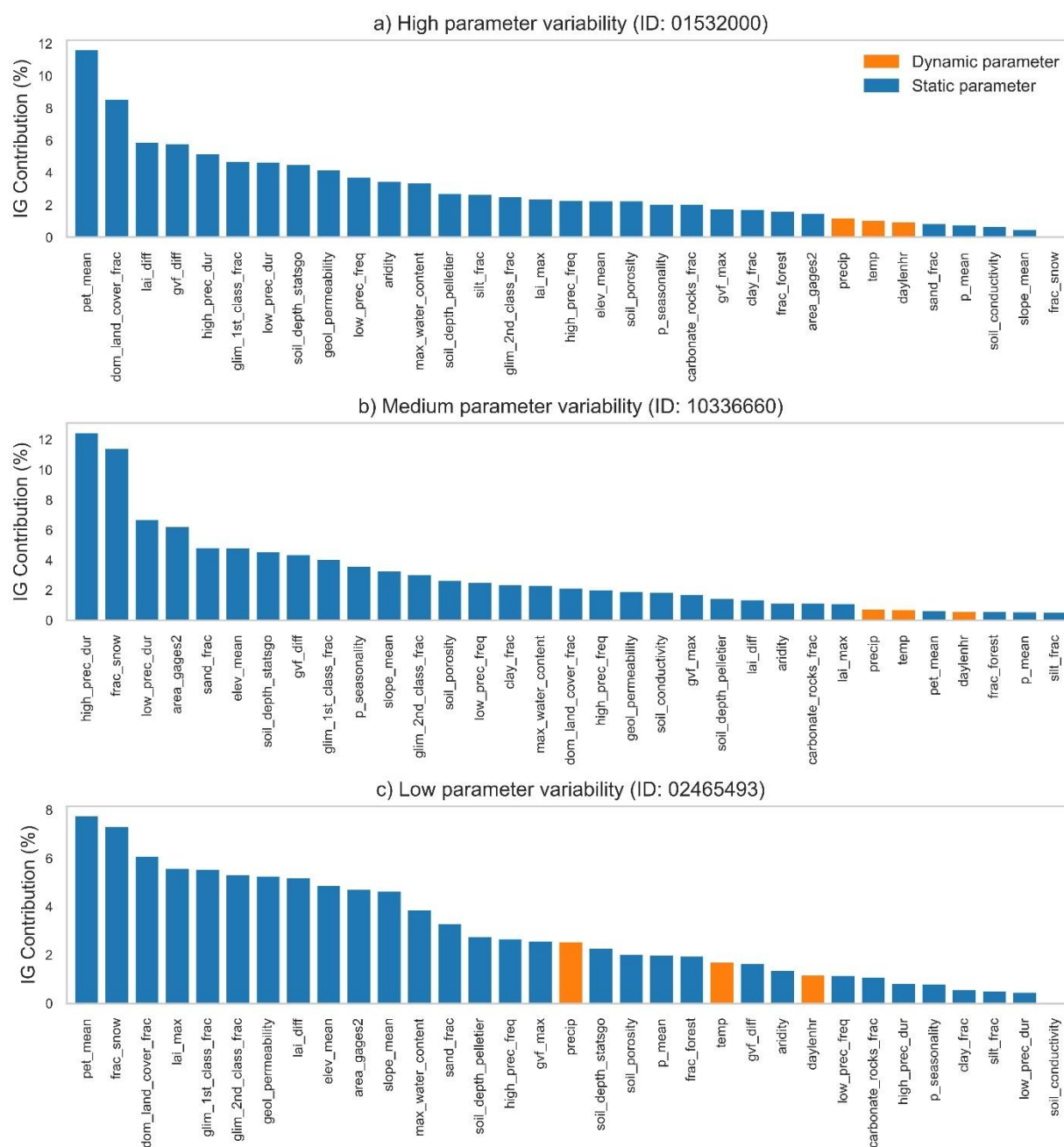


Figure D1: Feature importance for the LSTM+HBV differentiable hybrid model for basin with (a) high parameter variability, (b) medium parameter variability and (c) low parameter variability. Bars show the mean contribution of each input feature to parameter estimation, computed as the mean absolute integrated gradients and normalized to sum to 100%.



Code availability

The codes used to conduct all the analyses in this paper are publicly available at: <https://github.com/snpoudel/diff-hydro>

Data availability

The CAMELS-US dataset is publicly available at <https://doi.org/10.5065/D6MW2F4D> (Newman et al., 2022).

535 Author contribution

S.P. and S.S. conceptualized the project. S.P. developed the model code, conducted experiments, and prepared the initial draft. S.S. supervised the project and reviewed and edited the manuscript.

