Water Resources Research

RESEARCH ARTICLE
10.1029/2025WR040173

Probabilistic Physics-Guided Deep Neural Networks With Recurrence and Attention Mechanisms for Interpretable Daily Streamflow Simulation

Sadegh Sadeghi Tabas[1,2], Vidya Samadi[3,4], Catherine Wilson[5], and Biswa Bhattacharya[6]

[1]The Glenn Department of Civil Engineering, Clemson University, Clemson, SC, USA, [2]School of Computing, Clemson University, Clemson, SC, USA, [3]Department of Agricultural Sciences, Clemson University, Clemson, SC, USA, [4]School of Computing, Artificial Intelligence Research Institute for Science and Engineering (AIRISE), Clemson University, Clemson, SC, USA, [5]Hydro-Environmental Research Center, School of Engineering, Cardiff University, Cardiff, UK, [6]Department of Hydroinformatics and Socio-Technical Innovation, IHE Delft Institute for Water Education, Delft, The Netherlands

Key Points:

- Exploring the interpretability of deep neural networks for accurate daily streamflow simulation across the continental United States
- The physics-guided TFT exhibits superior simulation, emphasizing the role of catchment attributes in data-driven streamflow simulations
- Uncertainty quantifications revealed the capability of TFT in transparently handling both temporal and long-term streamflow dynamics

Correspondence to:
V. Samadi,
samadi@clemson.edu

Citation:
Sadeghi Tabas, S., Samadi, V., Wilson, C., & Bhattacharya, B. (2025). Probabilistic physics-guided deep neural networks with recurrence and attention mechanisms for interpretable daily streamflow simulation. *Water Resources Research*, *61*, e2025WR040173. https://doi.org/10.1029/2025WR040173

Received 7 FEB 2025
Accepted 31 JUL 2025

Author Contributions:
Conceptualization: Vidya Samadi, Catherine Wilson
Data curation: Sadegh Sadeghi Tabas, Vidya Samadi
Formal analysis: Sadegh Sadeghi Tabas, Vidya Samadi
Funding acquisition: Vidya Samadi
Investigation: Vidya Samadi
Methodology: Vidya Samadi, Catherine Wilson, Biswa Bhattacharya
Project administration: Vidya Samadi
Resources: Vidya Samadi
Software: Sadegh Sadeghi Tabas, Vidya Samadi
Supervision: Vidya Samadi, Catherine Wilson, Biswa Bhattacharya
Validation: Sadegh Sadeghi Tabas, Vidya Samadi, Biswa Bhattacharya
Visualization: Sadegh Sadeghi Tabas
Writing – original draft: Sadegh Sadeghi Tabas

Abstract As Deep Neural Networks (DNNs) are being increasingly employed to make important simulations in rainfall-runoff contexts, the demand for interpretability is increasing in the hydrology community. Interpretability is not just a scientific question, but rather knowing where the models fall flat, how to fix them, and how to explain their outcomes to scientific communities so that everyone understands how the model arrives at specific simulations This paper addresses these challenges by deciphering interpretable probabilistic DNNs utilizing the Deep Autoregressive Recurrent (DeepAR) and Temporal Fusion Transformer (TFT) for daily streamflow simulation across the continental United States (CONUS). We benchmarked TFT and DeepAR against conceptual to physics-based hydrologic models. In this setting, catchment physical attributes were incorporated into the training process to create physics-guided TFT and DeepAR configurations. Our proposed physics-guided configurations are also designed to aggregate the patterns across the entire data set, analyze the sensitivity of key catchment physical attributes and facilitate the interpretability of temporal dynamics in rainfall-runoff generation mechanisms. To assess the uncertainty, the modeling configurations were coupled with a quantile regression by adding Gaussian noise $N(0, \sigma)$ with increasing standard deviation to the individual catchment attributes. Analysis suggested that the physics-guided TFT was superior in predicting daily streamflow compared to the original TFT and DeepAR as well as benchmark hydrologic models. Predictive uncertainty intervals effectively bracketed most of the observational data by simultaneous simulation of various percentiles (e.g., 10th, 50th, and 90th). Interpretable physics-guided TFT proved to be a strong candidate for CONUS daily streamflow simulations.

Plain Language Summary Explanations supporting the output of deep neural networks (DNNs) are crucial in rainfall-runoff modeling, where experts require far more information from the model than a simple classical simulation to support modeling diagnosis. This research delves into exploring interpretable probabilistic DNNs by developing Deep Autoregressive Recurrent (DeepAR) and Temporal Fusion Transformer (TFT) models. These models were rigorously evaluated against traditional hydrologic methods, both conceptual and physics-based, emphasizing the integration of catchment physical attributes into the models for daily streamflow simulations across the continental United States (CONUS). Leveraging quantile regression to evaluate predictive uncertainty, the physics-guided TFT model notably outperformed other models by demonstrating superior predictive capabilities, particularly in managing high and low flow fluctuations. Notably, this study showed that physics-guided TFT model can effectively leverage its interpretable multi-head attention mechanism to weigh the importance of temporal flow dynamics based on the relationships between forcing data, catchment physical attributes, and streamflow records. The findings of this study show promising results of transformer rainfall-runoff simulations, thereby highlighting its robustness in effectively utilizing physical attributes and improving model interpretability.

1. Introduction

Deep neural networks (DNNs) have been widely used for enhancing sequential data modeling and building structured data models (Y. Chen et al., 2018; Fischer & Krauss, 2018; Kratzert et al., 2018; Kratzert, Herrnegger, et al., 2019; Tabas & Samadi, 2022). These models are particularly suitable for rainfall-runoff simulation in the

Writing – review & editing: Vidya Samadi, Catherine Wilson, Biswa Bhattacharya

context of giving precise and timely processing of arbitrary sequences of input-output data (Shen, 2018). Many state-of-the-art DNN approaches have recently been established based on Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM), to simulate rainfall-runoff processes in different settings. There are cell states in the LSTM networks that represent long-term memory, storing information across multiple time steps that can be interpreted as a kind of "storage" to carry relevant information through the local sequence-to-sequence processing. In LSTM, the updating of internal cell states is regulated through a number of gates: the first gate regulates the flow of information in and out of the memory cell, and the second one controls which new information from the current input should be added to the cell state, and the third gate decides what part of the cell's memory should be outputted at a given time step.

The elaborate gated design of LSTM partly addresses the long-term dependency problem in time series modeling (Fang et al., 2020). However, LSTM's structure is inherently sequential, with data dependencies flowing left to right (i.e., past to future), and it does not explicitly enforce causal masking. In applications such as rainfall-runoff modeling, where physical processes operate through strict cause-effect relationships, preserving causality is critical. Without explicit constraints, LSTMs may, in some configurations (e.g., bidirectional LSTMs or unconstrained training setups), inadvertently violate physical causality. In contrast, transformer models, through causal attention mechanisms, can explicitly model unidirectional relationships in time, allowing for a data-driven but causally consistent understanding of the interactions among hydrological variables (e.g., Berrevoets et al., 2023). This makes transformers particularly suitable for rainfall-runoff modeling tasks that require respect for the inherent directionality of physical processes.

The unidirectional structures can connect two arbitrary positions in a time series process directly by using a self-attention module. This can strengthen the connection of two arbitrary positions in time series data, where each data point in the sequence can "attend" to any other point in the sequence, not just the ones before or after it. In addition, transformers have a longer memory than the LSTM, thus superior in quality while being more parallelizable and requiring significantly less time for training (Vaswani et al., 2017). Transformer algorithms use self-attention and cross-attention mechanisms to create an explicit interpretable model, which follows the trend of Explainable Artificial Intelligence (XAI; see Wen et al., 2022). Using self-attention and cross-attention mechanisms, the input data (e.g., meteorological forcing, catchment attributes, and runoff observations) of a position is related to those of all positions. Thus, their correlations or data similarities can be obtained. This is beneficial for rainfall-runoff simulations because the model can give more attention (or larger weights) to the positions that have more correlations to the position needed for the runoff generation (i.e., cause-effect relationships or rainfall-runoff relationships).

Unlike traditional hydrologic models, transformer-based models do not inherently "know" the laws governing hydrologic processes, such as mass conservation or the physical mechanisms controlling storage, infiltration, evapotranspiration, and runoff generation within a drainage system. To gain confidence in transformer implementations beyond their use as "black box" models, it is essential to guide or constrain them using physically meaningful information. Physical consistency can be encouraged through techniques such as adding regularization terms to the loss function, penalizing violations of conservation principles like mass, energy, or momentum (e.g., Jia et al., 2019; Karpathy et al., 2015; Shen, 2018). Further advances include embedding physical laws directly into the network design, as seen in physics-informed architectures that explicitly encode conservation constraints (e.g., Hoedt et al., 2021; Karniadakis et al., 2021) or enforce physically realistic outputs through specialized training objectives (e.g., Daw et al., 2020). Another alternative to physics guidance is the incorporation of catchment physical attributes—such as soil properties, land cover, topography, and climate indices—directly into the model inputs, enabling the transformer to condition its predictions on physical basin characteristics during training rather than treating the transformer purely as a black-box, allowing the model to better capture physical variability across diverse basins. This strategy balances flexibility with physical realism, improving both generalization and interpretability without fully sacrificing the transformer's data-driven strengths.

Transformer approaches (and indeed most DNN algorithms) view data-driven processes as deterministic functions, and as a result, direct optimization (without complexity control) of these algorithms may lead to unreliable results due to uncertainty (Sadeghi Tabas, 2023). One reason for this is that the parameter (weight) estimation involves the inversion of a nonlinear system (here, catchment system) from noisy data, which is typically ill-posed (e.g., Casdagli, 1989; Haykin & Principe, 1998; Tabas and Samadi, 2022). In this situation, noises might exist within

observation that are referred to as data uncertainties (also called aleatoric uncertainty, see Der Kiureghian & Ditlevsen, 2009). In addition, there are many situations where uncertainties arise from the DNN structure choice and parameters. This is referred to as model uncertainty or epistemic uncertainty. The standard approach to tackling ill-posed problems (both aleatoric and epistemic uncertainties) is by means of applying probabilistic approaches such as the Gaussian process to modeling procedure (e.g., Moradkhani et al., 2005; Raftery et al., 2005; Samadi et al., 2020).

The transformers, along with their improved versions (see Dai et al., 2019), have successfully been applied in several simulation tasks recently (e.g., Q. Chen et al., 2019; Dai et al., 2019; Dosovitskiy et al., 2020; Gonzalez et al., 2021; Rasmy et al., 2021; Vaswani et al., 2017; Zhou et al., 2021). To our knowledge, many state-of-the-art DNN approaches for rainfall-runoff modeling are established based on LSTMs, and there are very few studies that implemented simple transformer models (e.g., Pölz et al., 2024; Yin et al., 2022) as well as other probabilistic DNNs such as deep auto-regressive approaches, Bayesian deep learning and variational Bayesian inference approaches (e.g., D. Li et al., 2021; Piazzi et al., 2021; Tabas & Samadi, 2022). Furthermore, incorporating catchment physical attributes (or exogenous features/parameters) and understanding the uncertainty associated with DNN modeling are rarely explored (Feng et al., 2020; Kratzert et al., 2021; Tabas & Samadi, 2022). The vision of this study is thus to address these knowledge gaps by investigating the potential of probabilistic and transformer algorithms for rainfall-runoff modeling and uncertainty assessment. The novelty of this research lies on several fronts notably (a) developing two advanced DNN approaches, that is, Deep Autoregressive Recurrent Networks (DeepAR) and Temporal Fusion Transformer (TFT), for rainfall-runoff modeling that can learn catchment similarities directly from meteorological forcing data and ancillary data of multiple catchments across the continental United States (CONUS), (b) demonstrating how climatic and catchment physical attributes control spatiotemporal variability of rainfall-runoff processes, and (c) quantifying uncertainty in rainfall-runoff modeling using quantile regression approach as the likelihood function (loss function) by adding Gaussian noise $N(0, \sigma)$ with increasing standard deviation to the individual attribute value.

Both TFT and DeepAR were trained using static (time-invariant/independent) and dynamic (time-variant/dependent) attributes to predict daily streamflow values across CONUS. In this setting, the algorithms learned how to combine different parts of the network to simulate various types of rainfall-runoff behaviors over time. In principle, the approach explicitly allows for sharing some parts of the networks for similarly behaving catchments while using different independent parts for those catchments with completely different rainfall-runoff behaviors. Furthermore, our proposed methodology provides a mapping function from catchment attribute space into a learned, high-dimensional space where catchments with similar rainfall–runoff behavior can be clustered together. The results are then used to perform a catchment similarity analysis. Through sensitivity analysis and hierarchical (clustering) temporal modeling, both algorithms offered a transparent view of short and long-term patterns within the daily streamflow time series data, facilitating a deeper understanding of the factors influencing rainfall-runoff generation mechanisms. The main novelty of this research lies in the use of a probabilistic TFT for daily streamflow prediction at CONUS. While similar efforts such as Koya and Roy (2024) have recently emerged, to the best of our knowledge, our work is among the first to demonstrate this approach using a probabilistic framework at CONUS. The use of the probabilistic TFT allowed point predictions as well as full predictive distributions, which will be helpful in operational hydrologic forecasting. Moreover, the combined use of both static and dynamic inputs reflects a physics-aware design, which is likely to enhance the model's generalization capacity (e.g., Kratzert, Herrnegger, et al., 2019). Finally, it is worth noting that interpretability in this work is achieved through TFT's built-in attention and variable selection mechanisms, which allow the model to identify the most relevant inputs across both temporal and feature dimensions. Our contribution lies in leveraging these interpretability tools at scale—systematically analyzing variable importance across CONUS. This large-scale application provides novel insights into the spatial variability of hydrological simulations and deepens understanding of the underlying physical processes.

The remainder of this paper is organized as follows. Section 2 discusses the study area and the data, followed by the mathematical formulation of DeepAR and TFT and the workflow structures of proposed modeling approaches. Section 3 presents the results of the modeling implementations. This is followed by Section 4, which provides discussion and future research.

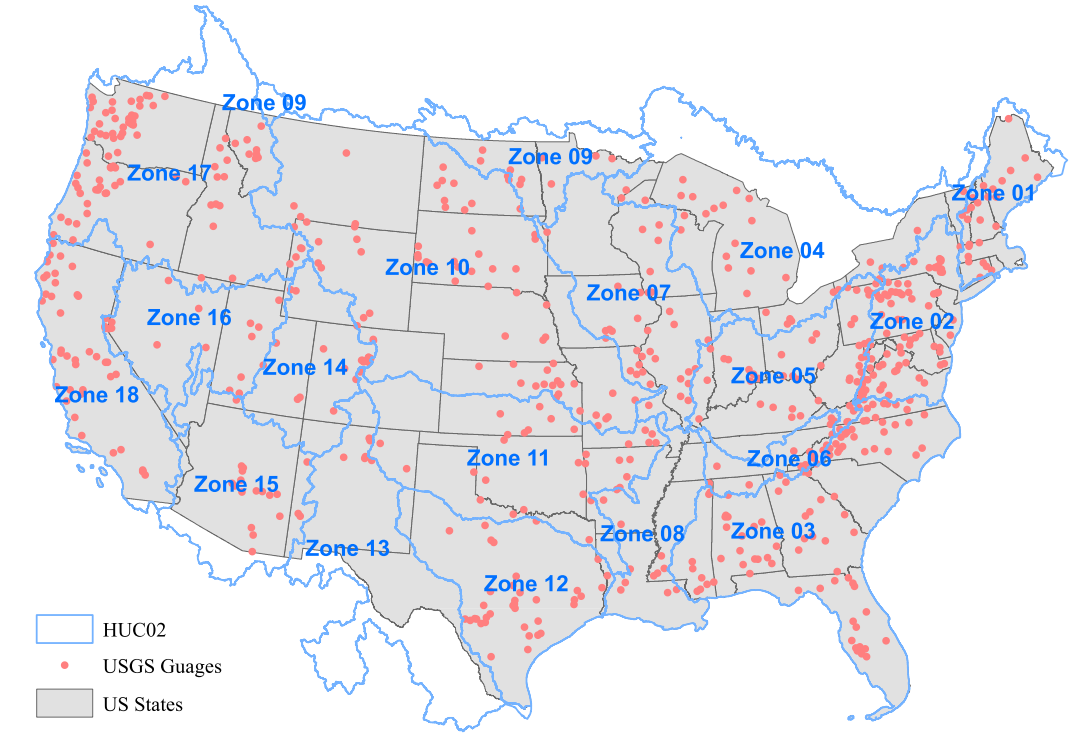

Figure 1. Overview of the 18 CAMELS HUC2 basins (or zone) across CONUS.

