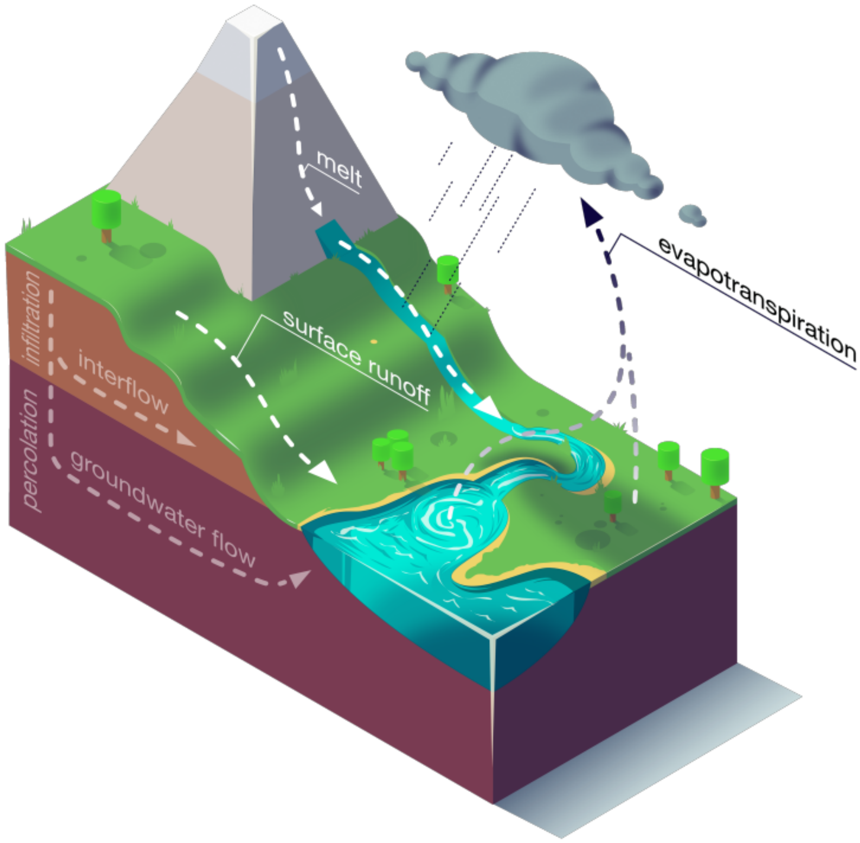
Transformer for Rainfall-Runoff Modeling

Assessing the applicability of Transformer-based architectures as rainfall-runoff models

MSc Graduation Report

Kangmin Mao



Transformer for Rainfall-Runoff Modeling

Assessing the applicability of Transformer-based architectures as rainfall-runoff models

by

Kangmin Mao

Committee TU Delft:	Dr. Riccardo Taormina
	Dr. Markus Hrachowitz
	Dr. Jacopo De Stefani
Supervisors Deltares:	Anaïs Couasnon
	Ruben Dahmh
	Dr. Jonathan Nuttall
Project Duration:	July, 2022 - January, 2023
Faculty:	Faculty of Civil Engineering and Geosciences, Delft

TU Delft Deltares

Preface

After more than two years of studying at Delft, I am about to complete my master's thesis and obtain my master's degree. During this period, of being away from my family and facing the negative effects of COVID, it feels great to finally be graduating. I would like to express my sincere gratitude to those who have provided me with great assistance during the past months of my thesis journey.

Firstly, I would like to thank Riccardo for guiding me throughout my internship and thesis. He gave me the opportunity to explore machine learning and apply it to my research. I also appreciate his prompt and active responses to my emails during the research, which helped me to progress quickly in my research.

Secondly, I would like to especially thank Anaïs for her involvement in almost every aspect of my research. Weekly meetings with her and getting timely feedback, and suggestions helped me to overcome difficulties and reduce my anxiety. I am very fortunate to have had her help throughout my thesis. Additionally, I would like to thank Ruben for accepting me as an intern at Deltares to do my thesis. His critical and essential questions have greatly influenced me and taught me how to conduct research. I also thank Xiaohan for supervising me in the early stages of my thesis.

Finally, I would like to thank Markus for his advice on hydrology and for emphasizing to me that we cannot fully explain the data-driven hydrological models only ”now”, which also encouraged me. Also, thanks to Jacopo for correcting the limitations of my project and giving suggestions from a machine learning perspective.

I would also like to extend my gratitude to my family and friends for their encouragement and support during this journey. This thesis would not have been possible without the help of all of you. Thank you.

Kangmin Mao
Delft, January 2023

Summary

Modeling the relationship between rainfall and runoff is a longstanding challenge in hydrology and is crucial for informed water management decisions. Recently, Deep Learning (DL) models, particularly Long Short-Term Memory (LSTM), have shown promising results in simulating this relationship. The Transformer, a newly proposed deep learning architecture, has also demonstrated the ability to outperform LSTM in machine translation, text classification, etc. However, there has been limited research on applying Transformers for rainfall-runoff modeling.