Competing interests

Authors have no competing interests.

540 Acknowledgements

This research was supported by the U.S. Geological Survey, United States (USGS; grant number G23AC00678-00). The views expressed in this work are those of the authors and do not reflect the views or policies of the USGS.

Financial support

545 This research was supported by the U.S. Geological Survey, United States (USGS; grant number G23AC00678-00). The views expressed in this work are those of the authors and do not reflect the views or policies of the USGS.

References

- Acuña Espinoza, E., Loritz, R., Álvarez Chaves, M., Bäuerle, N., & Ehret, U. (2024). To bucket or not to bucket? Analyzing the performance and interpretability of hybrid hydrological models with dynamic parameterization. *Hydrology and Earth System Sciences*, 28(12), 2705–2719. <https://doi.org/10.5194/hess-28-2705-2024>
- 550 Acuña Espinoza, E., Loritz, R., Kratzert, F., Klotz, D., Gauch, M., Álvarez Chaves, M., & Ehret, U. (2025). Analyzing the generalization capabilities of a hybrid hydrological model for extrapolation to extreme events. *Hydrology and Earth System Sciences*, 29(5), 1277–1294. <https://doi.org/10.5194/hess-29-1277-2025>



- Addor, N., Newman, A. J., Mizukami, N., & Clark, M. P. (2017). The CAMELS data set: Catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences*, 21(10), 5293–5313.
 555 <https://doi.org/10.5194/hess-21-5293-2017>
- Baste, S., Klotz, D., Acuña Espinoza, E., Bardossy, A., & Loritz, R. (2025). Unveiling the limits of deep learning models in hydrological extrapolation tasks. *Hydrology and Earth System Sciences*, 29(21), 5871–5891. <https://doi.org/10.5194/hess-29-5871-2025>
- Bennett, A., & Nijssen, B. (2021). Deep Learned Process Parameterizations Provide Better Representations of Turbulent
 560 Heat Fluxes in Hydrologic Models. *Water Resources Research*, 57(5), e2020WR029328.
<https://doi.org/10.1029/2020WR029328>
- Bergström, S. (1995). The HBV model. <https://www.cabidigitallibrary.org/doi/full/10.5555/19961904773>
- Beven, K., & Binley, A. (1992). The future of distributed models: Model calibration and uncertainty prediction. *Hydrological Processes*, 6(3), 279–298. <https://doi.org/10.1002/hyp.3360060305>
- 565 Blasone, R.-S., Vrugt, J. A., Madsen, H., Rosbjerg, D., Robinson, B. A., & Zyvoloski, G. A. (2008). Generalized likelihood uncertainty estimation (GLUE) using adaptive Markov Chain Monte Carlo sampling. *Advances in Water Resources*, 31(4), 630–648. <https://doi.org/10.1016/j.advwatres.2007.12.003>
- Bohl, J. P., Wood, R. R., Frank, C., Astagneau, P. C., Peters, J., & Brunner, M. I. (2025). Hybrid models generalize better to warmer climate conditions than process-based and purely data-driven models. *Catchment hydrology/Modelling approaches*.
 570 <https://doi.org/10.5194/egusphere-2025-5201>
- Chiang, Y.-M., Hao, R.-N., Ho, H.-C., Chang, T.-J., & Xu, Y.-P. (2017). Evaluating the contribution of multi-model combination to streamflow hindcasting by empirical and conceptual models. *Hydrological Sciences Journal*, 62(9), 1456–1468. <https://doi.org/10.1080/02626667.2017.1330543>
- Feng, D., Beck, H., de Bruijn, J., Sahu, R. K., Satoh, Y., Wada, Y., Liu, J., Pan, M., Lawson, K., & Shen, C. (2024). Deep
 575 dive into hydrologic simulations at global scale: Harnessing the power of deep learning and physics-informed differentiable models (δ HBV-globe1.0-hydroDL). *Geoscientific Model Development*, 17(18), 7181–7198. <https://doi.org/10.5194/gmd-17-7181-2024>
- Feng, D., Beck, H., Lawson, K., & Shen, C. (2023). The suitability of differentiable, physics-informed machine learning hydrologic models for ungauged regions and climate change impact assessment. *Hydrology and Earth System Sciences*,
 580 27(12), 2357–2373. <https://doi.org/10.5194/hess-27-2357-2023>
- Feng, D., Fang, K., & Shen, C. (2020). Enhancing Streamflow Forecast and Extracting Insights Using Long-Short Term Memory Networks With Data Integration at Continental Scales. *Water Resources Research*, 56(9), e2019WR026793. <https://doi.org/10.1029/2019WR026793>
- Feng, D., Liu, J., Lawson, K., & Shen, C. (2022). Differentiable, Learnable, Regionalized Process-Based Models With
 585 Multiphysical Outputs can Approach State-Of-The-Art Hydrologic Prediction Accuracy. *Water Resources Research*, 58(10), e2022WR032404. <https://doi.org/10.1029/2022WR032404>