2. Methodology

2.1. Study Area and Data

Experimental data were gathered from the publicly available Catchment Attributes and Meteorology for Large-sample Studies (CAMELS) curated by the National Center for Atmospheric Research (NCAR; Newman et al., 2015; Addor et al., 2017). CAMELS consists of 18 HUC2 (Hydrologic Unit Code 2) or zones containing overall 671 catchments (HUC8) ranging in size from 4 to 25,000 km^2 (Figure 1). These catchments were selected based on having minimal human intervention and long-term records of data (1980–2010) gathered from the United States Geological Survey (USGS) gauge II data and the National Water Information System (NWIS). In addition, CAMELS data sets include daily time series of hydrometeorological data such as Daymet (Thornton et al., 2021), Maurer (Maurer et al., 2002), and the North American Land Data Assimilation System (NLDAS; Xia et al., 2012) data sets. Daymet data set provides long-term, continuous, gridded estimates of daily weather and climatology variables, while the Maurer data set is a model-derived data set of land surface states and fluxes, and NLDAS is a quality-controlled and spatially and temporally consistent, land-surface model (LSM) data sets. CAMELS also includes several catchment physical attributes related to soil, climate, vegetation, topography, and geology (Addor et al., 2018). These catchment attributes were derived from maps, remote sensing products, and climate data that are generally available over CONUS. For this project, we used 531 of the 671 CAMELS catchments (those with an area of <2,000 km^2); these 531 catchments are the same ones that were used for model benchmarking by Newman et al. (2017). The CAMELS basins are shown with the HUC2 zones across CONUS (see Figure 1).

2.2. Catchment Static and Dynamic Attributes

The simulation of streamflow can be determined by the hydrological descriptors and catchment attributes that are independent of one another (Addor et al., 2017). Herein, we divided meteorological forcing data as well as catchment physical attributes into dynamic and static attributes, respectively. The continental-scale classification of dominant rainfall-runoff generating processes can define the timing and variability between catchment static and dynamic attributes and how their collective impact dominates the rainfall-runoff generation mechanism across CONUS. The dynamic attributes are defined as the catchments variables that are time-varying, such as (a) daily precipitation, (b) minimum daily air temperature, (c) maximum daily air temperature, (d) average shortwave radiation, and (e) vapor pressure (VP). On the other hand, catchment static attributes are those catchment attributes that remain fixed over time (time-independent variables), such as soil type, geological and topological conditions, and subsurface permeability. To construct physics-guided TFT and DeepAR, catchment static and dynamic attributes were incorporated into the TFT and DeepAR; these variables were chosen as a subset of characteristics explored by Addor et al. (2017) that are derivable from remote sensing and other data products (see Table 1). Prior to model training, all features were standardized by subtracting the mean and dividing by the standard deviation calculated over the training data set. This normalization was performed to stabilize training and ensure comparability between features.

2.3. Probabilistic Modeling Architectures

We employed two probabilistic DNN methods, including an advanced RNN method (so-called Amazon's DeepAR) as well as a Google transformer model, TFT. The DeepAR and TFT models are explained briefly in the following subsections.

2.3.1. DeepAR Architecture

DeepAR, proposed by Salinas et al. (2020), is an encoder-decoder LSTM architecture for the probabilistic simulation of multivariate time series. This approach creates a global model of a multivariate data set, containing related time series instead of creating individual models for each time series. In this setting, the model can extract interrelationships between the variables to provide special treatment for the case where the magnitudes of the time series vary widely (Salinas et al., 2020). According to Salinas et al. (2020), the key advantages of DeepAR over classical DNN approaches are that they (a) provide covariates to capture complex, group-dependent behavior by training on multiple time series simultaneously with minimal manual intervention because the model can learn seasonal behaviors and dependencies on given covariates across time series, (b) make probabilistic simulations in the form of Monte Carlo samples (Ghahramani, 2015; Salinas et al., 2020) that can be used to compute consistent quantile estimates for all sub-ranges in the simulation horizon, (c) provide simulations for the data that have little or no history available, a case where traditional hydrologic models may fail to provide accurate simulation, and (d) incorporate a wide range of likelihood functions, allowing the user to choose the one that is more appropriate for the statistical properties of the data. The goal of DeepAR is to model the conditional distribution which is presented as follows:

$$P(Z_{i,t_0:T}|Z_{i,1:t_0-1}, X_{i,1:T}) \tag{1}$$

where $Z_{i,t}$ is the value of time series i at time t. Given the past series $[Z_{i,1}, Z_{i,2}, \ldots, Z_{i,t_0-1}]$, this model can be employed to predict the future series $[Z_{i,t_0}, Z_{i,t_0+1}, \ldots, Z_{i,T}]$, where t_0 is the time point from which $Z_{i,t}$ needs to be predicted. $[1 : t_0 - 1]$ and $[t_0 : T]$ represent the conditioning range and simulation range, respectively. The DeepAR model predicts the value of the simulation range based on the value of the conditioning range. If covariate time series X_i is introduced in the model, the value of the X_i from time 1 to time $T(X_{i,1:T})$ can also be used for simulation. However, the value of the covariate time series must be available during the entire time period. DeepAR assumes that $P(Z_{i,t_0:T}|Z_{i,1:t_0-1}, X_{i,1:T})$ consists of likelihood factors. These likelihood factors are defined in Equations 2 and 3.

$$P(Z_{i,t_0:T}|Z_{i,1:t_0-1}, X_{i,1:T}) = \prod_{t=t0}^{T} P(Z_{i,t_0:T}|Z_{i,1:t_0-1}, X_{i,1:T}) = \prod_{t=t0}^{T} P(Z_{i,t}|\varnothing(h_{i,t}, \varnothing)) \tag{2}$$

Water Resources Research

10.1029/2025WR040173

Table 1
Daily Basin-Averaged Maurer Forcing Data (Addor et al., 2017) Are Used as Dynamic and Static Attributes, Including Long-Term Average Climatic, and Vegetation Indices, As Well As Soil and Topographical Properties

Type	No.	Category	Attributes and unit	Description
Dynamic Attributes	1	Meteorological attributes	prcp (mm/day)	Precipitation
	2	Meteorological attributes	srad (W/m^2)	Solar Radiation
	3	Meteorological attributes	t_{max} (C)	Maximum Temperature
	4	Meteorological attributes	tmin (C)	Minimum Temperature
	5	Meteorological attributes	vp (Pa)	Vapor Pressure
Static Attributes	1	Topographic	area_gages2 (km^2)	catchment area
	2	Topographic	slope_mean (m/km)	catchment mean slope
	3	Topographic	elev_mean (m above sea level)	catchment mean elevation
	4	Climate	p_mean (mm/day)	mean daily precipitation
	5	Climate	pet_mean (mm/day)	mean daily PET (potential evapotranspiration), estimated by N15 using Priestley–Taylor formulation calibrated for each catchment
	6	Climate	Aridity (−)	aridity (PET/P, a ratio of mean PET, estimated by N15 using Priestley–Taylor formulation calibrated for each catchment to mean precipitation)
	7	Climate	p_seasonality (−)	seasonality and timing of precipitation (estimated using sine curves to represent the annual temperature and precipitation cycles; positive (negative) values indicate that precipitation peaks in summer (winter); values close to 0 indicate uniform precipitation throughout the year)
	8	Climate	frac_snow (−)	fraction of precipitation falling as snow (i.e., on days colder than 0°C)
	9	Climate	high_prec_freq (days/year)	frequency of high precipitation days (≥5 times mean daily precipitation)
	10	Climate	high_prec_dur (days)	average duration of high precipitation events (number of consecutive days ≥5 times mean daily precipitation)
	11	Climate	low_prec_freq (days/year)	frequency of dry days (<1 mm day−1)
	12	Climate	low_prec_dur (days)	average duration of dry periods (number of consecutive days <1 mm day−1)
	13	Land cover	forest_frac (−)	forest fraction
	14	Land cover	lai_max (−)	maximum monthly mean of the leaf area index (based on 12 monthly means)
	15	Land cover	lai_diff (−)	difference between the maximum and minimum monthly mean of the leaf area index (based on 12 monthly means)
	16	Land cover	gvf_max (−)	maximum monthly mean of the green vegetation fraction (based on 12 monthly means)
	17	Land cover	gvf_diff (−)	difference between the maximum and minimum monthly mean of the green vegetation fraction (based on 12 monthly means)
	18	Soil attributes	soil_depth_pelletier (m)	depth to bedrock (maximum 50 m)
	19	Soil attributes	soil_depth_statsgo (m)	soil depth (maximum 1.5 m; layers marked as water and bedrock were excluded)
	20	Soil attributes	soil_porosity (−)	volumetric porosity (saturated volumetric water content estimated using a multiple linear regression based on sand and clay fraction for the layers marked as USDA soil texture class and a default value (0.9) for layers marked as organic material; layers marked as water, bedrock, and “other” were excluded)

Water Resources Research

10.1029/2025WR040173

Table 1
Continued

Type	No.	Category	Attributes and unit	Description
	21	Soil attributes	soil_conductivity (cm/h)	saturated hydraulic conductivity (estimated using a multiple linear regression based on sand and clay fraction for the layers marked as USDA soil texture class and a default value (36 cm h−1) for layers marked as organic material; layers marked as water, bedrock, and "other" were excluded)
	22	Soil attributes	max_water_content (m)	maximum water content (combination of porosity and soil_depth_statsgo; layers marked as water, bedrock, and "other" were excluded)
	23	Soil attributes	sand_frac (%)	sand fraction (of the soil material smaller than 2 mm; layers marked as organic material, water, bedrock, and "other" were excluded)
	24	Soil attributes	silt_frac (%)	silt fraction (of the soil material smaller than 2 mm; layers marked as organic material, water, bedrock, and "other" were excluded)
	25	Soil attributes	clay_frac (%)	clay fraction (of the soil material smaller than 2 mm; layers marked as organic material, water, bedrock, and "other" were excluded)
	26	Geological attributes	carbonate_rocks_frac (−)	fraction of the catchment area characterized as "carbonate sedimentary rocks"
	27	Geological attributes	geol_permeability (m^2)	subsurface permeability (log10)

$$h_{i,t} = h(h_{i,t-1}, Z_{i,t-1}, X_{i,t}, \varnothing) \tag{3}$$

where $h_{i,t}$ is the output of a multi-layer RNN constructed by an LSTM cell which is parametrized by θ. Given a time series as a conditioning range, we can obtain h_{i,t_0-1} by Equation 3 as the initial state. For the simulation range, we can sample $\tilde{Z}_{i,t}$ by $P(.|\varnothing(\tilde{h}_{i,t-1}, \varnothing))$ where $\tilde{h}_{i,t} = h(h_{i,t-1}, \tilde{Z}_{i,t}, X_{i,t-1}, \varnothing)$. The samples achieved in this way could be used to compute several statistics such as the mean and quantile.

DeepAR's core architecture builds on the same concept as the encoder-decoder structure. Instead of designing two separate modules, a single module for both the encoder and decoder phases is designed with shared weight matrices and parameters. In our modeling application, DeepAR is trained to output a one-step-ahead simulation at each unfolding of the LSTM. During the encoding phase, the module receives the values in the conditioning range, one at a time, of the previous time step $z_{i,t-1}$ and covariates at the current time step $x_{k,t}$ and outputs a one step ahead simulation $\hat{Z}_{i,t}$. The model is autoregressive in that it uses past values as inputs to the next layer to generate future values in the inference phase. DeepAR also incorporates a group-dependent embedding vector, which picks up group-specific properties for each time series.

Figure 2 illustrates the workflow of the DeepAR model used in this study for a multi-step sequential simulation. The left section of the figure displays the training phase (encoder), where the network receives the covariates $x_{k,t}$, the previous target values $z_{i,t}$, where $t < t_0$ and outputs the hidden state $h_{t,i}$ which is used to compute the one-step simulation $\hat{z}_{i,t+1}$. During DeepAR training, the outputs $\hat{Z}_{i,t}$ are used to compute the loss function and tune the parameters Θ of the model. The right part of the figure displays the inference phase (decoder), $t < t_0$ of the model. DeepAR receives information of previous values through the encoded state, outputs the parameters of a distribution, draws a sample $\hat{Z}_{i,t}$ and passes that sample forward to the next LSTM layer until the end of the simulation window is achieved. A pass of the above process is called a sample trace. By performing Monte-Carlo sampling, DeepAR can sample multiple traces to estimate a joint predictive distribution, which yields the target median, confidence intervals, and quantiles of interest.

2.3.2. TFT Model

Google recently developed the TFT as an attention-based DNN model for multi-horizon prediction, which is the prediction of variables of interest at multiple time steps. TFT is built to explicitly align the model with a broad multi-horizon forecasting task, resulting in greater accuracy and interpretability across a wide range of applications. Interpretability in this algorithm can be achieved by designing the internal structure of neural network models more transparent, revealing the features and concepts it has learned. TFT architecture combines a recurrent LSTM layer to capture local sequential dependencies with self-attention mechanisms to model longer-term relationships across time steps in a parallelizable manner. In other words, TFT integrates both local processing and global processing to handle temporal dependencies. The local dependencies are captured through the

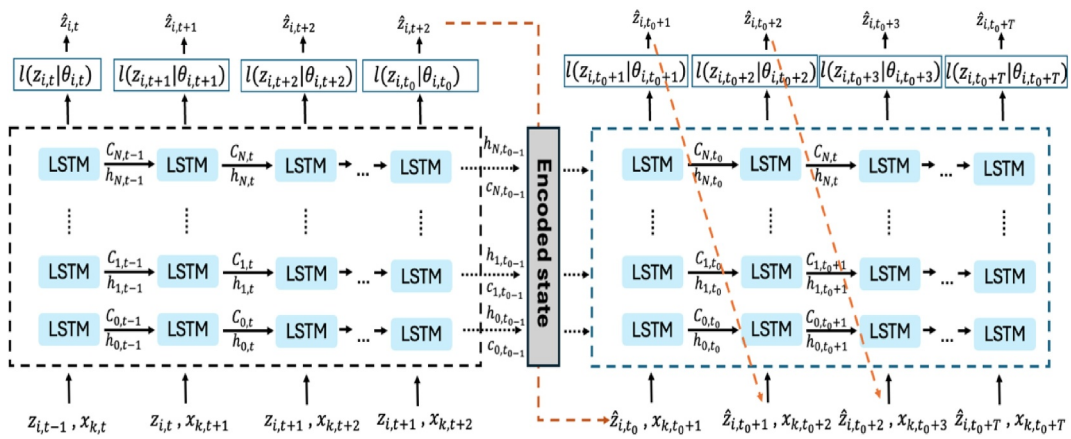

Figure 2. The workflow of DeepAR developed in this study. $h_{i,t}$ is the output of a multi-layer RNN constructed by an LSTM cell which is parametrized by θ. $Z_{i,t}$ is the value of time series i at time t, while $C_{N,t}$ denotes the cell state of the LSTM cell.