The research examined the performance of using Transformer architecture, including its time series forecasting variants, to develop rainfall-runoff models using the CAMELS (US) data set. These models were compared to the LSTM regional rainfall-runoff models, with a particular focus on snow-driven basins as the attention mechanism in Transformer is believed to allow it to attend to the earlier precipitation events in the meteorological forcing. Additionally, the Transformer's potential as a global rainfall-runoff model was also tested using the global Caravan data to determine if it could learn and generalize a wide range of rainfall-runoff behaviors, allowing it to potentially be applied in ungauged basins.

The results suggest that while Transformer and its variants may not be able to fully replace LSTM for rainfall-runoff modeling, the variant called Reformer has shown promise for daily discharge forecasting in snow-driven basins, particularly in terms of peak flow and low flow prediction. However, using the global Caravan data for building a global rainfall-runoff model was not successful due to uncertainty in the forcing data, particularly precipitation. The code for Transformer-based rainfall-runoff modeling is available publicly at https://github.com/Numpy-Panda/neuralhydrology_Transformer.

Contents

List of Figures

List of Tables

List of Abbreviations

ANNs	Artificial Neural Networks	1
CAMELS	Catchment Attributes and MEteorology for Large-sample Studies	2
CDF	Cumulative Density Functions	25
DL	Deep Learning	ii
ENSO	El Niño-Southern Oscillation	3
FDC	Flow Duration Curve	18
FHV	FDC High-segment Volume	18
FLV	FDC Low-segment Volume	18
FMS	FDC Midsegment Slope	18
GPU	Graphics Processing Unit	2
GRU	Gate Recurrent Unit	3
KGE	Kling-Gupta Efficiency	18
LSTM	Long Short-Term Memory	ii
LSH	large-sample hydrology	2
MLP	Multilayer Perceptron	9
NSE	Nash–Sutcliffe Efficiency Coefficient	2
PUB	Prediction in Ungauged Basins	1
RNNs	Recurrent Neural Networks	9
TRMM	Tropical Rainfall Measuring Mission	43
UPH	Unsolved Problems in Hydrology	1

1

Introduction

1.1. Background

Stream flow forecasting is crucial for hydrological research and water resource management, especially in the context of flooding control and drought prevention (Brath et al., 2002). Worldwide, various natural hazards threaten human life, among which floods are arguably one of the most devastating natural hazards, contributing to one-third of the economic losses caused by natural hazards, in recent decades, floods have caused thousands of deaths with an increasing trend, disrupted economic activities, and destroyed infrastructure (Pinos & Quesada-Román, 2021). Drought was one of the hazards that led to the largest human losses from 1970 to 2019, with a total of approximately 650,000 deaths (Edith et al., 2021).

Rainfall-runoff relations denote how the basin discharge responds to the mass (i.e. water from precipitation) and energy (e.g. radiation) inputs. Rainfall-runoff modeling is important for water management decision-making. For example, accurate and advanced streamflow forecasts from rainfall-runoff simulation can help mitigate the impacts of flood and drought hazards by providing early warning, allowing people to evacuate before floods or prepare for drought conditions. Rainfall-runoff models use meteorological data, such as precipitation and evaporation, as inputs to predict the discharge of a basin. The existing modeling approaches, depending on the extent to which physical process knowledge is imposed in the simulation, range from fully data-driven, over conceptual, to physically based approaches. Physics-based basin-scale models, which are based on a detailed understanding of physical processes, generally require a large amount of computational resources and data and are rarely used for operational stream flow forecasting, except in small experimental basins. This limits their use in larger basins (Kratzert et al., 2018). However, conceptual models, which are usually simpler and require less data, are more commonly used for operational forecasting.

Prediction in Ungauged Basins (PUB), where discharge observations are lacking, is one of the twenty-three Unsolved Problems in Hydrology (UPH) (Blöschl et al., 2019). When transferring a conceptual model from gauged to ungauged basins, there may be sources of uncertainty due to errors in computing local and regional model parameters, as well as the relationship between local parameters and catchment attributes, and the model structure (Wagener & Wheater, 2006). But the hydrologic model-independent data-driven methods can avoid the impact t of hydrologic model structure and parameter uncertainty (Razavi & Coulibaly, 2013) and so be the potential solution to PUB. Over the past decades, numerous data-driven techniques, largely attributed to machine learning, have been developed and applied in rainfall-runoff modeling (Shrestha & Solomatine, 2008). These techniques include Artificial Neural Networks (ANNs) (Daniell, 1991), fuzzy regression (Bardossy et al., 1990), genetic programming (Babovic & Keijzer, 2000), model trees (Solomatine & Dulal, 2003), and support vector machines (Bray & Han, 2004).