- Gupta, H. V. (2026). On generalization, language, interpretability and the future of geo-scientific machine learning. *Environmental Modelling & Software*, 197, 106834. <https://doi.org/10.1016/j.envsoft.2025.106834>
- Hamon, W. R. (1961). Estimating Potential Evapotranspiration. *Journal of the Hydraulics Division*, 87(3), 107–120. <https://doi.org/10.1061/JYCEAJ.0000599>
- Heudorfer, B., Gupta, H. V., & Loritz, R. (2025). Are Deep Learning Models in Hydrology Entity Aware? *Geophysical Research Letters*, 52(6), e2024GL113036. <https://doi.org/10.1029/2024GL113036>
- Heudorfer, B., Liesch, T., & Broda, S. (2024). On the challenges of global entity-aware deep learning models for groundwater level prediction. *Hydrology and Earth System Sciences*, 28(3), 525–543. <https://doi.org/10.5194/hess-28-525-2024>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. *Neural Computation*. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Janizek, J. D., Sturmfels, P., & Lee, S.-I. (2020). Explaining Explanations: Axiomatic Feature Interactions for Deep Networks (arXiv:2002.04138). *arXiv*. <https://doi.org/10.48550/arXiv.2002.04138>
- Ji, H., Song, Y., Bindas, T., Shen, C., Yang, Y., Pan, M., Liu, J., Rahmani, F., Abbas, A., Beck, H., Lawson, K., & Wada, Y. (2025). Distinct hydrologic response patterns and trends worldwide revealed by physics-embedded learning. *Nature Communications*, 16(1), 9169. <https://doi.org/10.1038/s41467-025-64367-1>
- Jing, X., Yang, X., Luo, J., & Zuo, G. (2025). Exploring Kolmogorov-Arnold neural networks for hybrid and transparent hydrological modeling. *Environmental Modelling & Software*, 193, 106648. <https://doi.org/10.1016/j.envsoft.2025.106648>
- Karpatne, A., Atluri, G., Faghmous, J. H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., & Kumar, V. (2017). Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10), 2318–2331. <https://doi.org/10.1109/TKDE.2017.2720168>
- Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization (arXiv:1412.6980). *arXiv*. <https://doi.org/10.48550/arXiv.1412.6980>
- Kirchner, J. W. (2006). Getting the right answers for the right reasons: Linking measurements, analyses, and models to advance the science of hydrology. *Water Resources Research*, 42(3). <https://doi.org/10.1029/2005WR004362>
- Kraft, B., Jung, M., Körner, M., Koirala, S., & Reichstein, M. (2022). Towards hybrid modeling of the global hydrological cycle. *Hydrology and Earth System Sciences*, 26(6), 1579–1614. <https://doi.org/10.5194/hess-26-1579-2022>
- Kratzert, F., Gauch, M., Klotz, D., & Nearing, G. (2024). HESS Opinions: Never train an LSTM on a single basin. *Hydrology and Earth System Sciences Discussions*, 1–19. <https://doi.org/10.5194/hess-2023-275>
- Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks. *Hydrology and Earth System Sciences*, 22(11), 6005–6022. <https://doi.org/10.5194/hess-22-6005-2018>
- Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T. Y., & Tegmark, M. (2025). KAN: Kolmogorov-Arnold Networks (arXiv:2404.19756). *arXiv*. <https://doi.org/10.48550/arXiv.2404.19756>