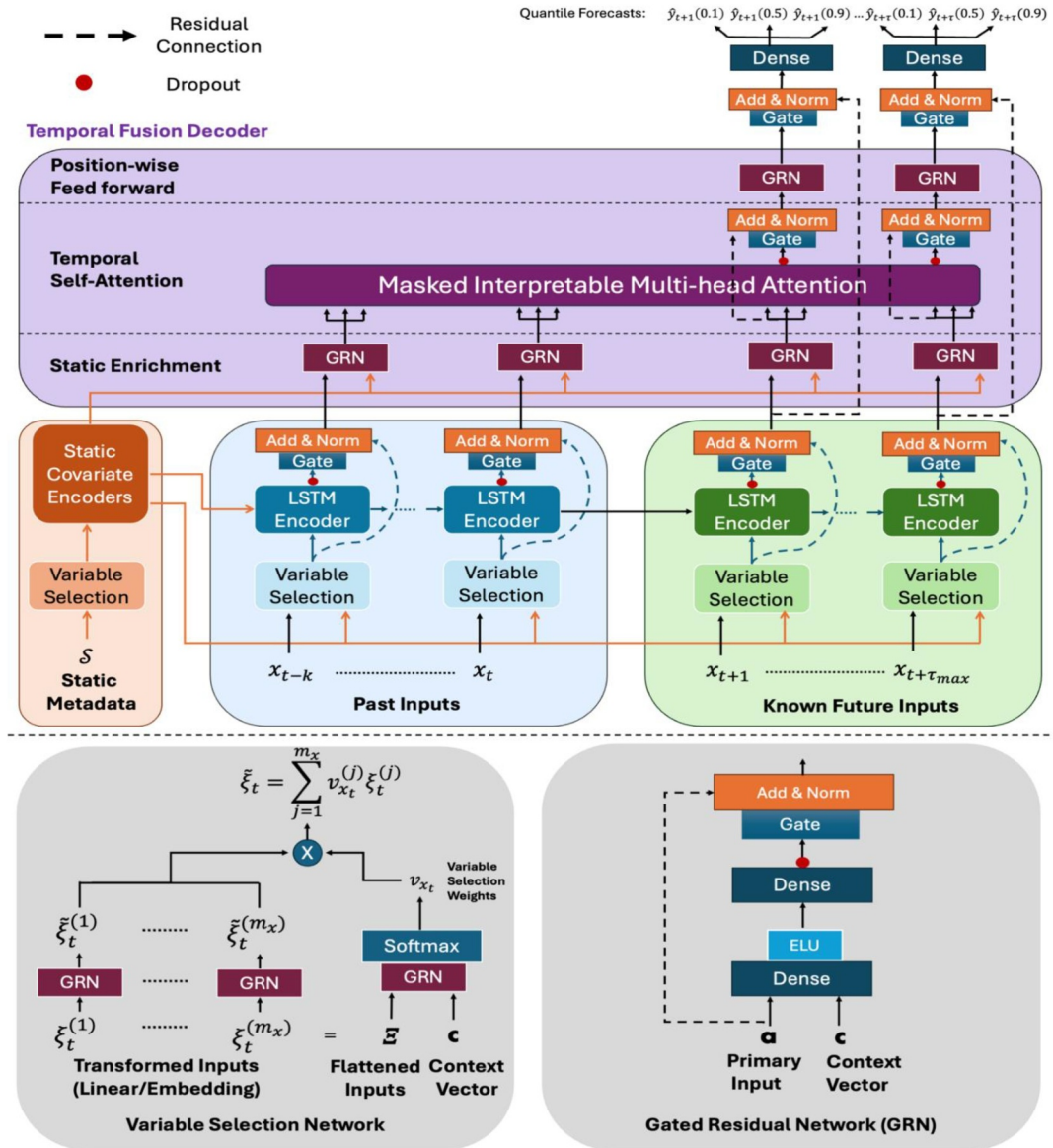

Figure 3. The algorithmic workflow of the TFT developed in this study (partially adapted from Lim et al. (2021)).

LSTM encoder, which processes short-term correlations in the time series. In contrast, global dependencies are captured through multi-head attention, which enables the model to focus on long-term relationships across all time steps in the sequence. This hybrid approach enhances the model's ability to handle complex and varied temporal patterns, making it particularly effective for streamflow prediction tasks. TFT supports three types of features, including (a) temporal data with known inputs into the future, (b) temporal data known only up to the present, and (c) catchment static attributes. These features support training on multiple time series, coming from different distributions. To achieve this, the TFT architecture splits processing into two parts: (a) local processing, which focuses on the characteristics of specific events, and (b) global processing, which captures the collective characteristics of all-time series data. By taking advantage of self-attention, TFT presents a novel multi-head attention mechanism (see Figure 3) which provides extra insight into attribute importance. The major components of TFT include.

2.3.2.1. Gating Mechanisms

TFT uses a Gated Residual Network (GRN) to skip over any unneeded components, allowing for flexible depth and network complexities to suit a wide range of data sets (Lim et al., 2021). GRN motivates TFT to apply non-linear processing only where needed (Lim et al., 2021). The GRN takes in a primary input a and an optional context vector c and yields:

$$\mathrm{GRN}_{\omega}(a,c) = \mathrm{LayerNorm}\left(a + \mathrm{GLU}_{\omega}\left(\eta_1\right)\right) \tag{4}$$

$$\eta_1 = \left(W_{1,\omega}\eta_2 + b_{1,\omega}\right) \tag{5}$$

$$\eta_2 = ELU(W_{2,\omega}a + W_{3,\omega}c + b_{2,\omega}) \tag{6}$$

Where ELU is the Exponential Linear Unit activation function (Clevert et al., 2015), η_1 and η_2 are intermediate layers, LayerNorm is the standard layer normalization function and ω is an index to denote weight sharing (see Lim et al., 2021). Component gating layers were used based on Gated Linear Units (GLUs; Dauphin et al., 2017) to provide the flexibility to suppress any parts of the architecture that are not required for a given data set.

2.3.2.2. Variable Selection Network (VSN)

At each time step, the VSN provides a selection of important input variables (see Equation 7). While traditional DNNs may overfit irrelevant features, attention-based variable selection can help enhance generalization by pushing the model to focus the majority of its learning capacity on the most important feature (Lim et al., 2021).

$$v_{X,t} = \mathrm{Softmax}(\mathrm{GRN}_{v_X}(\Xi_t, c_s)) \tag{7}$$

In Equation 7, Ξ_t presents transformed inputs at time step t, c_s is a constant vector, GRN is the GRN, and softmax denotes the transfer function.

2.3.2.3. Static Covariate Encoders

This approach incorporates static covariates to regulate the temporal dynamics in the TFT modeling (Lim et al., 2021). The static covariate encoders learn context vectors from static metadata and inject them at different locations of the TFT modeling network, through three mechanisms: (a) temporal variable selection, (b) local processing of temporal representations in the Sequence-to-Sequence layer, and (c) static enrichment of temporal representations. These mechanisms allow the conditioning of the temporal representation learning with static information through encoding context vectors to condition temporal dynamics.

2.3.2.4. Temporal Processing

Temporal processing learns both long- and short-term temporal associations by incorporating dynamic attributes into the TFT algorithmic structure that are both observed and known. Local processing is handled by a Sequence-to-Sequence layer, which benefits from its inductive bias for ordered information processing. On the other hand, long-term dependencies are handled by a unique interpretable multi-head attention block mechanism (Equations 8 and 9). This mechanism can shorten the effective path length of information, as any previous step containing relevant data can be targeted immediately (Lim et al., 2021).

$$\mathrm{InterpretableMultiHead}(Q,K,V) = \tilde{H}W_h \tag{8}$$

$$\tilde{H} = \frac{1}{m_H}\sum_{h=1}^{m_H}\mathrm{Attention}\left(QW_Q^{(h)}, KW_K^{(h)}, VW_v\right) \tag{9}$$

Where V is the attention mechanisms scale values, and K *and* Q denote related keys and queries, W_h linearly combining outputs concatenated from all heads H_h. W_Q, W_K and W_V denote head-specific weights for keys, queries, and values, respectively. m_H is the number of heads.

2.3.2.5. Temporal Fusion Decoder

The temporal fusion decoder uses a series of layers described below to learn temporal relationships presented in the data set.

- Locality Enhancement with Sequence-to-Sequence Layer

In rainfall-runoff data, meaningful points are often identified relative to their surroundings, such as anomalies, change points, or cyclical patterns. Utilizing local context through feature construction, which incorporates pattern information alongside individual values, can enhance the performance of attention-based architectures. For instance, D. Li et al. (2021) adopt a single convolutional layer for locality enhancement, extracting local patterns using the same filter across all time. However, this might not be suitable for cases when observed inputs exist, due to the differences in past and future inputs. As such, TFT uses a Sequence-to-Sequence layer to naturally handle these differences.

- Static Enrichment Layer

TFT leverages a static enrichment layer to enhance temporal features with static metadata. For a given position index n, static enrichment takes the form:

$$\theta(t,n) = \mathrm{GRN}_{\theta}\left(\tilde{\varphi}(t,n), c_e\right) \tag{10}$$

where the weights of GRN_{θ} are shared across the entire layer, and c_e is a context vector from a static covariate encoder.

- Temporal Self-Attention Layer

TFT leverages self-attention layers for learning long-term dependencies. In this algorithm, all static-enriched temporal features are first grouped into a single matrix $\Theta(t)$, and interpretable multi-head attention is applied at each simulation time (see D. Li et al., 2021):

$$B(t) = \mathrm{InterpretableMultiHead}(\Theta(t), \Theta(t), \Theta(t)) \tag{11}$$

Decoder masking (D. Li et al., 2021; Vaswani et al., 2017) is also applied to the multi-head attention layer to ensure that each temporal dimension can only attend to the features preceding it. TFT inputs static metadata, time-varying past inputs, and time-varying priori known future inputs. Variable selection is used for the judicious selection of the most salient features based on the input. Gated information is added as a residual input, followed by normalization. GRN blocks enable efficient information flow with skip connections and gating layers. Time-dependent processing is based on the LSTMs for local processing, and multi-head attention for integrating information from any time step (see Lim et al., 2021).

2.3.2.6. TFT Training Procedure

Considering Figure 3, for a given timestep t, a lookback window k, and a $\tau_{\max}$ step ahead window, where $t \in [t-k, \ldots, t+\tau_{\max}]$, the model takes as input, (a) observed past inputs x in the time period $[t-k..t]$, (b) future known inputs x in the time period $[t+1..t+\tau_{\max}]$, and (c) a set of static variables s. The target variable y also spans the time window $[t+1..t+\tau_{\max}]$. TFT is composed of different components, including LSTM blocks, GRN blocks, and VSNs. GRN has two dense layers, and two types of activation functions called ELU and GLU (see Figure 3). Both of these activation functions help the network understand which input transformations are simple or more complex. The final output passes through standard layer normalization. The GRN also contains a residual connection, meaning that the network is able to learn or skip the input entirely. In some cases, depending on where the GRN is situated, the network can also make use of static variables.

The VSN network proposes a feature selection mechanism (see Figure 3). Since all input time series do not have a complex pattern, the model is able to distinguish discerning features from noisy ones. In addition, TFT uses three instances of the VSN for the three types of inputs discussed above. Each instance has different weights (marked with different colors in Figure 3). Indeed, the VSN utilizes GRN under the hood for its filtering capabilities. At the time t, the flattened vector of all past inputs (called Ξ_t) of the corresponding lookback period was fed through a GRN unit and then a softmax function, producing a normalized vector of weights u. Moreover, each feature passes

through its own GRN, which leads to the creation of a processed vector called ξ_t, one for every variable. Finally, output is calculated as a linear combination of ξ_t and u. Note that each feature has its own GRN, but the GRN for each feature is the same across all time steps during the same lookback period. The VSN for static variables does not take into account the context vector c.

The input that is passed through VSN has properly encoded and weighed the features. However, since the input is time-series data, the model should also make sense of the time and sequential ordering. Consequently, the first goal of the LSTM encoder-decoder module is to produce context-aware embeddings, which are denoted as ϕ. This is similar to the positional encoding used in the classic transformer, where sine and cosine signals are added to positional embeddings. In this setting, TFT, however, utilizes the LSTM encoder-decoder instead as the model should account for all types of input. The known past inputs are fed into the encoder, while the known future inputs are fed into the decoder. For the static inputs, TFT applies the LSTM encoder-decoder proposed by Karpathy and Fei-Fei (2015) to correctly condition the input based on exogenous data.

In the final step, TFT applies a well-known self-attention mechanism proposed by Vaswani et al. (2017), which helps the model learn long-term dependencies across different time steps. TFT proposes a novel interpretable multi-head attention mechanism, which, contrary to the standard implementation, provides feature interpretability. Indeed, TFT's multi-head attention adds a new matrix or grouping such that the different heads share some weights which can be interpreted in terms of seasonal analysis. In this study, feature importance was measured by analyzing the weights u of all VSN modules across the entire test set. This created an interpretable multi-head attention layer to calculate the persistent temporal patterns in data which determined the most important time steps during the lookback period for the TFT training.

2.4. Experimental Design, Interpretability, and Uncertainty Quantification

When building a DNN-driven rainfall-runoff architecture, it is necessary to provide the network with information on catchment characteristics which allow the model to discriminate different catchment settings. Ideally, the network should be able to condition the processing of the dynamic inputs on a set of catchment static attributes. In this process, the network learns a mapping function from meteorological forcing into streamflow values. The mapping function depends on a set of catchment static attributes that can, in principle, be incorporated anywhere in the modeling domain. To this end, we built probabilistic, physics-guided DeepAR and TFT that learn catchment similarities directly from meteorological forcing data and ancillary data of multiple basins. We evaluated these modeling performances in a "gauged" setting, meaning that we never ask the network to predict rainfall-runoff process on unseen data. Because the model is trained using both static and dynamic attributes, it can learn how to combine different parts of the network to simulate various rainfall–runoff behaviors. In principle, this approach explicitly allows for sharing some parts of the networks for similar behaving basins while using different independent parts for basins with completely different rainfall-runoff behavior. Considering the large spatial extent of the study area and the availability of a relatively small number of gauges, it was necessary to build a physics-guided model to simulate the all-season hydrology of a large area with relatively small inputs. Our methodology provides a mapping function from catchment attribute space into a learned, high-dimensional space in which catchments with similar rainfall–runoff behavior can be clustered together.

The static and dynamic attributes (see Table 1) were incorporated separately into the architecture to assign them a particular task. This approach, so called physics-guided DeepAR and TFT, explicitly differentiates between similar types of dynamical behaviors (i.e., rainfall–runoff processes) that differ between individual entities in the catchment. After training, the static input gate of the network contains a series of real values in a range of [0, 1] that allows certain parts of the input gate to be active through the simulation of any individual catchment.

Model training was performed based on the water year starting from 1 October 1989 through 30 September 1999. The models and benchmark validation were performed from 1 October 1999 through 30 September 2008. We trained both DeepAR and TFT using calibration data from all basins and evaluated the results using validation data. This structure implies that a single parameter set per model was trained to work across all basins. In this study, four modeling configurations were performed and tested including (a) TFT with static attributes, hereafter physics-guided TFT, (b) TFT without catchment static attributes or original TFT, (c) DeepAR with static attributes, hereafter physics-guided DeepAR, and (d) DeepAR without catchment static attributes or original DeepAR. To construct the original DeepAR and TFT, a single model of each network was trained on a combined calibration data from all basins, using only the meteorological forcing data while physics-guided DeepAR and

TFT models were trained based on combined calibration data of all basins using meteorological forcing data as well as static attributes. The catchment attributes were incorporated into the static input gate, while the meteorological inputs were fed into the rest of the network structure.

Once the original and physics-guided TFT and DeepAR configurations were prepared, we demonstrated how our modeling design allowed for analysis of its individual components to interpret the rainfall-runoff relationships it has learned. This study demonstrated two interpretability cases: (a) examining the sensitivity of each catchment attributes in simulation and (b) capturing persistent temporal patterns in observed and predicted daily streamflow data. The interpretability of our proposed configurations focused on the ways to aggregate the patterns across the entire data set—extracting generalizable insights about temporal dynamics in rainfall-runoff records.

An objective function is required for training the network. For regression tasks such as rainfall-runoff simulation, the mean-squared error (MSE) is commonly used. In addition, quantile regression, which is an extension of standard linear regression, can be used to estimate the conditional median of the target variable when assumptions of linear regression are not met. Apart from the median, quantile regression can also calculate the 0.025 and 0.975 quantiles so called 95% simulation uncertainty (95PPU), which means the model has the ability to output a simulation interval around the actual simulation. All four configurations were calibrated using the quantile regression likelihood function (or loss function; see Equation 12).

Given y and $\hat{y}$ the actual value and the simulation, respectively, and q a value for the quantile between 0 and 1, the quantile loss function is defined as:

$$QL(y, \hat{y}, q) = \max[q(y - \hat{y}), (1 - q)(y - \hat{y})] \tag{12}$$

As the value of q increases, overestimation is penalized by a larger factor compared to underestimation. For instance, for q equal to 0.75, overestimation will be penalized by a factor of 0.75, and underestimation by a factor of 0.25. This is how simulation intervals are created to assess uncertainty. There are two main types of uncertainties in modeling: epistemic (model uncertainty) and aleatoric (data uncertainty). In this study, we followed the approach of Kendall and Gal (2017) to quantify uncertainties by modeling aleatoric uncertainty through input perturbations, while simultaneously employing a quantile regression loss function to capture epistemic uncertainty during model training (also see Gal & Ghahramani, 2016; Tabas & Samadi, 2022).