Recently, due to the availability of huge data sets and powerful Graphics Processing Unit (GPU), the neural networks-based DL technique has achieved great performance in a variety of fields (Schmidhuber, 2015) such as computer vision (Farabet et al., 2012), natural language processing (Sutskever et al., 2014), and speech recognition (Hinton et al., 2012). These advances have also drawn the attention of the hydrological community, inspiring new efforts to apply the DL techniques in rainfall-runoff modeling. One notable work (Kratzert et al., 2018) that boosted machine learning development in hydrology is the use of Long Short-Term Memory (LSTM) networks (Hochreiter & Schmidhuber, 1997) for 241 catchments rainfall-runoff modeling in the US when the large large-sample hydrology (LSH) dataset, Catchment Attributes and MEteorology for Large-sample Studies (CAMELS) (US), was available. This LSTM model outperformed many conceptual models in simulating the rainfall-runoff relations in the US and demonstrated the feasibility of using LSTM for this task, which then led to a proliferation of research on LSTM-based rainfall-runoff modeling. For example, a single LSTM model (Kratzert, Klotz, Shalev, et al., 2019) was used for daily rainfall-runoff modeling in 531 basins in the US, considering catchment attributes to help the model learn differences in hydrologic behavior between basins. This LSTM model achieved a median Nash–Sutcliffe Efficiency Coefficient (NSE) (Nash & Sutcliffe, 1970) of 0.73 on the 531 basins and performed better than the previous LSTM model applied to 241 basins, which did not utilize basin attribute data. Additionally, a multi-time scale LSTM model (Gauch et al., 2021) was proposed, using hourly and daily forcing data as input to predict daily discharge in the same 531 basins, resulting in better performance (a median NSE of 0.77) than the previous 531 basins single LSTM model. Concurrently, LSTM-based rainfall-runoff models have also achieved remarkable success when used for out-of-sample prediction problems such as PUB (Kratzert, Klotz, Herrnegger, et al., 2019) and extreme low-probability streamflow events (Frame et al., 2022) in US basins. LSTM has also been used for rainfall-runoff modeling in areas outside of the US, such as Great Britain (Lees et al., 2021) and Chile (Ma et al., 2021), with good performance.

In rainfall-runoff modeling, the discharge at a specific time step is determined by the meteorological input from the previous time period. Building a DL-based rainfall-runoff model requires selecting a suitable DL architecture and training it on a reliable dataset. In terms of architecture, while LSTM models are currently the most accurate for rainfall-runoff prediction when DL methods are applied to this problem (Frame et al., 2022), there are still some debates on the limitation of LSTM, which may pose challenges in rainfall-runoff modeling. One potential challenge is that LSTM processes the input sequence one step at a time, which can result in a long path between the input and output sequences in the model, making it difficult to learn dependencies (Hochreiter et al., 2001). One experiment (Khandelwal et al., 2018) shows that LSTM language models have an effective context size of about 200 tokens on average. Another experiment (Kratzert, Klotz, Shalev, et al., 2019) explicitly demonstrated that an LSTM-based rainfall-runoff model achieved better performance when a past 270-day meteorological sequence was used as input, rather than longer sequences like 365 or 720 days. Based on these findings, it is unclear whether LSTM can fully utilize and learn the information from very previous meteorological events in a long meteorological input sequence in rainfall-runoff modeling. This may present a challenge for modeling in basins with significant storage effects, such as snow-driven basins, where water requires a long time to be released into streamflow from snowfall. Besides, some research (Kratzert, Klotz, Shalev, et al., 2019; Ma et al., 2021) also showed that LSTM performs badly in arid-basin discharge forecasting. In terms of data, large sample datasets are necessary for developing generalizable hydrologic data-driven models. As a result, there has been an increasing number of publicly available LSH datasets in specific regions or countries, such as Australia (Fowler et al., 2021), Brazil (Chagas et al., 2020), and North America (Arsenault et al., 2020). Many of these datasets are referred to as Catchment Attributes and MEteorology for Large-sample Studies (CAMELS) datasets. Furthermore, the availability of region-specific data may limit hydrological research due to the lack of common standards for intercomparison, etc. (Addor et al., 2020). As a result, a global LSH data set called Caravan (a series of CAMELS) (Kratzert et al., 2022) has been published. Uncertainty can be introduced into this LSH data due to inappropriate measurement, interpolation, and other factors, leading to bias and error in analyses and conclusions. Therefore, the estimation of uncertainty is a crucial aspect of hydrological research (McMillan et al., 2018).

The Transformer (Vaswani et al., 2017), also known as the Vanilla Transformer, is a DL architecture that was proposed in 2017 for machine translation. Nowadays, in the field of DL, the Transformer is considered a foundational model that can be trained on broad data and be adapted to a wide range of downstream tasks (Bommasani et al., 2021). In the past years, Transformer architecture has been used in various tasks with great success thanks to its self-attention mechanism (Vaswani et al., 2017), which can effectively establish long-range dependencies and shorten the calculation path between input and output sequences (Bommasani et al., 2021). However, Transformer-based models have struggled with modeling extremely long-range dependencies due to the large amount of quadratic computation required by the self-attention mechanism (Wang et al., 2020). To address this issue, numerous Transformer variants have been proposed to reduce the computational costs, such as the Informer (H. Zhou et al., 2021), Reformer (Kitaev et al., 2020), and FED former (T. Zhou et al., 2022), which are specifically designed for effective long-term time series forecasting.