- Mangunkiya, N. K., & Sharma, A. (2025). Integrating Reservoir Dynamics Into Differentiable Process-Based Hydrological Model for Enhanced Streamflow Estimation. *Water Resources Research*, 61(7), e2025WR040268. <https://doi.org/10.1029/2025WR040268>
- 625 Natel de Moura, C., Seibert, J., & Detzel, D. H. M. (2022). Evaluating the long short-term memory (LSTM) network for discharge prediction under changing climate conditions. *Hydrology Research*, 53(5), 657–667. <https://doi.org/10.2166/nh.2022.044>
- Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C., & Gupta, H. V. (2021). What Role Does Hydrological Science Play in the Age of Machine Learning? *Water Resources Research*, 57(3), e2020WR028091. <https://doi.org/10.1029/2020WR028091>
- 630 Newman, A. J., Mizukami, N., Clark, M. P., Wood, A. W., Nijssen, B., & Nearing, G. (2017). Benchmarking of a Physically Based Hydrologic Model. <https://doi.org/10.1175/JHM-D-16-0284.1>
- Poudel, S., Najibi, N., & Steinschneider, S. (2026). Uncertainty in estimating the relative change of design floods under climate change: A stylized experiment with process-based, deep learning, and hybrid models. *Journal of Hydrology*, 664, 134427. <https://doi.org/10.1016/j.jhydrol.2025.134427>
- 635 Razavi, S. (2021). Deep learning, explained: Fundamentals, explainability, and bridgeability to process-based modelling. *Environmental Modelling & Software*, 144, 105159. <https://doi.org/10.1016/j.envsoft.2021.105159>
- Reichert, P., Ma, K., Höge, M., Fenicia, F., Baity-Jesi, M., Feng, D., & Shen, C. (2024). Metamorphic testing of machine learning and conceptual hydrologic models. *Hydrology and Earth System Sciences*, 28(11), 2505–2529. <https://doi.org/10.5194/hess-28-2505-2024>
- 640 Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., & Prabhat. (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, 566(7743), 195–204. <https://doi.org/10.1038/s41586-019-0912-1>
- Seibert, J. (2005). HBV light version 2, user's manual. Department of Earth Sciences, Uppsala University, Uppsala.
- Shen, C., Appling, A. P., Gentine, P., Bandai, T., Gupta, H., Tartakovsky, A., Baity-Jesi, M., Fenicia, F., Kifer, D., Li, L.,
- 645 Liu, X., Ren, W., Zheng, Y., Harman, C. J., Clark, M., Farthing, M., Feng, D., Kumar, P., Aboelyazeed, D., ... Lawson, K. (2023). Differentiable modelling to unify machine learning and physical models for geosciences. *Nature Reviews Earth & Environment*, 4(8), 552–567. <https://doi.org/10.1038/s43017-023-00450-9>
- Shin, S., Her, Y., Muñoz-Carpena, R., & Khare, Y. P. (2023). Multi-parameter approaches for improved ensemble prediction accuracy in hydrology and water quality modeling. *Journal of Hydrology*, 622, 129458. <https://doi.org/10.1016/j.jhydrol.2023.129458>
- 650 Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic Attribution for Deep Networks (arXiv:1703.01365). arXiv. <https://doi.org/10.48550/arXiv.1703.01365>
- Todini, E. (2007). Hydrological catchment modelling: Past, present and future. *Hydrology and Earth System Sciences*, 11(1), 468–482. <https://doi.org/10.5194/hess-11-468-2007>



- 655 Tsai, W.-P., Feng, D., Pan, M., Beck, H., Lawson, K., Yang, Y., Liu, J., & Shen, C. (2021). From calibration to parameter learning: Harnessing the scaling effects of big data in geoscientific modeling. *Nature Communications*, 12(1), 5988. <https://doi.org/10.1038/s41467-021-26107-z>
- Wagener, T., Boyle, D. P., Lees, M. J., Wheater, H. S., Gupta, H. V., & Sorooshian, S. (2001). A framework for development and application of hydrological models. *Hydrology and Earth System Sciences*, 5(1), 13–26.
- 660 <https://doi.org/10.5194/hess-5-13-2001>
- Wi, S., & Steinschneider, S. (2022). Assessing the Physical Realism of Deep Learning Hydrologic Model Projections Under Climate Change. *Water Resources Research*, 58(9), e2022WR032123. <https://doi.org/10.1029/2022WR032123>
- Wi, S., & Steinschneider, S. (2024). On the need for physical constraints in deep learning rainfall–runoff projections under climate change: A sensitivity analysis to warming and shifts in potential evapotranspiration. *Hydrology and Earth System*
- 665 *Sciences*, 28(3), 479–503. <https://doi.org/10.5194/hess-28-479-2024>
- Zhong, L., Lei, H., & Gao, B. (2023). Developing a Physics-Informed Deep Learning Model to Simulate Runoff Response to Climate Change in Alpine Catchments. *Water Resources Research*, 59(6), e2022WR034118. <https://doi.org/10.1029/2022WR034118>
- Zhong, L., Lei, H., & Yang, J. (2024). Development of a Distributed Physics-Informed Deep Learning Hydrological Model
- 670 for Data-Scarce Regions. *Water Resources Research*, 60(6), e2023WR036333. <https://doi.org/10.1029/2023WR036333>