2.5. Benchmark Hydrologic Models

We used conceptual to physics-based rainfall-runoff models to benchmark DeepAR and TFT and borrowed a set of existing hydrologic models gathered by Kratzert, Klotz, et al. (2019) that were configured and calibrated by previous studies using CAMELS data. These models are (a) Sacramento Soil Moisture Accounting Model (SAC-SMA; Burnash & Ferral, 1973) coupled with the Snow-17 snow routine (Anderson, 1973), (b) Variable Infiltration Capacity (VIC; Liang, 1994), (c) Framework for Understanding Structural Errors (FUSE) with three different model structures of 900, 902, 904 (Clark et al., 2008; Henn et al., 2015), (d) Hydrologiska Byråns Vattenbalansavdelning (HBV; Seibert and Vis, 2012), and (e) mesoscale Hydrologic Model (mHM; Kumar et al., 2013; Samaniego et al., 2010).

Each set of simulations that we used for benchmarking is documented elsewhere in the literature (references below). Each of these benchmark hydrologic models used Maurer forcing data, the same input data that we used to set up DeepAR and TFT models. For a fair comparison, all models were calibrated and validated in the same time period. These benchmark hydrologic models can be distinguished into two different groups. The first group is those models that were calibrated for each basin individually. They are SAC-SMA (Newman et al., 2017), VIC (Newman et al., 2017), FUSE, mHM (Mizukami et al., 2019), and HBV (Seibert et al., 2018). The HBV model supplied both lower and upper benchmarks, where the lower benchmark is an ensemble mean of 1000 uncalibrated HBV models, whereas the upper benchmark is an ensemble of 100 calibrated HBV models. The second group is those models that were regionally calibrated. These models share one parameter set for all basins in the data set. The second group comprises VIC (Mizukami et al., 2017) and mHM (Rakovec et al., 2019) simulations. Readers are referred to Newman et al. (2017), Mizukami et al. (2017, 2019), Seibert et al. (2018), and Rakovec et al. (2019) for more information on these benchmark modeling simulations.

2.6. Performance Assessment Metrics

This study used a variety of performance metrics for model benchmarking including the Nash-Sutcliffe efficiency (NSE; Nash & Sutcliffe, 1970; Equation 13 in Table 2), $\alpha-$NSE decomposition (Equation 14 in Table 2, Gupta et al., 2009) and $\beta-$NSE decomposition (Equation 15 in Table 2; Gupta et al., 2009). These metrics focus specifically on assessing overall performance using a decomposition of the standard squared error metrics that are less sensitive to bias (Gupta et al., 2009). In addition, three different signatures of the flow duration curve (FDC) were used to evaluate the performance of specific ranges of discharge simulations. Following Yilmaz et al. (2008), we partitioned the FDC into three different segments (a) high-flow segment (0–0.02 flow exceedance probabilities) characterizing watershed response to large precipitation events, (b) mid-flow segment (0.2–0.7 flow exceedance probabilities) that specifies by flows from moderate size precipitation events and also related to the intermediate-term primary and secondary base flow relaxation response of the watershed system, and (c) low-flow segment (0.7–1.0, flow exceedance probabilities), related to the long-term sustainability of flow and controlled by the interaction of baseflow with riparian evapotranspiration during extended dry periods (see Equations 16–18 in Table 2).

To quantify the goodness of uncertainty estimation, two indices were used, that is, P-factor which is the percentage of data bracketed by a 95PPU (Abbaspour et al., 2007; Equation 19 in Table 2), and R-factor, which is the average width of the uncertainty band divided by the standard deviation of the corresponding measured variable (the minimum value is zero; Abbaspour et al., 2007; Equation 20 in Table 2). Ideally, a P-factor of 0.95 indicates that 95% of observations fall within the predictive interval, aligning with the confidence level, while an R-factor close to or below 1.0 indicates a tight uncertainty band. Although previous studies suggested that $P > 0.7$ and $R < 1.5$ are acceptable thresholds, we emphasize that such broad ranges may limit the practical interpretability of models. Therefore, in this study, we treat P-factors near 0.95 and R-factors near or below 1.25 as an indicative of high-quality uncertainty quantification. The total uncertainty index (TUI) is also calculated based on the P-factor and R-factor for each model (Equation 21 in Table 2). In addition, a probability plot in a continuous fashion suggested by Laio and Tamea (2007) was used to illustrate the uncertainty quantification (see Figure 4). A deviation from the 1:1 line shows the expected calibration error and the sum of which is referred to as the miscalibration area (e.g., Naeini et al., 2015; Tabas and Samadi, 2022).

Table 2
Overview of the Evaluation Metrics Used in This Study

	Eq. Number	Equation
Performance Evaluation Metrics	(13)	$\mathrm{NSE} = 1 - \frac{\sum_{i=1}^{n}(Qs_i - Qo_i)^2}{\sum_{i=1}^{n}(Qo_i - \overline{Qo})^2}$
	(14)	$\alpha - \mathrm{NSE} = \sigma_s/\sigma_o$
	(15)	$\beta - \mathrm{NSE} = (\mu_s - \mu_o)/\sigma_o$
	(16)	$\%\mathrm{BiasFHV} = \frac{\sum_{h=1}^{H}(Qs_h - Qo_h)}{\sum_{h=1}^{H} Qo_h} \times 100$
	(17)	$\%\mathrm{BiasFMS} = \frac{[\log(Qs_{m1}) - \log(Qs_{m2})] - [\log(Qo_{m1}) - \log(Qo_{m2})]}{[\log(Qo_{m1}) - \log(Qo_{m2})]} \times 100$
	(18)	$\%\mathrm{BiasFLV} = -1 \cdot \frac{\sum_{l=1}^{L}[\log(Qs_l) - \log(Qs_L)] - \sum_{l=1}^{L}[\log(Qo_l) - \log(Qo_L)]}{\sum_{l=1}^{L}[\log(Qo_l) - \log(Qo_L)]} \times 100$
Uncertainty Assessment Metrics	(19)	$P - Factor = \frac{\text{Observations bracketed by 95PPU}}{\text{Number of observations}} \times 100$
	(20)	$R - \mathrm{Factor} = \frac{\frac{1}{k}\sum_{i=1}^{k}(X_U - X_L)}{\sigma_x}$
	(21)	$\mathrm{TUI} = \frac{P_{\mathrm{Factor}}}{R_{\mathrm{Factor}}}$

Note. The notation of the original publications is kept.

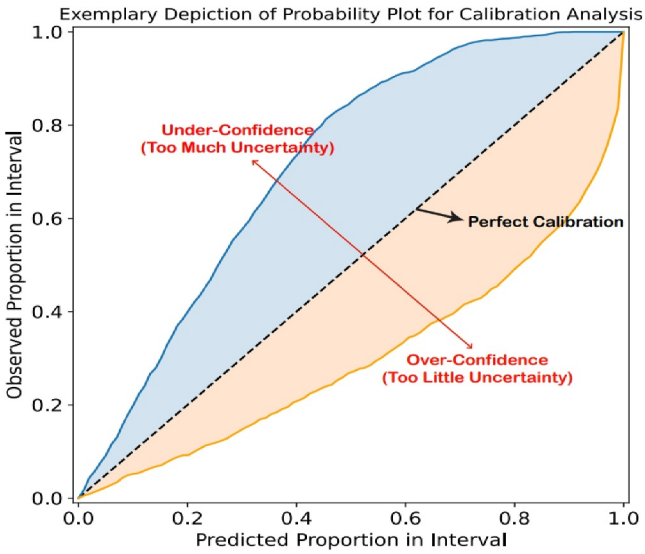

Figure 4. The probability plot for the evaluation of predictive distributions. The *x*-axis shows the estimated cumulative distribution over all time steps by a given model, and the *y*-axis shows the actual observed cumulative probability distribution. A conditional probability distribution was produced by each model for each time step in each basin. A hypothetically perfect model will have a probability plot that falls on the 1:1 line.

3. Results and Discussion

3.1. Hyperparameter Tuning

The probabilistic nature of DeepAR and TFT accounts for stochasticity in the network initialization and optimization procedures. Common hyperparameters that need to be optimized by tuning include learning rate, batch size, dropout rate, or even network parameters like the number of layers in a network or the pooling strategy. In this study, hyperparameter optimization is conducted using searching over a pre-defined search space with the same number of iterations across all modeling configurations for a given data set. We used the Optuna model (Akiba et al., 2019) to find optimal values for the networks' structure as well as hyperparameters. Optuna is an automatic hyperparameter optimization algorithm, particularly designed to dynamically construct the search spaces for hyperparameter optimization. Optuna enables efficient hyperparameter optimization by adopting state-of-the-art algorithms for sampling hyperparameters and pruning efficiently unpromising trials. Hyperparameter optimization was conducted via Optuna search, using 100 epochs for TFT and DeepAR. Dropout was applied in TFT before the gating layer and layer normalization while zoneout regularization or layer normalization was applied in DeepAR to regulate the network dropout value. Zoneout is a regularization method that stochastically forces some hidden units to maintain their previous values. This technique improves training, while balances robustness to batch size variations by randomly preserving hidden activations during training and improving generalization. Like dropout, zoneout uses random noise to train a pseudo-ensemble and improve network regulation. This was performed for the DeepAR configurations by preserving instead of dropping hidden units, gradient information, and state information that was more readily propagated through time, as in feed-forward stochastic depth networks.

We selected the maximum number of epochs and the number of trials equal to 100 and 200 for each modeling configuration. The number of hidden layers ranged from 10 to 100 layers, dropout ranged from 0.1 to 0.3, and the learning rate ranged from 0.0001 to 0.3. Considering Optuna results, we assumed similar values for the number of hidden layers (50 layers) and the dropout rate equal to 0.2, for all modeling configurations. In the case of the learning rate, we considered different values derived from Optuna test results for each modeling setup (see Figure 5). Optuna automatic tuning searches the hyperparameters space that resulted in minimizing the loss function.

There were other hyperparameters whose optimal values were adjusted manually or by trial- error due to computational cost. For example, the size of the attention head and the number of continuous hidden layers of the TFT model were selected as 1 and 8, respectively. Also, two LSTM layers in the DeepAR configurations presented the best performance. In addition, distribution functions were programmed as the likelihood (or loss) function in the DeepAR model. In this study, the quantile regression loss function named multivariate quantile distribution loss (MQF2DistributionLoss; Kan et al., 2022) was programmed into the DeepAR algorithmic structure to improve the accuracy of simulation.

3.2. TFT and DeepAR Simulations

This section presents a comparison between four modeling configurations. The results of benchmark hydrologic models are also presented to make a comparison with the performances of the probabilistic DeepAR and TFT models. The key results of DeepAR and TFT approaches are illustrated in Figure 6, with the cumulative density functions (CDFs) of NSE values for all four modeling configurations and conceptual to distributed hydrologic models across catchment scale and CONUS.

As expected, incorporating catchment static and dynamic attributes into the algorithmic structures improved the overall modeling performance of TFT and DeepAR models compared to the original configurations. It is important to note that some of the errors in the DeepAR and TFT models are due to randomness in the training

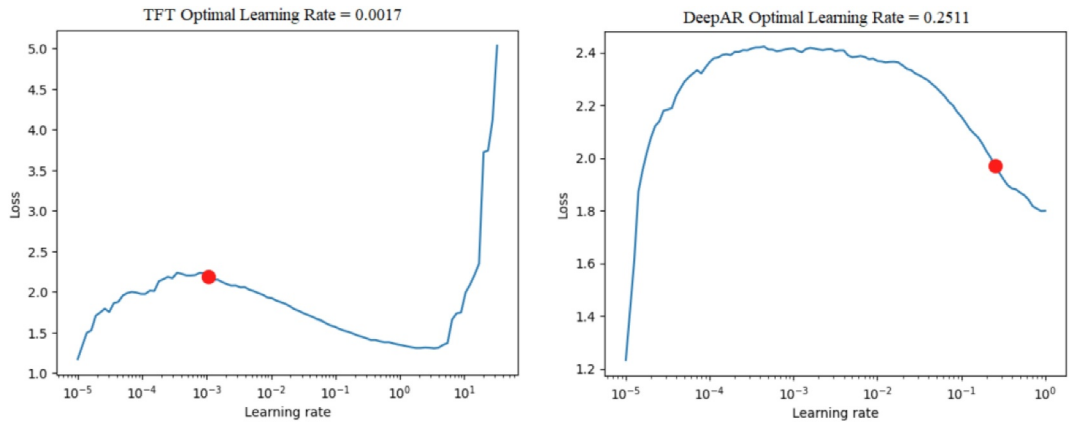

Figure 5. Optimal learning rate values (red dot) driven from Optuna test for the TFT and DeepAR algorithms.

procedure. Although, the probabilistic nature of the models and the use of the Optuna algorithm for hyper-parameter tuning significantly mitigated this error. As shown in Figure 6, there is a slight difference between physics-guided DeepAR and TFT models and the original configurations. The overall mean NSE value across catchment scales improved significantly compared with the original configurations by approximately 0.04 in both TFT and DeepAR models (see Table 3). Overall, the physics-guided TFT model performed (NSE = 0.741) as good as the Entity-Aware LSTM (EA-LSTM) ensemble (NSE = 0.742) of Kratzert, Klotz, et al. (2019). The original TFT (NSE = 0.704) also demonstrated a relatively close performance to the LSTM ensemble (0.758). On the other hand, the physics-guided DeepAR (NSE = 0.689) and the original DeepAR (NSE = 0.648) performed somewhat weaker than the LSTM-based models. Similar trends were observed for the FDC metrics, indicating that the models exhibited consistent performance with the findings of Kratzert, Klotz, et al. (2019).

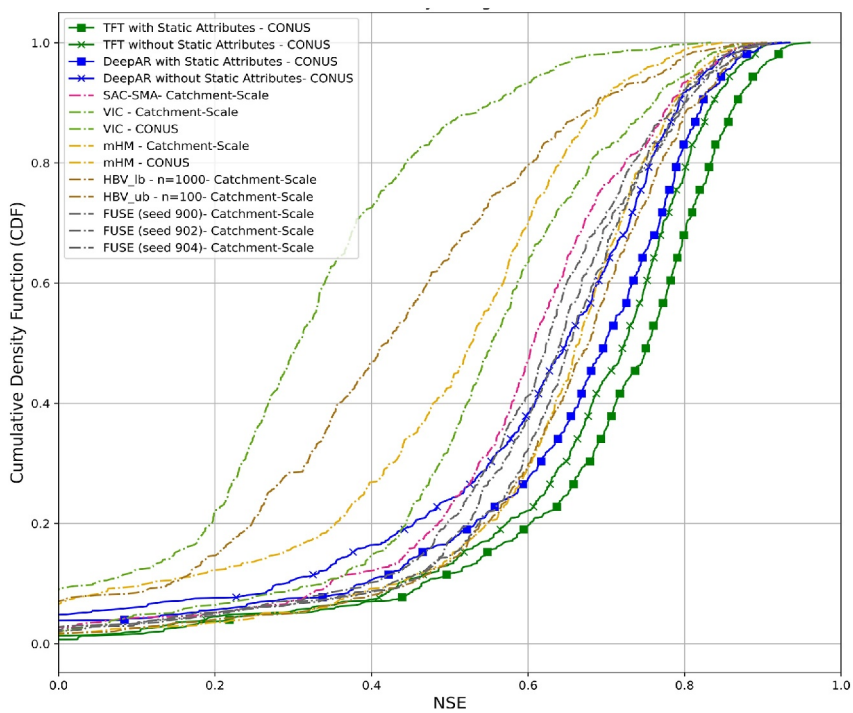

Figure 6. CDFs of the specific NSE values for all four configurations as well as calibrated conceptual to physics-based hydrologic models at catchment scale and across CONUS.