In the last few years, there has been little research on the application of Transformer architectures for hydrology, and only in recent months has some research in this area emerged. Most of these studies have shown that Transformer-based hydrological models outperformed LSTM-based models in terms of discharge prediction. For example, RR-former (Yin et al., 2022) utilizes a nearly identical architecture to the vanilla Transformer, which inputs both meteorological forcing and historical (known) runoff to simulate multi-step-ahead daily discharge (i.e. 1-7 days) using CAMELS (US) data. The results were compared with two LSTM-based rainfall-runoff models (LSTM-MSV-S2S (Yin et al., 2021) and LSTM-S2S (Xiang et al., 2020)) and revealed that the RR-Former performed better than the LSTM-based model in both individual (one model for each basin) and regional (one model for all basins) modeling, and RR-Former is well-suited for large datasets. In addition, a study (Amanambu et al., 2022) used Transformer architecture to predict hydrological drought for 30, 60, 90, 120, and 180 days into the future, using daily stage-height data from two gauging stations in the Apalachicola River, Florida. The results showed that, on average, the Transformer-based models performed better than the LSTM models across all timestamps for predicting hydrological drought. A study (Castangia et al., 2023) utilized the Transformer architecture, which inputs daily water level data from 13 upstream hydrological stations to predict flooding in the downstream area of Doboj, Bosnia and Herzegovina. The results showed that the Transformer-based forecasting model was superior to Gate Recurrent Unit (GRU) (Cho et al., 2014) or LSTM-based models. The outstanding performance of the Transformer-based model can be interpreted by its ability to accurately identify upstream stations with strong predictive capabilities and its attention score maps that rapidly change at the start of a flooding event and quickly restore at the end of the event. In addition, a study (Liu et al., 2022) employed a double-encoder Transformer architecture, where two encoders input the streamflow and El Niño-Southern Oscillation (ENSO) (Ropelewski & Halpert, 1986) data, respectively. Then, the ”cross-attention” mechanism was then utilized to capture the relationship between the two time series sequences, enabling the ability to make precise long-term predictions for the flow of the Yangtze River.

1.2. Research Motivation

Currently, most existing studies on Transformer-based hydrological modeling have demonstrated superior performance in forecasting discharge when compared to the LSTM architecture, which has been the leader in this task. However, in these studies, discharge (or water level) has been used as input for the Transformer model, which makes it difficult to transfer these models to ungauged basins.

From the perspective of Transformer architecture, using it for rainfall-runoff simulation seems desirable for several reasons. First, its self-attention mechanism can shorten the path between input and output sequences and effectively focus on earlier rainfall events more efficiently than LSTM, which is useful for simulating rainfall-runoff relations in basins with significant storage-effect, such as snow-driven basins. Second, Transformer architecture has more trainable parameters compared to LSTM, an experiment (Popel & Bojar, 2018) has shown that Transformer generally performs better on larger training datasets, which is consistent with the findings of RR-former study (Yin et al., 2022). This suggests that using a larger dataset, containing a greater number of basins, to train a rainfall-runoff model based on Transformer could be beneficial. On one hand, the Transformer may perform better on a larger dataset, and on the other hand, it perhaps learns more rainfall-runoff behaviors from a diverse range of basins, for example, from the global LSH Caravan data, which could potentially be transferred to ungauged basins, thus addressing the PUB problem.

Given the limitations of existing Transformer-based hydrological models in terms of not being transferable and the potential suitability of Transformer architecture for rainfall-runoff modeling, a study focusing on Transformer-based rainfall-runoff modeling that predicts discharge solely through meteorological forcing would be significant and meaningful.

1.3. Problem Statement

The problem of this thesis is to investigate the potential or applicability of Transformer-based models for rainfall-runoff modeling. It is hypothesized that Transformer architectures may be more effective for rainfall-runoff modeling due to their ability to consider earlier rainfall events more effectively than LSTM models and potentially being able to learn more rainfall-runoff behaviors, allowing for transfer to ungauged basins. Exploring the potential and applicability of Transformer architecture includes the possibility of outperforming LSTM-based models and serving as a global rainfall-runoff model based on the Caravan dataset. This study is significant as it may lead to the development of a better rainfall-runoff model and contribute to the advancement of machine learning in the field of hydrology.

1.4. Research Objective

The purpose of this research is to explore the applicability of the Transformer architecture to accurately describe daily rainfall-runoff patterns in multiple catchments. It is generally computationally expensive for the Transformer to predict streamflow on a finer timescale, such as hourly, due to the long input sequence required (Gauch et al., 2021). As a result, the focus of this study will be on daily prediction. The study also aims to compare the performance of both the Transformer and LSTM architectures in rainfall-runoff modeling. To determine the generalizability and effectiveness of the Transformer-based approach, experiments will be conducted on a large number of catchments from the CAMELS (US) data set, which some LSTM-based rainfall-runoff models rely on. The Transformer-based rainfall-runoff model will also be trained on the global LSH data set, Caravan, to assess its ability to operate as a global model and evaluate the uncertainty in the global LSH data.

1.5. Research Question

The main research question is: **How do Transformer-based architectures perform as rainfall-runoff models?** To answer this main research question, the following sub-research questions have been formulated:

- **Sub-Research Question 1**: Can the Transformer architectures outperform LSTM in rainfall-runoff modeling?
- **Sub-Research Question 2**: Can the Transformer be a global rainfall-runoff model based on the Caravan data set?

1.6. Reading Guide

The report is organized as follows: Chapter 2 provides background information on hydrological modeling based on DL techniques, including a description of the model calibration procedure and the feasible DL architecture for rainfall-runoff modeling. Chapter 3 presents a detailed overview of the data sets used in the research, and evaluation methods, and shows how the experiments were designed to address the research questions. Chapter 4 presents the experimental results and provides some discussion on the limitations of the experiments. Chapter 5 gives conclusions and recommendations and discusses future work.

2

Theoretical background

This chapter introduces the background information on hydrological modeling based on DL techniques. Section 2.1 explains how a DL-based hydrological model is calibrated, including the process of generating training, validation, and testing data, and how the model is trained and tested. Section 2.2 presents some DL architectures that are suitable for hydrological modeling, with a focus on the Transformer architecture.

2.1. Calibration procedure

Like a conceptual hydrological model, a DL-based rainfall-runoff model also needs to be calibrated, which is commonly referred to as training. Essentially, the rainfall-runoff model translates meteorological data into discharge. For a specific catchment, the discharge can be seen as a function of past meteorological data, such as precipitation, temperature, vapor pressure, and shortwave radiation, given that the catchment characteristics do not vary over time.

To prevent overfitting, which is a common issue in DL that can lead to poor model generalization to unseen data (Ying, 2019), it is necessary to split the observed data into training, validation, and test sets. In this research, to ensure model convergence, the model was trained for 100 epochs based on training data, which means the model was calibrated using the entire training data set 100 times and the model parameters were saved after each epoch. This resulted in a total of 100 sets of parameters (or weights) for the model. The validation data was then used to select the parameter set with the best performance (seen Fig. 2.1). The model with the selected parameter set was finally tested on the test data set using various metrics. This process can be seen in Fig. 2.2 and helps to ensure that the model's prediction capabilities are evaluated using data it has never seen before.

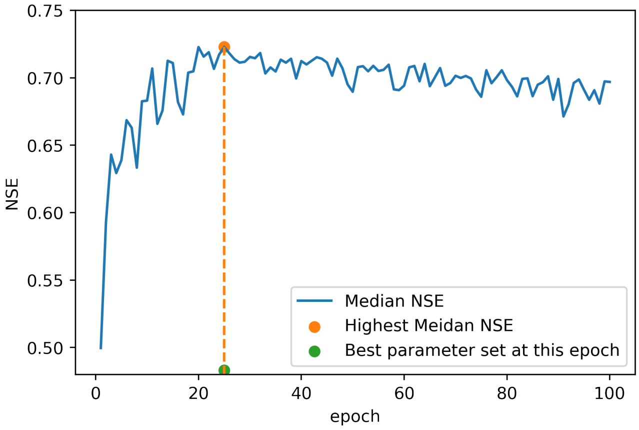


Figure 2.1: An illustration of a training curve, the model was trained for 100 epochs to guarantee convergence and was evaluated in each epoch using validation data by NSE loss. The parameter set chosen in the epoch where the the model achieved best median NSE will be used for further testing.

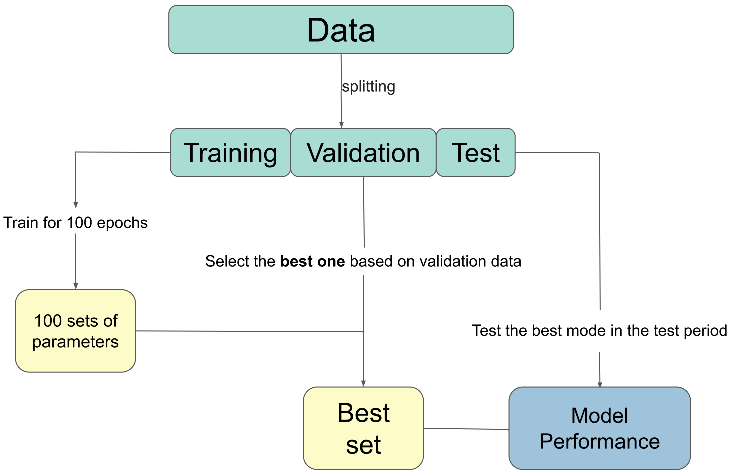


Figure 2.2: The data splitting: training, test, and evaluation sets.