Table 3
A Comparison of the Original and Physics-Guided TFT and DeepAR Performance Versus Benchmark Hydrologic Simulation Results

Model	Calibration type	NSE (median)	α-NSE (median)	β-NSE (median)	FHV (median)	FMS (median)	FLV (median)
Physics-guided TFT	CONUS	0.741	0.81	−0.02	−15.24	−8.46	25.37
Original TFT	CONUS	0.704	0.81	−0.02	−17.55	−14.38	30.39
Physics-guided DeepAR	CONUS	0.689	0.80	−0.02	−18.01	−19.18	67.65
Original DeepAR	CONUS	0.648	0.80	−0.04	−18.95	−20.56	70.45
SAC-SMA	Catchment Scale	0.603	0.78	−0.07	−20.40	−14.30	37.30
VIC	Catchment Scale	0.551	0.72	−0.02	−28.10	−6.60	−70.00
VIC	CONUS	0.307	0.46	−0.07	−56.50	−28.00	17.40
mHM	Catchment Scale	0.666	0.81	−0.04	−18.60	−7.20	11.40
mHM	CONUS	0.527	0.59	−0.04	−40.20	−30.40	36.40
HBV (lower)	Catchment Scale	0.416	0.58	−0.02	−41.90	−15.90	23.90
HBV (upper)	Catchment Scale	0.676	0.79	−0.01	−18.50	−24.90	18.30
FUSE (900)	Catchment Scale	0.639	0.80	−0.03	−18.90	−5.10	−11.40
FUSE (902)	Catchment Scale	0.65	0.80	−0.05	−19.40	9.60	−33.20
FUSE (904)	Catchment Scale	0.622	0.78	−0.07	−21.40	15.50	−66.70

The overall model performance due to the inclusion of catchment attributes implies that these attributes contain information that can provide additional learning features to distinguish different catchment-specific rainfall–runoff behaviors. A significant performance improvement was observed when catchment static and dynamic attributes were incorporated into the models compared to the original configurations due to a significant increase in the number of tunable parameters in the network. Both TFT and DeepAR results showed that physical attributes slightly improved (NSE increased by ∼0.04) the simulation performances. Both networks used different algorithmic conceptualizations to encourage the models to learn physically relevant representations of the rainfall-runoff mechanisms. For example, DeepAR learned group-level time series patterns through a categorical grouping feature that was embedded into the network to learn group-level time series patterns. Indeed, the model learned an embedding vector of size, "*embedding_dimension*", for each group, allowing the network to capture the common rainfall-runoff behaviors of catchments. In TFT, the GRN blocks were able to weed out the unimportant and unused inputs, which decreased the tunable model parameters and helped the model recognize important catchment physical attributes. This is important because the TFT decoder enables the interpretability of results, including static and dynamic attributes, which helps to understand which attributes have strong control over rainfall-runoff generation mechanisms.

Overall, analysis suggests that the physics-guided TFT model outperformed DeepAR in both physics-guided and the original configurations (see Table 3). This is because TFT used two different mechanisms for long and short-term pattern recognitions in rainfall-runoff simulation. First a Sequence-to-Sequence encoder/decoder and the LSTM blocks which summarized shorter rainfall-runoff patterns in data weighed the importance of each input feature. Second, a temporal self-attention decoder that learned how long-term dependencies present within the data set (i.e., seasonality) can prioritize the most relevant patterns. The use of these specialized mechanisms also facilitated interpretability of the results such as identifying the importance and sensitivity of exogenous variables (or catchment physical attributes) for the simulation problem and persistent temporal patterns that are discussed in Section 3.4.

3.3. Conceptual to Physics-Based Hydrologic Simulations

The DeepAR and TFT models were first compared with the VIC and mHM models that were regionally calibrated. Specifically, each model was calibrated using a single set of transfer functions that mapped out catchment static attributes to model parameters. The procedure for parameterizing these models for regional simulations is described in detail by Mizukami et al. (2017) and Rakovec et al. (2019). Figure 6 shows both TFT and DeepAR results which outperformed regionally calibrated conceptual to physics-based hydrologic models by a large

margin. Even the original DeepAR and TFT that are trained without catchment physical attributes consistently outperformed regionally calibrated hydrologic models. The median NSE score across the catchments for the physics-guided TFT model was 0.74. In contrast, VIC showed a median NSE of 0.31 while the mHM presented a median NSE of 0.53 which can be categorized as unsatisfactory performance.

Figure 6 compares CDFs of the catchment scale NSE values for all benchmark models across CONUS. Table 3 contains the performance metrics for benchmark hydrologic models as well as the TFT and DeepAR models. Analysis suggested that the TFT model significantly outperformed all hydrologic models even without the catchment's physical attribute incorporation. The two best-performing hydrologic models were the ensemble ($n = 100$) of catchment-calibrated HBV models with a median NSE score equal to 0.67, and a single catchment-calibrated mHM model with a median NSE score equal to 0.66. In addition, the physics-guided DeepAR model outperformed both HBV and mHM models while the performance of the original DeepAR with a median NSE of 0.65 was comparable with the hydrologic modeling simulations.

Overall, physics-guided TFT outperformed both conceptual and physics-based hydrologic models, although TFT performance was not very skillful in calibrating intermediate and low flow values. This is because TFT has no exponential outflow function, and thus the simulation value can be easily dropped to minuscule numbers. In traditional hydrologic models, the flow dynamics—especially for low flows—are often governed by exponential outflow functions that mimic the gradual depletion of water storage in natural systems, such as aquifers and reservoirs. TFT, being a data-driven model, lacks this type of built-in physical mechanism, making it prone to dropping predicted streamflow values to extremely small (minuscule) numbers during low-flow conditions. This can lead to unrealistic underestimations of FLVs. When the predicted streamflow approaches near-zero or minuscule values, it can distort downstream metrics like FDCs, negatively affect hydrological performance assessments, and reduce the model's reliability during simulation. To overcome the limitations, we incorporated a programmatic adjustment inspired by Tabas and Samadi (2022). This adjustment involves introducing an additional parameter to constrain the simulated streamflow during low-flow conditions. Specifically, a parameter was introduced to enforce a lower bound on the simulated streamflow values. This lower bound was defined as the minimum observed flow in the data set for each basin or catchment (meaning that the model won't generate streamflow values lower than the minimum defined baseflow for that specific catchment). This ensures that the simulated flow does not drop below physically realistic levels, which aligns the model's simulations more closely with catchment hydrological behavior.

During post-processing or directly within the TFT model's output, any predicted daily streamflow value greater than zero but below the minimum observed flow was adjusted upward to match the minimum observed flow. This adjustment acts as a safeguard to prevent unphysical underestimation during low-flow periods. By imposing this constraint, the TFT model achieved better performance in calibrating intermediate and low-flow values. The adjustment further improved the FDC representation, particularly in the lower quantiles and reduced errors in key metrics such as FLV for low-flow conditions. Please note that, while the adjustment improved calibration, it did not fundamentally alter the TFT architecture to incorporate physical processes explicitly. Future work could explore integrating exponential outflow functions or storage-discharge relationships directly into the model's structure to make it inherently more hydrologically consistent. Figure 7 presents the spectrum of NSE values for all 531 catchments across CONUS per each modeling configuration.

As illustrated in Figure 7, in a few modeling configurations, there were several catchments with very low NSE values, particularly in the Great Plains, although, incorporating static and dynamic attributes into the modeling architectures enhanced modeling performance and reduced the number of poor-performing catchments in this region. In TFT, the components responsible for capturing temporal relationships in daily streamflow time series data, local processing, and self-attention layers, have the largest impact on performance by increasing or decreasing loss functions. While local processing is critical in the rainfall-runoff generation mechanism, the higher performance of TFT configurations compared to DeepAR indicates the fact that this algorithm can be advantageous in daily streamflow simulations due to its self-attention layer that plays a more vital role in simulation. A possible explanation for poor performance in the Great Plains catchments is related to persistent variability and seasonality of streamflow records that might dominate other temporal patterns in the data sets. However, the diversity across CAMELS daily streamflow time series data sets can also have a significant effect on the respective temporal variability. Another reason for poor performance in some catchments is related to error and uncertainty associated with simulation which is discussed in Section 3.5.

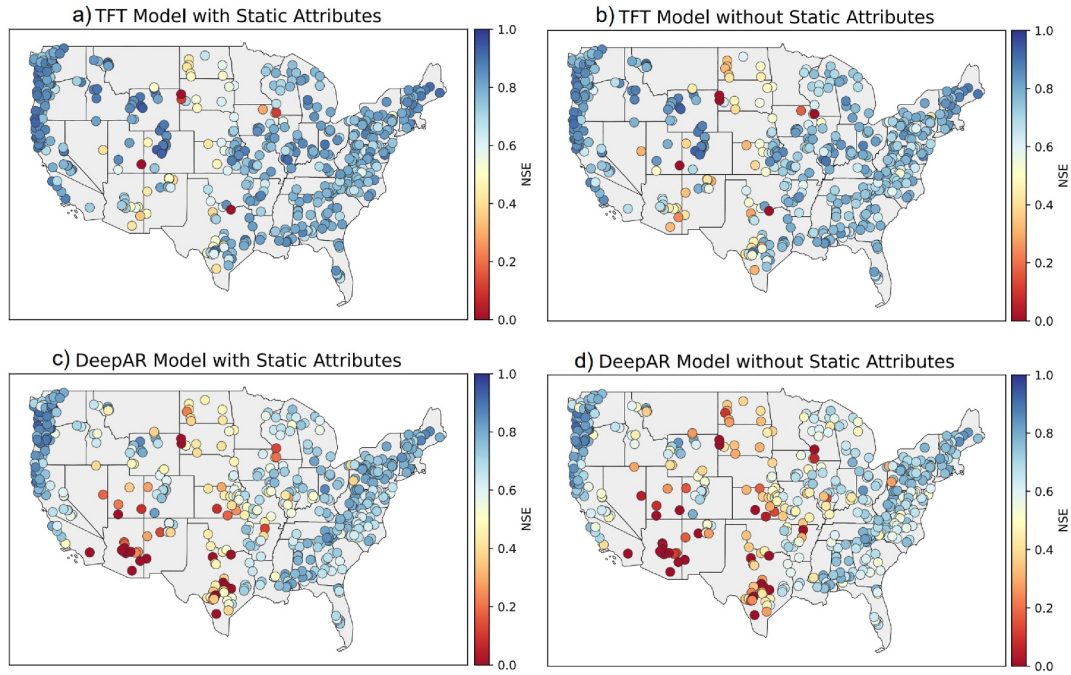

Figure 7. The NSE values for all 531 catchments across CONUS per each modeling configuration. As illustrated TFT and somewhat DeepAR provided better simulation results for the east and west coast catchments while their performance degraded in central US particularly in Midwest.

3.4. Interpretability of Simulations

We demonstrated interpretability to audit the simulation and enforce fairness and descriptive accuracy in the simulation process. We defined interpretability as the process of extracting relevant knowledge from rainfall-runoff relationships that allow the model to learn the data patterns during training. It should be noted that interpretability was assessed for the TFT results only since the DeepAR model was unable to provide interpretability results due to its recurrent structure. We demonstrated two interpretability analyses: (a) examining the sensitivity of each static and dynamic attributes in simulations; and (b) capturing persistent temporal patterns in observation and simulations. Our interpretability analysis focused on examining the sensitivity of catchment attributes and defining proper mechanisms to aggregate the patterns across the entire data set to understand how attention-based architecture can provide insights into temporal rainfall-runoff dynamics. TFT's multi-head attention adds a new matrix/grouping such that the different heads share some weights, which then can be interpreted in terms of sensitivity analysis. To preserve interpretability, we embraced a single interpretable multi-head attention layer only.

Figures 8–11 present the sensitivity of catchment static attributes during the TFT validation period across CONUS. These analyses were derived by normalizing the sensitivity measures per catchment to a range of [0, 100] considering all 27 catchment static attributes explained in Table 1. As illustrated, the most sensitive catchment static attributes across CONUS are (a) geological attributes including subsurface permeability (log10), soil depth, and the fraction of the catchment area characterized as "carbonate sedimentary rocks," (b) climate indices such as mean precipitation, the average frequency of dry days and seasonality and timing of precipitation and (c) land cover attributes such as forest fraction and maximum monthly mean of the green vegetation fraction. Sensitivity analysis further revealed that mean daily precipitation (*p-mean*) and mean daily potential evapotranspiration (*petmean*) are more sensitive in coastal catchments across CONUS. As shown, aridity (*PET/P*; *a ratio of mean PET*) seems to control rainfall-runoff behaviors in those catchments with less than average precipitation compared to the coastal regions (see Figure 8). Interestingly, our results qualitatively agree with much of the analysis presented by Addor et al. (2018).

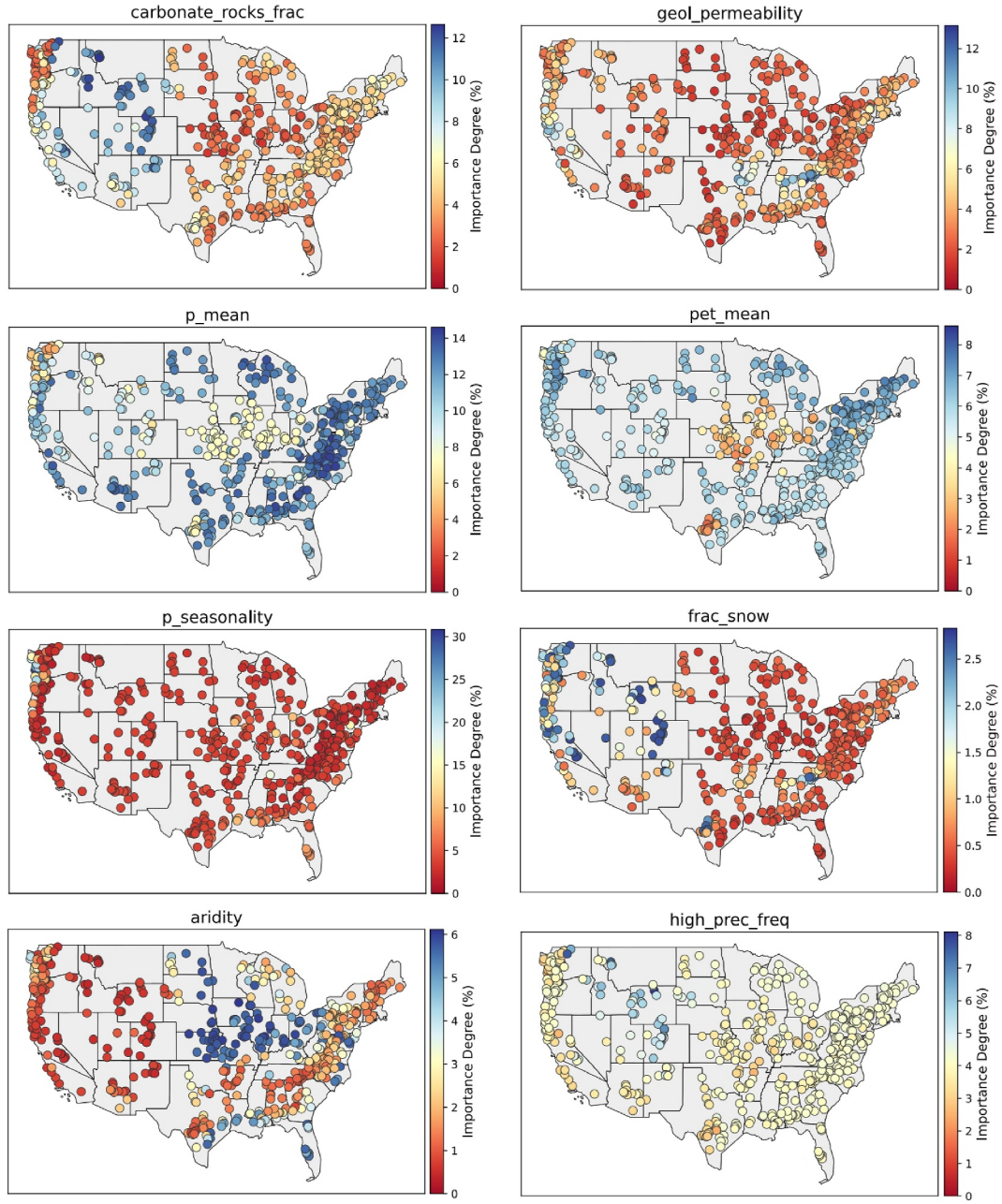

Figure 8. The spatial variability and degree of sensitivity (%) of catchment static attributes across CONUS including aridity, seasonality and timing of precipitation (*p_seasonality*), frequency of high precipitation, the fraction of precipitation falling as snow, mean precipitation, mean daily PET, geological permeability, and the fraction of the catchment area characterized as "carbonate sedimentary rocks."

In the case of dynamic attributes, the sensitivity results were driven by the decoder part of the physics-guided TFT model. This analysis provided the sensitivity (%) for each dynamic attribute per catchment across CONUS, illustrated in Figure 12. We observed that among multiple dynamic attributes, *VP*, precipitation, and minimum temperature showed high sensitivity while the time index showed low sensitivity. Specifically, precipitation was more sensitive in the majority of catchments across CONUS while it was moderately sensitive in the west coast and the Rocky Mountains regions. The sensitivity of *VP* refers to changes in atmosphere conditions in response to variations in temperature and humidity. *VP* is considered a key factor in understanding the impacts of drought and water deficit on streamflow variability.