A large number of training samples are typically needed to train a DL-based rainfall-runoff model. Each sample usually consists of observed meteorological forcing and discharge data. These samples are obtained from multi-year historical observation data by using a length-n moving window that selects the meteorological forcing data from the first day of the training period. Every time the window is moved by one stride, a length-n (days) forcing data and the discharge data from the n-th time step (or day) are obtained as one training sample. This process is repeated until it is no longer possible to select a complete length-n forcing data, and it is important to ensure that every sample has the same length, n. In addition, the DL model must be able to recognize differences in hydrologic behaviors between different catchments because basins may respond differently to similar forcing input. To allow the model to make these distinctions, extra information in the form of static catchment attributes such as catchment slope and mean elevation, which do not change over time in our study, must be provided. These static data are usually concatenated into each step of the samples, and samples from different basins are concatenated with their unique basin attributes at each step. Then, m random samples are shuffled and combined into a batch, as shown in Fig. 2.3 and Fig. 2.4. Shuffling the m samples can help the model converge faster (Kratzert et al., 2018). The validation and test data come from the validation and test periods.

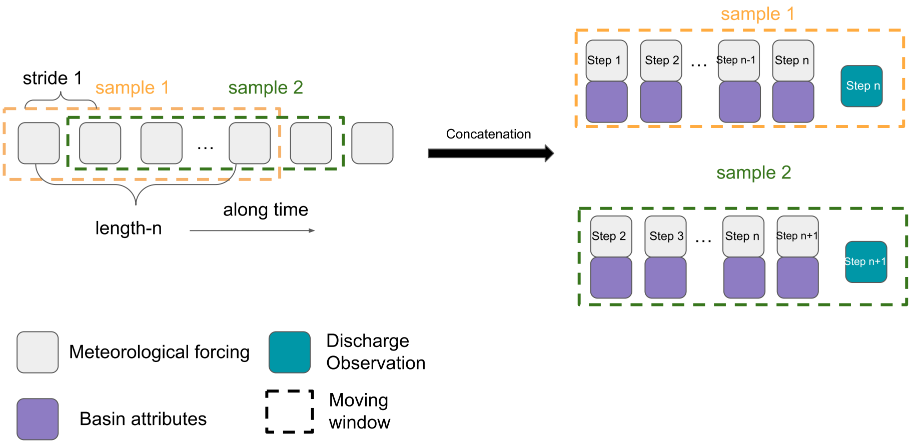


Figure 2.3: Examples of training samples taken from a historical record's training period, typically spanning several years.

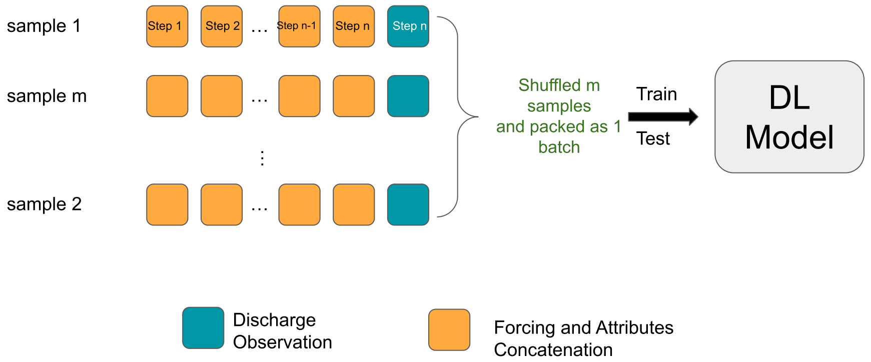


Figure 2.4: Batch of *m* shuffled samples being fed to deep learning model.

DL-based rainfall-runoff models take in m random samples at the same time and predict m discharges, which are compared to the observed discharges based on a loss function, which measures the error between the prediction and the observation and guides the model to update its trainable parameters. It is important to ensure that the value of m is not too small in order to avoid fluctuations in the loss during training. In this way, the DL model can be trained to map meteorological forcing data to discharge (Kratzert et al., 2018).

It's important to note that the discharge observations can only be used for loss calculation and not as input. There are time series forecasting neural networks that require both meteorological forcing data and historical discharge as inputs to produce accurate predictions. However, a key reason for using a DL model to simulate discharge in this research is to solve the PUB problem, and when transferring the DL-based rainfall-runoff models to ungauged basins, discharge observations will not be available.

2.2. Models

This section will introduce some suitable DL-based models for rainfall-runoff modeling.

2.2.1. Artificial Neural Network

ANNs are a fundamental and essential component of many DL architectures, also known as dense layers or Multilayer Perceptron (MLP) (Hastie et al., 2009). As illustrated in Fig. 2.5, an ANNs architecture typically consists of an input layer, an output layer, and one or more hidden layers (two hidden layers, in this case). The circles represent neurons, which gather information from previous layers and transmit it to the next layers, as indicated by the black arrows. In this way, the input signal can pass the network layer by layer and be transformed into the final output.

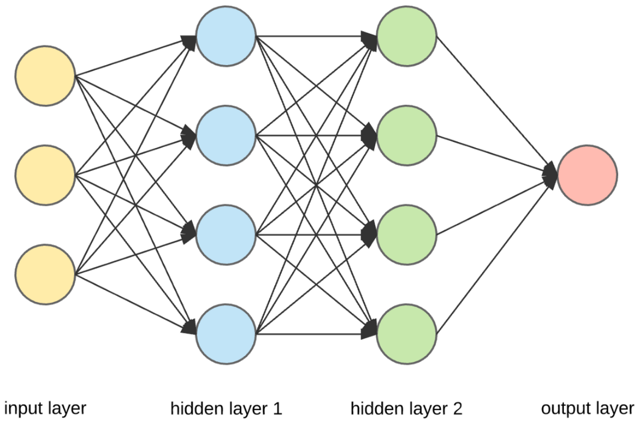


Figure 2.5: Diagram of an artificial neural network architecture (Dertat, 2017).

Each neuron receives information from all neurons in the previous layer. The received information is a weighted sum of the information transmitted by the previous layer neurons, plus a bias term, and is then activated using a nonlinear function. Simple ANNs can be used in rainfall-runoff modeling (Daniell, 1991; Halff et al., 1993), but perform poorly because they are not able to effectively utilize sequential order information.

2.2.2. Recurrent Neural Network

Recurrent Neural Networks (RNNs) (Rumelhart et al., 1986) are designed to handle sequential data, and the most commonly used type of RNNs is LSTM (Hochreiter & Schmidhuber, 1997). Neuralhydrology (Kratzert, Herrnegger, et al., 2019) is a Python library for training neural networks (primarily based on LSTMs) with a strong emphasis on hydrological applications. It has been widely used in research in recent years (Frame et al., 2022; Frame et al., 2021; Gauch et al., 2021), and has demonstrated that LSTMs are capable of accurately simulating rainfall-runoff behaviors. The architecture of an LSTM can be seen in Fig. 2.6.

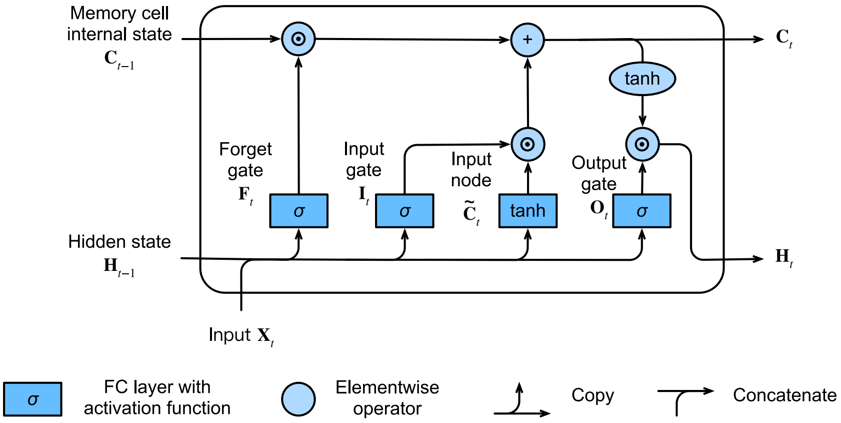


Figure 2.6: Diagram of LSTM unit architecture (Calzone, 2022).

As depicted in Fig. 2.6, an LSTM unit consists of four neural networks that serve the function of selecting information, namely the forget gate $\mathbf{F}_t$, the input gate $\mathbf{I}_t$, the output gate $\mathbf{O}_t$, and the input node $\tilde{\mathbf{C}}_t$. The forget gate $\mathbf{F}_t$ determines which information should be discarded from the cell's internal state at the previous step $\mathbf{C}_{t-1}$ based on the hidden state at the previous step $\mathbf{H}_{t-1}$ and the current input $\mathbf{X}_t$. Then, the input node $\tilde{\mathbf{C}}_t$ generates candidate information $\tilde{\mathbf{C}}_t$, which is then selected by the input gate $\mathbf{I}_t$ and added to the internal cell state $\mathbf{C}_t$. This process updates the cell state $\mathbf{C}_t$ and can be described using the Eq. 2.1 (Calzone, 2022).

$$\begin{aligned}
\mathbf{F}_t &= \sigma\left(\mathbf{X}_t \mathbf{W}_{xf} + \mathbf{H}_{t-1}\mathbf{W}_{hf} + \mathbf{b}_f\right) \\
\tilde{\mathbf{C}}_t &= \tanh\left(\mathbf{X}_t \mathbf{W}_{xc} + \mathbf{H}_{t-1}\mathbf{W}_{hc} + \mathbf{b}_c\right) \\
\mathbf{I}_t &= \sigma\left(\mathbf{X}_t \mathbf{W}_{xi} + \mathbf{H}_{t-1}\mathbf{W}_{hi} + \mathbf{b}_i\right) \\
\mathbf{C}_t &= \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tilde{\mathbf{C}}_t
\end{aligned} \tag{2.1}$$

where $\mathbf{W}$ is weight parameters and $\mathbf{b}$ is bias parameters, tanh and σ are the activation function, and $\odot$ denotes product operator.