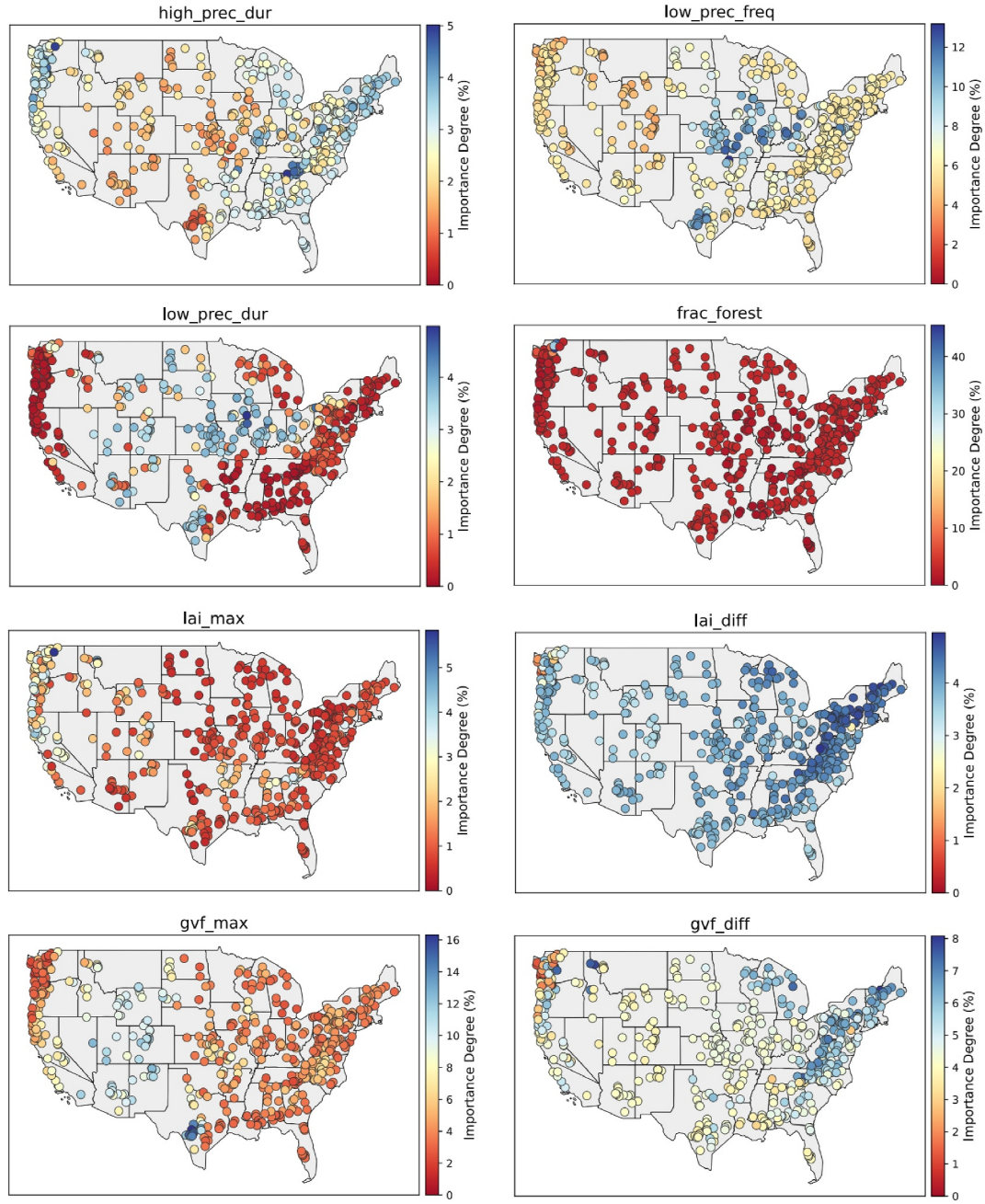

Figure 9. The spatial variability and degree of sensitivity (%) of catchment static attributes across CONUS including average duration of high precipitation events, frequency of dry days, forest fraction, average duration of dry periods, maximum monthly mean of the leaf area index, difference between the maximum and minimum monthly mean of the leaf area index, maximum monthly mean of the green vegetation fraction, and mean of the green vegetation fraction.

Overall, sensitivity analysis indicated that the rainfall-runoff process in most catchments is dominated by climate attributes such as mean daily precipitation, seasonality and timing of precipitation (*p-seasonality*). Meteorological patterns such as mean precipitation showed less sensitivity when moving away from the Appalachians toward the Great Plains. This is likely because elevation and slope begin to play less of a role in precipitation generation. However, the frequency of dry days (*low-prec-freq*) attributed more sensitivity to runoff generation in the Great Plains. In the Rocky Mountains, most catchments were sensitive to the fraction of the catchment area characterized as "carbonate sedimentary rocks" and forest fraction (*frac-forest*), with moderate sensitivity to the frequency of dry days (*low-prec-freq*) in the New Mexico region. Similar to the east coast, the sensitivity of

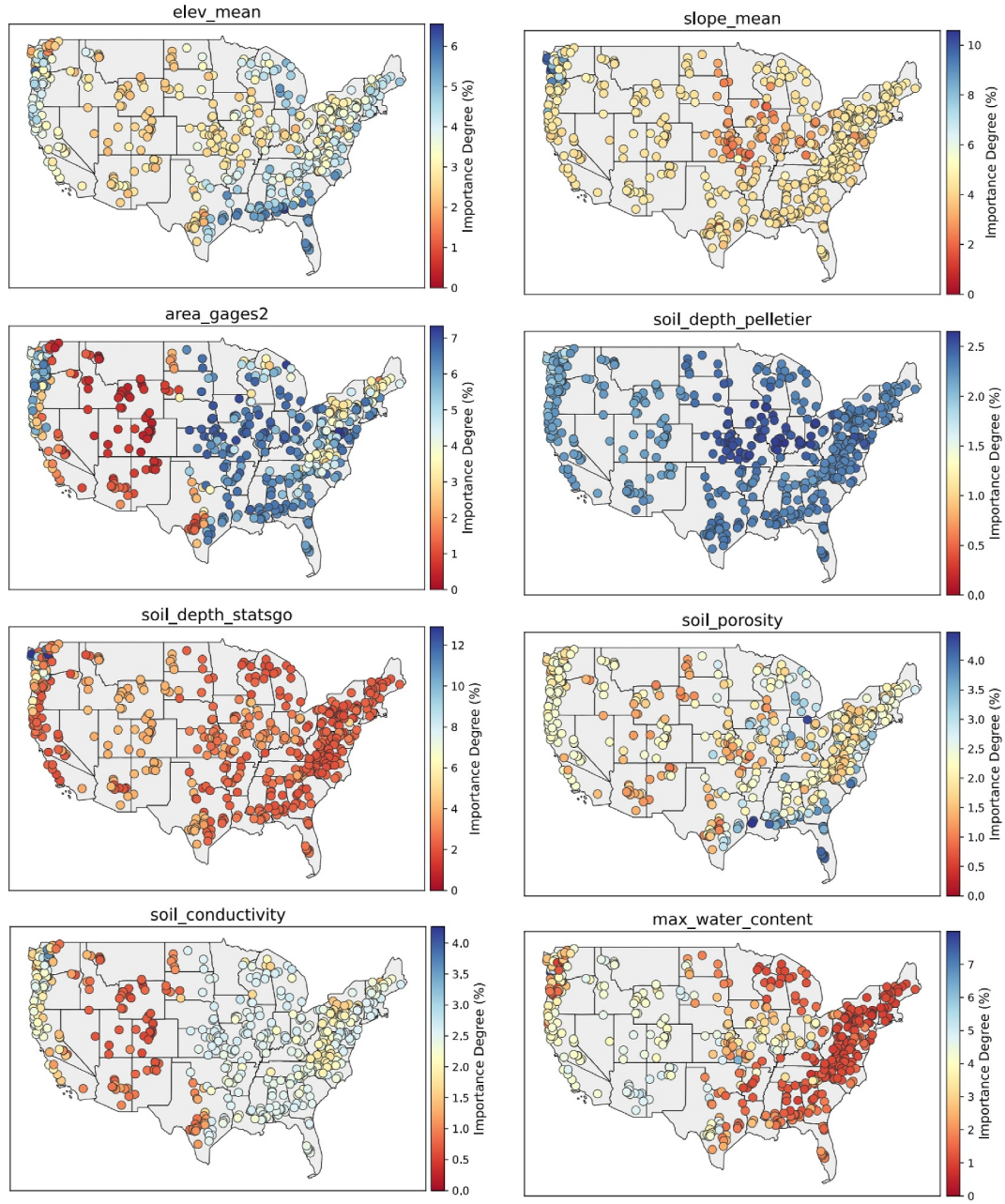

Figure 10. The spatial variability and degree of sensitivity (%) of catchment static attributes across CONUS including catchment mean elevation, catchment mean slope, catchment area, soil depth, soil volumetric porosity, saturated hydraulic conductivity, and maximum water content.

catchment static attributes on the west coast was dominated by climatic attributes such as mean daily precipitation and seasonality as well as the timing of precipitation (*p-seasonality*). Figure 13 illustrates the most sensitive static and dynamic attributes per catchment.

In addition, the sensitivity of dynamic attributes in most catchments in the Appalachian Mountains and the eastern US were generally dominated by the amount of precipitation (*prcp*) and minimum daily temperature (*tmin*) values. Precipitation showed low sensitivity to rainfall-runoff processes as we moved away from the Appalachians toward the Great Plains. Again, this is because elevation and slope begin to play less of a role, and minimum daily temperature attributed more weights to rainfall-runoff generation mechanism. In the Rocky Mountains, *VP* was the most dominant dynamic attribute in most of the catchments. While on the west coast, solar

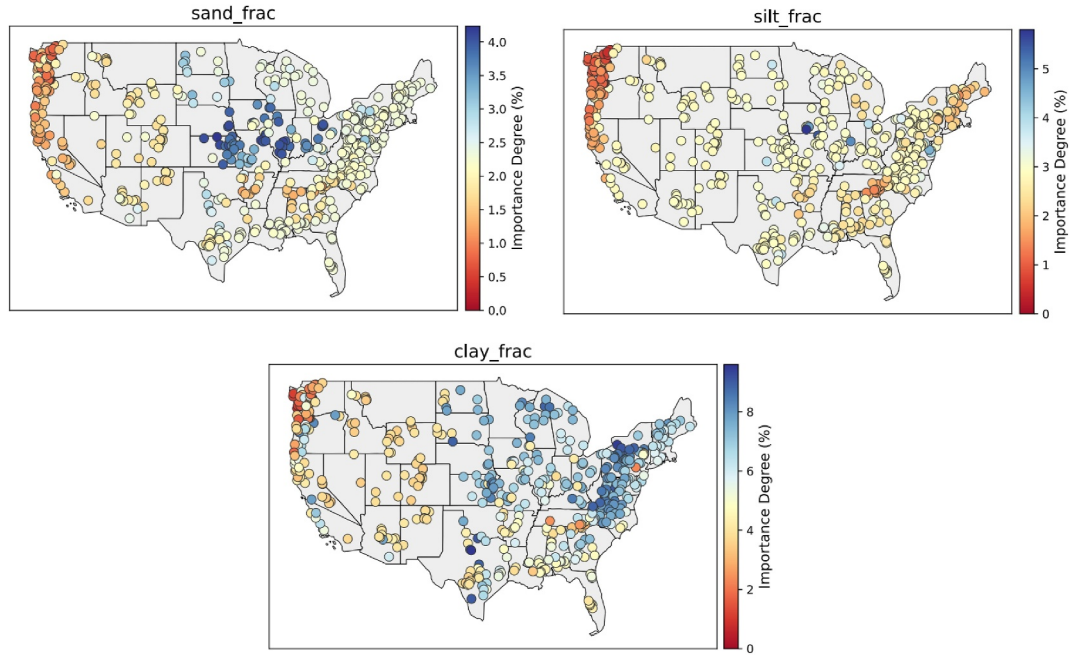

Figure 11. The spatial variability and degree of sensitivity (%) of catchment static attributes across CONUS including sand fraction, silt fraction, and clay fraction.

radiation (*srad*) and maximum temperature (*tmax*) were the most important dynamic attributes. In the south, the minimum daily temperature was sensitive and contributed more weight to rainfall-runoff simulations. A highly sensitive *VP* reflects the fact that small changes in maximum temperature and humidity can lead to significant shifts in drought conditions and water availability in the region.

3.5. Uncertainty Assessment

We evaluated the robustness of the trained TFT model to the noises arising from catchment attributes by adding Gaussian perturbations $(0, \sigma)$ to the standardized attribute values across a range of $\sigma \in [0, 1]$. This analysis does not aim to quantify aleatoric uncertainty directly, but rather to assess the model's sensitivity to noisy or imperfect input features. Aleatoric uncertainty is instead captured by the model's built-in probabilistic forecasting framework, which outputs prediction intervals based on learned distributions. We added Gaussian noise $N\,(0, \sigma)$ with increasing standard deviation (σ) to the individual attribute values and assessed the resulting changes in modeling performance for each noise level. Concretely, additive noises were drawn from normal distribution with various selected standard deviations in a range of [0, 1]. Next, static and dynamic attributes were standardized with zero mean and unit variance before training; thereby these perturbations were independent from the units or relative magnitudes of the individual catchment attributes. As expected, the model performance degraded with increasing noise in the catchment static attributes. However, the degradation did not occur abruptly, but smoothly with increasing level of noise, which is an indication that the model is not overfitted when the static and dynamic attributes were incorporated into the algorithmic structure. This also indicates that the model did not remember each basin status with its set of attributes precisely but rather learned a smooth mapping function between the attributes and model output. It is interesting to note that, the perturbation noise was always relative to the overall standard deviation of the static and dynamic attributes across all catchments, which was always $\sigma = 1$ (i.e., all input features were normalized prior to training). When noise with a small standard deviation was added ($\sigma < 0.2$) to the features, the mean and median NSE were relatively stable. However, the median NSE decreased from 0.79 without noise to 0.65 with an added noise equal to the total variance of the input features ($\sigma = 1$). Similar results were also obtained by Kratzert, Klotz, et al. (2019) when they applied a noise with $\sigma = 1$ to the attributes. In addition, Tabas and Samadi, (2022) calibrated the magnitude of the noise through an advanced procedure of obtaining the noise standard deviation magnitude from the loss function for the recurrent models. Similarly, they found an optimal range of [0, 0.2] for noise standard deviation. As a result, a Gaussian noise of

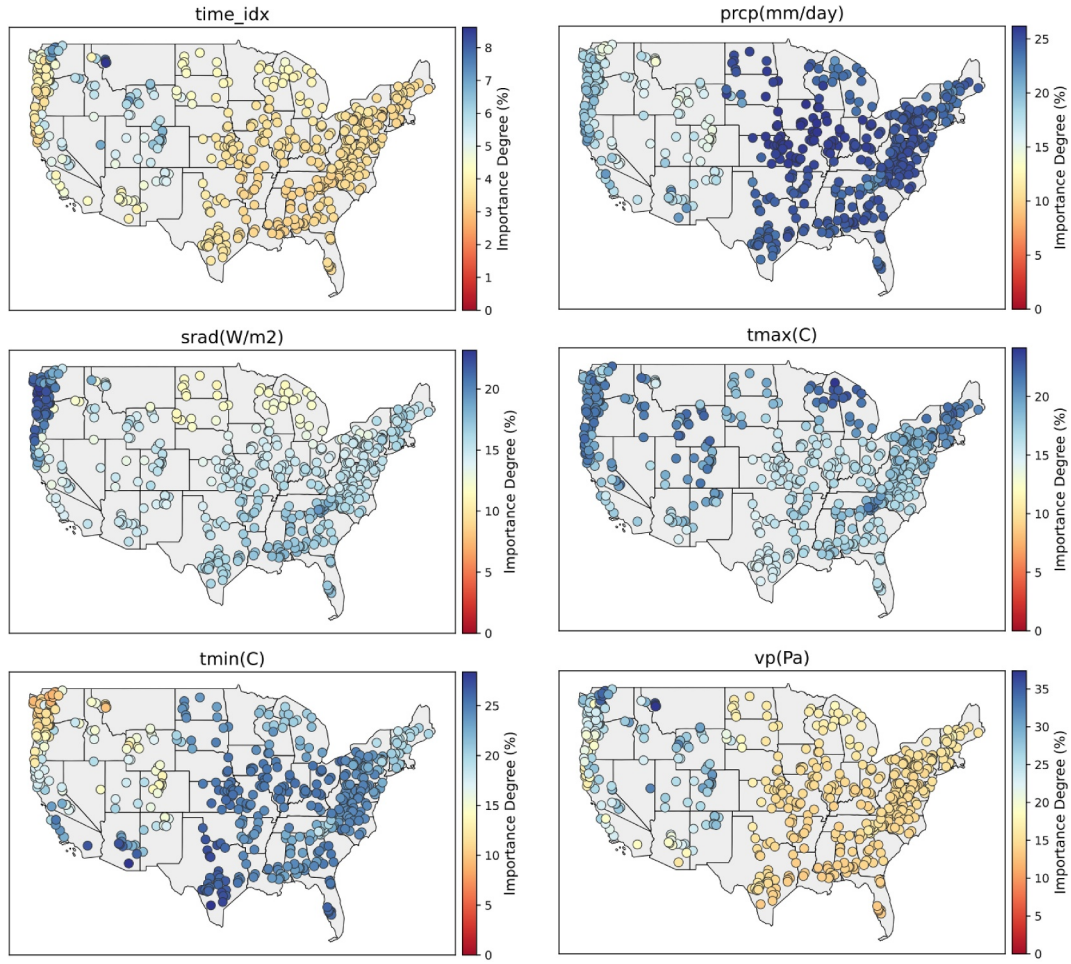

Figure 12. The spatial variability and degree of sensitivity (%) of dynamic attributes across CONUS, including time index, precipitation, solar radiation, maximum temperature, minimum temperature, and *VP*.

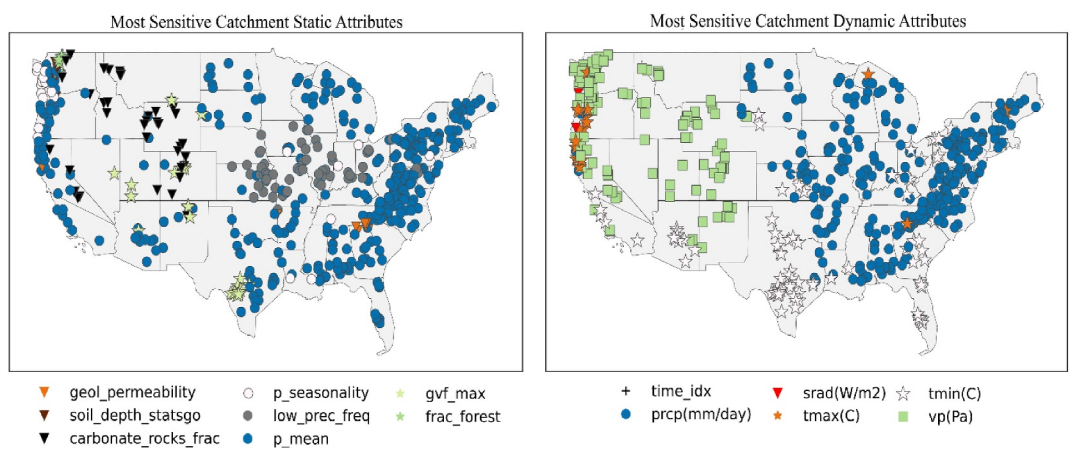

Figure 13. The most sensitive static and dynamic attributes of each catchment driven from physics-guided TFT simulations.

Table 4
The Performance of the Physics-Guided TFT Model and Uncertainty Assessment for 18 Selected Catchments Across CONUS

		Performance assessment						Uncertainty assessment		
Zone #	Basin ID	NSE	α-NSE	β-NSE	FHV	FMS	FLV	P-factor	R-factor	TUI
1	**01031500**	**0.924**	**1.00**	**0.01**	**1.99**	**−11.58**	**1.43**	**99.73**	**1.11**	**0.90**
2	02018000	0.860	0.81	−0.04	−19.25	−1.89	41.83	98.77	0.88	1.13
3	**02177000**	**0.917**	**0.95**	**0.02**	**−2.82**	**−6.42**	**44.59**	**99.59**	**1.24**	**0.81**
4	04027000	0.920	0.95	−0.01	−7.35	−8.89	62.36	96.58	1.16	0.83
5	03241500	0.895	0.93	0.00	−5.79	−16.47	55.40	99.45	1.21	0.83
6	03500000	0.901	0.90	0.05	−9.42	−18.57	55.51	97.26	1.19	0.82
7	05362000	0.831	0.72	−0.04	−30.55	−8.55	68.78	99.66	1.06	0.94
8	08014500	0.795	0.89	0.04	−14.27	−1.43	12.54	96.99	0.79	1.23
9	05057200	0.515	0.42	−0.16	−60.42	−47.24	70.85	95.48	1.2	0.8
10	**06623800**	**0.922**	**0.90**	**−0.01**	**−9.82**	**8.26**	**−79.52**	**99.59**	**1.07**	**0.93**
11	06919500	0.857	0.75	−0.02	−28.06	−42.18	80.44	98.97	0.88	1.12
12	08066200	0.852	0.84	−0.01	−16.18	−37.73	97.10	98.49	0.75	1.32
13	08271000	0.857	0.99	0.10	−9.66	2.37	51.93	95.25	0.75	1.26
14	09066200	0.935	0.96	0.07	−5.03	−17.41	72.46	95.62	1.16	0.82
15	09494000	0.696	0.55	−0.12	−45.44	−43.82	97.76	96.03	1.39	0.69
16	10336645	0.857	0.97	0.07	−11.82	−4.46	74.77	97.12	1.14	0.85
17	13011900	0.960	0.95	0.04	−7.26	−6.17	77.44	88.42	1.07	0.83
18	**11532500**	**0.896**	**0.91**	**0.02**	**−12.67**	**−11.33**	**10.83**	**89.79**	**0.79**	**1.14**
CONUS (Median)	–	0.773	0.86	−0.01	−15.24	−8.46	51.37	96.64	0.97	0.94

Note. The high TUI index indicates the most efficient uncertainty assessment. The uncertainty simulation results of bolded catchments (best performances) are shown in Figure 14.

$N(0, 0.15)$ was added to the individual attribute value and assessed the resulting changes in modeling performance over time.

To quantify aleatoric uncertainty, we used quantile regression to estimate the conditional median of the target variable (Tagasovska and Lopez-Paz, 2019) in the physics-guided TFT model, which can be used when assumptions of linear regression are not met. The ideal procedure to perform quantile regression is to use a special loss function, namely quantile loss. The quantile loss takes a parameter, α, which indicates which quantile should be targeted by the model. In the case of $\alpha = 0.5$, this is equivalent to asking the model to predict the median value of the target, and not the most likely value, which would be the mean. A well-defined strategy would be to produce a confidence interval for each simulation. Indeed, if the lower and upper quantiles of the target are predicted, then a "trust region" to which the true daily streamflow value is likely to belong can be obtained. We selected 0.025 and 0.975 quantiles, the so-called 95PPU, which means the model has the ability to output a 95% confidence interval around the actual simulation. The quantile loss function was selected as the likelihood function of the physics-guided TFT model, trained by minimizing the quantile loss summed across $q \in [0.025, 0.0975]$. To present uncertainty results, we selected 18 catchments among 531 CAMELS catchments across CONUS (one per each HUC2 zone) to assess the data uncertainty associated with the physics-guided TFT simulation. The uncertainty assessment results are presented in Table 4.

Overall, uncertainty results revealed that including a Gaussian noise $N(0, \sigma)$ with increasing standard deviation to the individual attribute was beneficial in reducing errors in TFT simulations. Considering all the catchments, uncertainty metrics (median over CONUS) revealed that the 95PPU interval of physics-guided TFT bracketed most of the observed streamflow data (>95%). Accordingly, quantile regression was efficient in quantifying the average width of the uncertainty band (R-Factor = 1.51) for each streamflow gauging station.

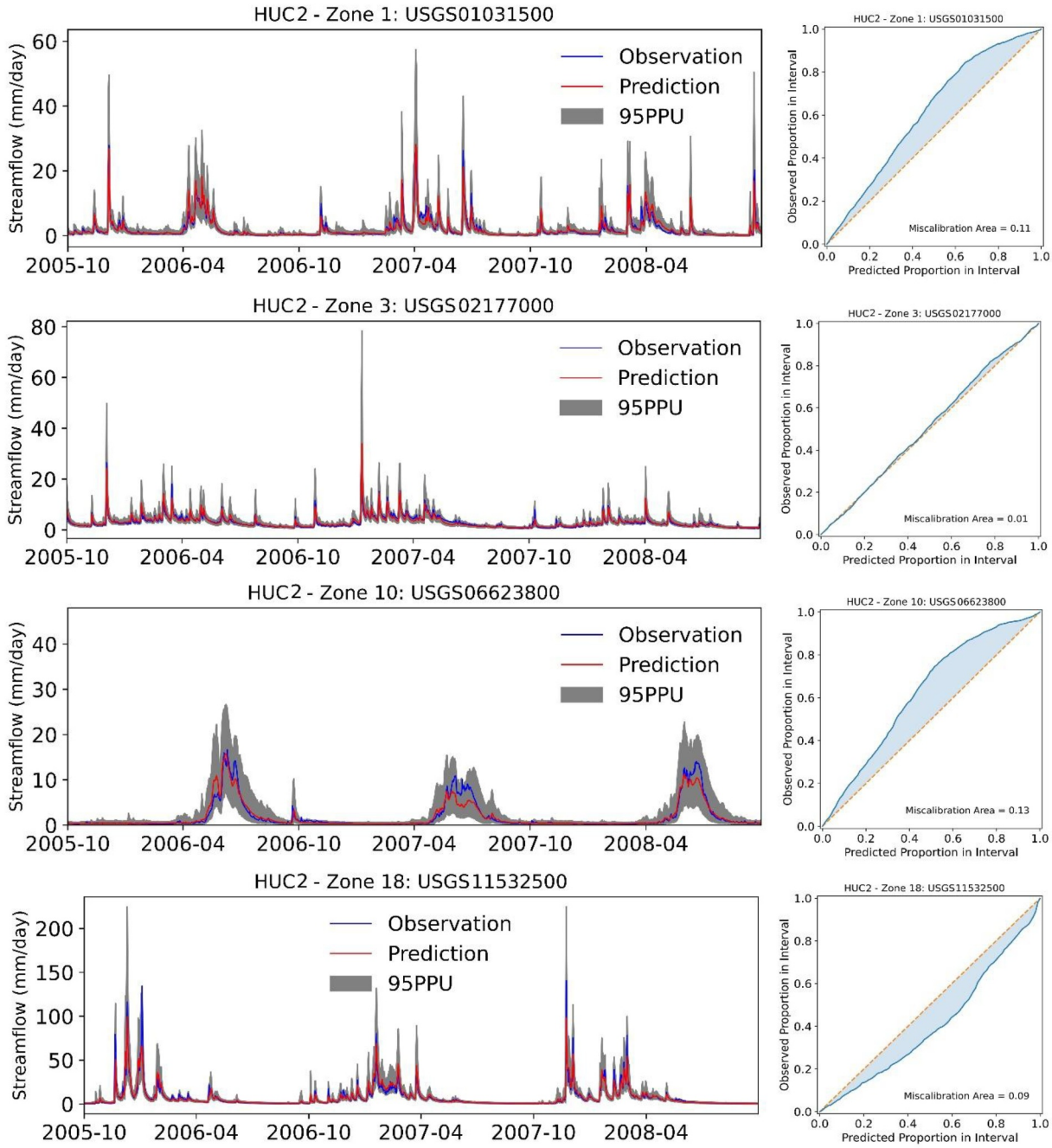

Figure 14. Observed, simulated and 95PPU along with probability plots for the four selected USGS gauging stations.

Figure 14 showed observed, simulated and 95PPU bands along with probability plots for four selected USGS gauging stations, that is, USGS01031500 (Piscataquis River near Dover-Foxcroft, ME), USGS02177000 (Chattooga River Near Clayton, GE), USGS06623800 (Encampment River above Hog Park Center, Near Encampment, WY), and USGS11532500 (Smith River Near Crescent City, CA). Overall, uncertainty analysis revealed that the 95PPU of the physics-guided TFT successfully bracketed 99% of observations with P-factor of >99.50 in these gauging stations. The average width of the uncertainty band was as small as ∼1.2 (R-factor < 1.25) with very good calibration performance (NSE ∼ 0.92). This indicates the fact that physics-guided TFT provided reliable simulations particularly for those catchments with unique and complex rainfall-runoff processes. In the case of USGS11532500, physics-guided TFT bracketed more than 89% of observations (P-factor = 89.79). However, the

average width of the uncertainty band was as small as 0.79 (R-factor = 0.79) which resulted in overconfident simulations (probability plot miscalibration area = 0.09). The quantile regression overestimated uncertainty in USGS01031500 and USGS06623800 while the error was underestimated across USGS11532500. As shown, the physics-guided TFT model provided the best uncertainty assessment for USGS02177000 located in Chattooga River Near Clayton, GA with a hypothetically probability plot that falls on the 1:1 line.

4. Conclusions and Future Works

This study implemented TFT and DeepAR algorithms for interpretable daily rainfall-runoff simulation across CONUS. The novelty of this study lies in multiple fronts. First, we employed TFT for daily streamflow simulation at a continental scale, which to our knowledge has not been extensively explored in prior hydrology literature. Probabilistic framing enables not just point predictions but full predictive distributions, which are critical for operational hydrologic forecasting and risk management. Second, our approach is physics-guided; we incorporated static physical features (e.g., basin characteristics such as area, slope, soil types) that encode important hydrologic properties, along with dynamic meteorological drivers (e.g., precipitation, temperature) into the models, thus grounding the learning process in known physical principles. This physics-aware design enhances model generalization across highly diverse catchments and hydrologic regimes. Third, we provide variable analyses on the relative importance of variables, offering physical insights into the dominant mechanisms controlling streamflow prediction in different regions and seasons. This interpretability moved beyond black box forecasting and contributed toward understanding hydrologic processes from the learned models. Finally, our work extends TFT into a physics-guided, probabilistic, and interpretable framework for large-scale hydrologic simulation, representing an important methodological and practical contribution to the field of hydrologic modeling.

To construct physics-guided, probabilistic, and interpretable framework, catchment static and dynamic attributes were incorporated into the algorithmic structure. The physics-guided configurations were compared with the original models as well as with benchmark hydrologic simulation models. On a wide range of CAMELS data sets, our research demonstrated significant performance improvements of physics-guided TFT over DeepAR as well as benchmark hydrologic models. The physics-guided TFT model showed superior simulation capability capturing both short- and long-terms dependencies in rainfall-runoff records across CONUS. Physics-guided TFT was not only a reliable candidate for both local and regional daily streamflow simulations but also revealed potential deficiencies in current conceptual to physics-based hydrologic modeling structures. For example, in the case of mesoscale models such as VIC and mHM, soil moisture dynamics and variability in infiltration and surface runoff can dominate the simulation results while a lack of soil moisture dynamics to calculate infiltration-excess overland flow in the FUSE model can challenge surface runoff computation, especially in arid climate zones. Incorporating several key catchment attributes such as saturated hydraulic conductivity, soil maximum water content, and mean daily PET into the TFT and DeepAR networks reduced the sensitivity and improved the descriptions of the model. This indicates the fact that these attributes that control loss, water dynamics through a vertical soil profile and storage have a strong control on rainfall-runoff generation mechanisms in many HUC2 zones across CONUS. These attributes determine catchment wetness conditions and soil water transport process, thereby overestimation or underestimation of them may cause abrupt shift in rainfall-runoff magnitude when the catchment's initial abstraction threshold is exceeded.

Hyperparameters, such as embedding dimension and the number of heads/layers showed a large effect on the performance of models. Automated configuration of these hyperparameters using the Optuna algorithm was computationally efficient and resulted in optimal performance. Additionally, adopting a single interpretable multi-head attention layer in the TFT structure enhanced the interpretability of daily streamflow simulations. By applying interpretable multi-head attention at each simulation time, TFT learned the persistent sensitivity patterns of static and dynamic attributes providing insightful explanations about rainfall-runoff dynamics. In this process, TFT alleviated the importance of catchment attributes by using a separate encoder-decoder attention at each time step on top of the self-attention to determine the contribution of time-dependent and time-independent catchment physical attributes in rainfall-runoff generation. For daily streamflow data with varying magnitudes and long-term dependencies, such a capability is expected to be practically useful in discovering which catchment attributes are important and equally which ones can be ignored and removed from the simulation process.

We believe physics-guided TFT will continue to gain popularity and may remain comparable to existing data-driven models such as LSTM as well as traditional hydrologic models. However, more benchmarking on localized streamflow data sets and new methods to render interpretability are needed to fully capture its applicability in hydrologic simulation settings. For example, recent progress in the field of mechanistic interpretability (e.g., activation patching and intrinsic methods) can be adapted for hydrologic time series simulation. Activation patching (Olah et al., 2017) and intrinsic methods (Swamy et al., 2024) are new techniques that can be adapted for mechanistic interpretability to identify causal relationships and pinpoint important model components. Mechanistic interpretability seeks to reverse engineer transformers, similar to how one might reverse engineer a compiled binary computer program.

Although this study successfully addressed interpretability, it is crucial to acknowledge potential limitations. A key limitation is that interpretability often relies on approximations, which may not reflect the complex structure of DNN models (e.g., X. Li et al., 2022). Moreover, the internal structure and parameters of the networks such as weights and biases are not easily understandable and interpretable (Räuker et al., 2023). This lack of transparency may cause significant barriers to understanding DNN's decisions. Our perception about the future of interpretability agrees with the compelling need for a proper understanding of the potential and caveats opened by interpretability techniques discussed herein. It is our vision that model interpretability must be addressed jointly with requirements and constraints related to aleatoric and epistemic uncertainties. An interpretable transformer rainfall-runoff simulation can be only guaranteed if all these principles are studied jointly. This will help water resources managers and stakeholders understand how a data-driven model arrives at specific simulation, fostering transparency, confidence, and trust in the results. Gaining trust in transformer time series simulation is an important step forward for the hydrologic modeling community facing the challenges and opportunities associated with the growing availability of big data and the desire for fair and interpretable data-driven decision-making.

Data Availability Statement

The Data used in this study are from CAMELS (https://ral.ucar.edu/solutions/products/camels), geospatial data from USDA (https://datagateway.nrcs.usda.gov/). The codes can be obtained after publication from the corresponding author upon request.

Acknowledgments
This work was supported by the U.S. National Science Foundation (NSF) Directorate for Engineering under Grant CBET1901646, CBET2429082, and CMMI2219656. All opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. The NCAR is acknowledged for providing CAMELS data free of charge. Clemson University is acknowledged for generous allotment of computing time on the Palmetto cluster.

References

Abbaspour, K. C., Yang, J., Maximov, I., Siber, R., Bogner, K., Mieleitner, J., et al. (2007). Modelling hydrology and water quality in the pre-alpine/alpine Thur watershed using SWAT. *Journal of Hydrology*, *333*(2–4), 413–430. https://doi.org/10.1016/j.jhydrol.2006.09.014

Addor, N., Nearing, G., Prieto, C., Newman, A. J., Le Vine, N., & Clark, M. P. (2018). A ranking of hydrological signatures based on their predictability in space. *Water Resources Research*, *54*(11), 8792–8812. https://doi.org/10.1029/2018wr022606

Addor, N., Newman, A. J., Mizukami, N., & Clark, M. P. (2017). The CAMELS data set: Catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences*, *21*(10), 5293–5313. https://doi.org/10.5194/hess-21-5293-2017

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2623–2631).

Anderson, E. A. (1973). *National weather service river forecast system: Snow accumulation and ablation model*. US Department of Commerce, National Oceanic and Atmospheric Administration.

Berrevoets, J., Kacprzyk, K., Qian, Z., & van der Schaar, M. (2023). Causal deep learning. *arXiv preprint arXiv:2303.02186*.

Burnash, R. J. C., & Ferral, R. L. (1973). *A generalized streamflow simulation system: Conceptual modeling for digital computers*. US Department of Commerce, National Weather Service, and State of California.

Casdagli, M. (1989). Nonlinear prediction of chaotic time series. *Physica D: Nonlinear Phenomena*, *35*(3), 335–356. https://doi.org/10.1016/0167-2789(89)90074-2

Chen, Q., Zhao, H., Li, W., Huang, P., & Ou, W. (2019). Behavior sequence transformer for e-commerce recommendation in Alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data* (pp. 1–4).

Chen, Y., Fan, R., Yang, X., Wang, J., & Latif, A. (2018). Extraction of urban water bodies from high-resolution remote-sensing imagery using deep learning. *Water*, *10*(5), 585. https://doi.org/10.3390/w10050585

Clark, M. P., Slater, A. G., Rupp, D. E., Woods, R. A., Vrugt, J. A., Gupta, H. V., et al. (2008). Framework for understanding structural errors (FUSE): A modular framework to diagnose differences between hydrological models. *Water Resources Research*, *44*(12). https://doi.org/10.1029/2007wr006735

Clevert, D. A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (ELUS). *arXiv preprint arXiv:1511.07289*, *4*(5), 11.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv Prepr. arXiv1901.02860*.

Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017). Language modeling with gated convolutional networks. In *International Conference on Machine Learning* (pp. 933–941).

Daw, A., Thomas, R. Q., Carey, C. C., Read, J. S., Appling, A. P., & Karpatne, A. (2020). Physics-guided architecture (PGA) of neural networks for quantifying uncertainty in lake temperature modeling. In *Proceedings of the 2020 Siam International Conference on Data Mining* (pp. 532–540).

Der Kiureghian, A., & Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter? *Structural Safety*, *31*(2), 105–112. https://doi.org/10.1016/j.strusafe.2008.06.020

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv Prepr. arXiv2010.11929*.

Fang, K., Kifer, D., Lawson, K., & Shen, C. (2020). Evaluating the potential and challenges of an uncertainty quantification method for long short-term memory models for soil moisture predictions. *Water Resources Research*, *56*(12), e2020WR028095. https://doi.org/10.1029/2020wr028095

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

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning* (pp. 1050–1059).

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, *521*(7553), 452–459. https://doi.org/10.1038/nature14541

Gonzalez, J. A., Hurtado, L.-F., & Pla, F. (2021). TWilBert: Pre-trained deep bidirectional transformers for Spanish Twitter. *Neurocomputing*, *426*, 58–69. https://doi.org/10.1016/j.neucom.2020.09.078

Gupta, H. V., Kling, H., Yilmaz, K. K., & Martinez, G. F. (2009). Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *Journal of Hydrology*, *377*(1–2), 80–91. https://doi.org/10.1016/j.jhydrol.2009.08.003

Haykin, S., & Principe, J. (1998). Making sense of a complex world [chaotic events modeling]. *IEEE Signal Processing Magazine*, *15*(3), 66–81. https://doi.org/10.1109/79.671132

Henn, B., Clark, M. P., Kavetski, D., & Lundquist, J. D. (2015). Estimating mountain basin-mean precipitation from streamflow using Bayesian inference. *Water Resources Research*, *51*(10), 8012–8033. https://doi.org/10.1002/2014wr016736

Hoedt, P.-J., Kratzert, F., Klotz, D., Halmich, C., Holzleitner, M., Nearing, G. S., et al. (2021). MC-LSTM: Mass-conserving LSTM. In *International Conference on Machine Learning* (pp. 4275–4286).

Jia, X., Willard, J., Karpatne, A., Read, J., Zwart, J., Steinbach, M., & Kumar, V. (2019). Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM International Conference on Data Mining* (pp. 558–566).

Kan, K., Aubet, F. X., Januschowski, T., Park, Y., Benidis, K., Ruthotto, L., & Gasthaus, J. (2022). Multivariate quantile function forecaster. In *International Conference on Artificial Intelligence and Statistics* (pp. 10603–10621). PMLR.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, *3*(6), 422–440. https://doi.org/10.1038/s42254-021-00314-5

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3128–3137).

Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). Visualizing and understanding recurrent networks. *arXiv Prepr. arXiv1506.02078*.

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, *30*.

Koya, S. R., & Roy, T. (2024). Temporal fusion transformers for streamflow prediction: Value of combining attention with recurrence. *Journal of Hydrology*, *637*, 131301. https://doi.org/10.1016/j.jhydrol.2024.131301

Kratzert, F., Herrnegger, M., Klotz, D., Hochreiter, S., & Klambauer, G. (2019). NeuralHydrology--interpreting LSTMs in hydrology. In *Explainable AI: Interpreting, explaining and visualizing deep learning* (pp. 347–362). Springer.

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

Kratzert, F., Klotz, D., Hochreiter, S., & Nearing, G. S. (2021). A note on leveraging synergy in multiple meteorological data sets with deep learning for rainfall--runoff modeling. *Hydrology and Earth System Sciences*, *25*(5), 2685–2703. https://doi.org/10.5194/hess-25-2685-2021

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

Kumar, R., Samaniego, L., & Attinger, S. (2013). Implications of distributed hydrologic model parameterization on water fluxes at multiple scales and locations. *Water Resources Research*, *49*(1), 360–379. https://doi.org/10.1029/2012wr012195

Laio, F., & Tamea, S. (2007). Verification tools for probabilistic forecasts of continuous hydrological variables. *Hydrology and Earth System Sciences*, *11*(4), 1267–1277. https://doi.org/10.5194/hess-11-1267-2007

Li, D., Marshall, L., Liang, Z., Sharma, A., & Zhou, Y. (2021). Bayesian LSTM with stochastic variational inference for estimating model uncertainty in process-based hydrological models. *Water Resources Research*, *57*(9), e2021WR029772. https://doi.org/10.1029/2021wr029772

Li, X., Xiong, H., Li, X., Wu, X., Zhang, X., Liu, J., et al. (2022). Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. *Knowledge and Information Systems*, *64*(12), 3197–3234. https://doi.org/10.1007/s10115-022-01756-8

Liang, X. (1994). *A two-layer variable infiltration capacity land surface representation for general circulation models*. University of Washington.

Lim, B., Ar\ik, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, *37*, 1748–1764.

Maurer, E. P., Wood, A. W., Adam, J. C., Lettenmaier, D. P., & Nijssen, B. (2002). A long-term hydrologically based dataset of land surface fluxes and states for the conterminous United States. *Journal of Climate*, *15*(22), 3237–3251. https://doi.org/10.1175/1520-0442(2002)015<3237:althbd>2.0.co;2

Mizukami, N., Clark, M. P., Newman, A. J., Wood, A. W., Gutmann, E. D., Nijssen, B., et al. (2017). Towards seamless large-domain parameter estimation for hydrologic models. *Water Resources Research*, *53*(9), 8020–8040. https://doi.org/10.1002/2017wr020401

Mizukami, N., Rakovec, O., Newman, A. J., Clark, M. P., Wood, A. W., Gupta, H. V., & Kumar, R. (2019). On the choice of calibration metrics for "high-flow" estimation using hydrologic models. *Hydrology and Earth System Sciences*, *23*(6), 2601–2614. https://doi.org/10.5194/hess-23-2601-2019

Moradkhani, H., Hsu, K. L., Gupta, H., & Sorooshian, S. (2005). Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter. *Water Resources Research*, *41*(5). https://doi.org/10.1029/2004wr003604

Naeini, M. P., Cooper, G., & Hauskrecht, M. (2015). Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Nash, J. E., & Sutcliffe, J. V. (1970). River flow forecasting through conceptual models: Part 1. A discussion of principles. *Journal of Hydrology*, *10*, 282–290. https://doi.org/10.1016/0022-1694(70)90255-6

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

Newman, A. J., Mizukami, N., Clark, M. P., Wood, A. W., Nijssen, B., & Nearing, G. (2017). Benchmarking of a physically based hydrologic model. *Journal of Hydrometeorology*, *18*(8), 2215–2225. https://doi.org/10.1175/jhm-d-16-0284.1

Olah, C., Mordvintsev, A., & Schubert, L. (2017). Feature visualization. *Distill*, *2*(11), e7. https://doi.org/10.23915/distill.00007

Piazzi, G., Thirel, G., Perrin, C., & Delaigue, O. (2021). Sequential data assimilation for streamflow forecasting: Assessing the sensitivity to uncertainties and updated variables of a conceptual hydrological model at basin scale. *Water Resources Research*, *57*(4). https://doi.org/10.1029/2020wr028390

Pölz, A., Blaschke, A. P., Komma, J., Farnleitner, A. H., & Derx, J. (2024). Transformer versus LSTM: A comparison of deep learning models for karst spring discharge forecasting. *Water Resources Research*, *60*(4), e2022WR032602. https://doi.org/10.1029/2022wr032602

Raftery, A. E., Gneiting, T., Balabdaoui, F., & Polakowski, M. (2005). Using Bayesian model averaging to calibrate forecast ensembles. *Monthly Weather Review*, *133*(5), 1155–1174. https://doi.org/10.1175/MWR2906.1

Rakovec, O., Mizukami, N., Kumar, R., Newman, A. J., Thober, S., Wood, A. W., et al. (2019). Diagnostic evaluation of large-domain hydrologic models calibrated across the contiguous United States. *Journal of Geophysical Research: Atmospheres*, *124*(24), 13991–14007. https://doi.org/10.1029/2019jd030767

Rasmy, L., Xiang, Y., Xie, Z., Tao, C., & Zhi, D. (2021). Med-BERT: Pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *npj Digital Medicine*, *4*(1), 86. https://doi.org/10.1038/s41746-021-00455-y

Räuker, T., Ho, A., Casper, S., & Hadfield-Menell, D. (2023). Toward transparent AI: A survey on interpreting the inner structures of deep neural networks. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SATML)* (pp. 464–483). IEEE.

Sadeghi Tabas, S. (2023). Explainable physics-informed deep learning for rainfall-runoff modeling and uncertainty assessment across the continental United States. Retrieved from https://open.clemson.edu/all_dissertations/3269/

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, *36*(3), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

Samadi, S., Pourreza-Bilondi, M., Wilson, C. A. M. E., & Hitchcock, D. B. (2020). Bayesian model averaging with fixed and flexible priors: Theory, concepts, and calibration experiments for rainfall-runoff modeling. *Journal of Advances in Modeling Earth Systems*, *12*(7), e2019MS001924. https://doi.org/10.1029/2019ms001924

Samaniego, L., Kumar, R., & Attinger, S. (2010). Multiscale parameter regionalization of a grid-based hydrologic model at the mesoscale. *Water Resources Research*, *46*(5). https://doi.org/10.1029/2008wr007327

Seibert, J., & Vis, M. J. P. (2012). Teaching hydrological modeling with a user-friendly catchment-runoff-model software package. *Hydrology and Earth System Sciences*, *16*(9), 3315–3325. https://doi.org/10.5194/hess-16-3315-2012

Seibert, J., Vis, M. J. P., Lewis, E., & van Meerveld, H. J.(2018). Upper and lower benchmarks in hydrological modelling. *Hydrological Processes*, *32*(8), 1120–1125. https://doi.org/10.1002/hyp.11476

Shen, C. (2018). A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resources Research*, *54*(11), 8558–8593. https://doi.org/10.1029/2018wr022643

Swamy, V., Montariol, S., Blackwell, J., Frej, J., Jaggi, M., & Käser, T. (2024). InterpretCC: Intrinsic user-centric interpretability through global mixture of experts. *arXiv preprint arXiv:2402.02933*.

Tabas, S. S., & Samadi, S. (2022). Variational Bayesian dropout with a Gaussian prior for recurrent neural networks application in rainfall--runoff modeling. *Environmental Research Letters*, *17*, 65012.

Tagasovska, N., & Lopez-Paz, D. (2019). Single-model uncertainties for deep learning. *Advances in Neural Information Processing Systems*, *32*.

Thornton, P. E., Shrestha, R., Thornton, M., Kao, S. C., Wei, Y., & Wilson, B. E. (2021). Gridded daily weather data for North America with comprehensive uncertainty quantification. *Scientific Data*, *8*(1), 190. https://doi.org/10.1038/s41597-021-00973-0

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., & Sun, L. (2022). Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*.

Xia, Y., Mitchell, K., Ek, M., Sheffield, J., Cosgrove, B., Wood, E., et al. (2012). Continental-scale water and energy flux analysis and validation for the North American Land Data Assimilation System project phase 2 (NLDAS-2): 1. Intercomparison and application of model products. *Journal of Geophysical Research*, *117*(D3). https://doi.org/10.1029/2011jd016048

Yilmaz, K. K., Gupta, H. V., & Wagener, T. (2008). A process-based diagnostic approach to model evaluation: Application to the NWS distributed hydrologic model. *Water Resources Research*, *44*(9), 1–18. https://doi.org/10.1029/2007WR006716

Yin, H., Guo, Z., Zhang, X., Chen, J., & Zhang, Y. (2022). RR-Former: Rainfall-runoff modeling based on Transformer. *Journal of Hydrology*, *609*, 127781. https://doi.org/10.1016/j.jhydrol.2022.127781

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325