The output gate $\mathbf{O}_t$ filters the information from the updated internal cell state of $\mathbf{C}_t$ into LSTM cell output $\mathbf{H}_t$ based on the previous hidden state $\mathbf{H}_{t-1}$ and current input $\mathbf{X}_t$. This process can be described using Eq. 2.2 (Calzone, 2022).

$$\begin{aligned}
\mathbf{O}_t &= \sigma\left(\mathbf{X}_t \mathbf{W}_{xo} + \mathbf{H}_{t-1}\mathbf{W}_{ho} + \mathbf{b}_o\right) \\
\mathbf{H}_t &= \mathbf{O}_t \odot \tanh\left(\mathbf{C}_t\right)
\end{aligned} \tag{2.2}$$

When applying LSTM to rainfall-runoff modeling, the LSTM unit digests a sample step by step, as depicted in Fig. 2.7. It is important to note that the gray square represents the same LSTM unit at a different time step. It can also be observed that the output of the LSTM unit at time step i serves as its input at step $i+1$, and the LSTM unit can only input data for step $i+1$ once the processing for step i is completed. This is known as recurrent architecture. By processing the data step by step, the LSTM is able to fully utilize the sequential order information within the sequence. The black square represents the dense layer or ANNs, which transforms the output of the LSTM into the discharge at time step n. The LSTM converts meteorological data of length n (concatenated with the attributes of the basin) into a runoff sequence of length 1 (in the case of daily prediction in this research, the prediction length is only 1).

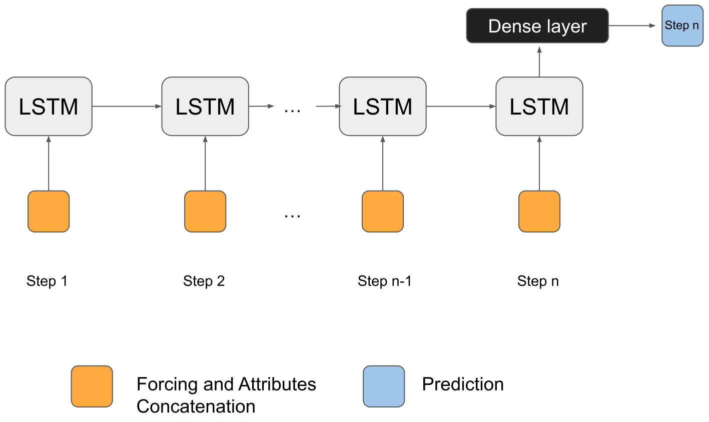


Figure 2.7: Illustration of LSTM processing input step-by-step for rainfall-runoff modeling.

2.2.3. Transformers

Transformer (Vaswani et al., 2017) was a novel DL architecture that is particularly effective for processing sequential data such as time series, the architecture is shown in Fig. 2.8.

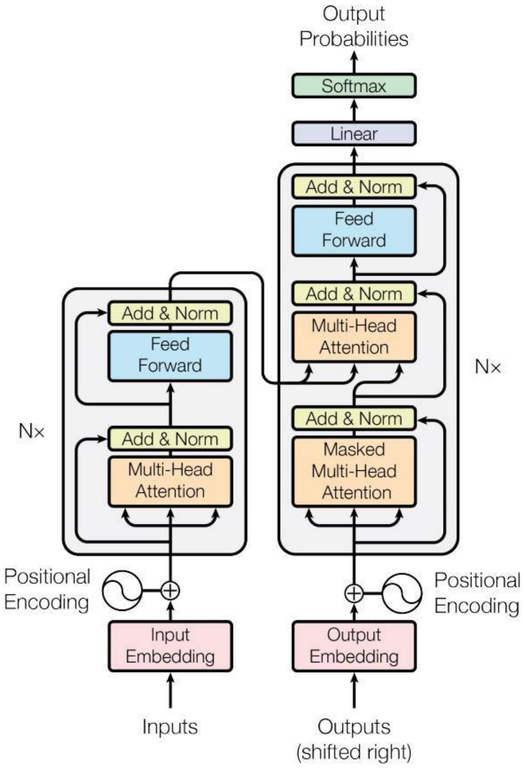


Figure 2.8: Transformer architecture (Vaswani et al., 2017).

As depicted in Fig. 2.8, the Transformer utilizes an encoder-decoder structure, denoted by ”N×”, meaning N encoders or N decoders, which consists of feedforward neural networks and multiple self-attention layers. The encoder processes the input sequence and extracts features, while the decoder uses these features to generate the output sequence (Vaswani et al., 2017).

The LSTM models dependencies based on its recurrent architecture, while the Transformer relies on attention mechanisms (as represented by the ”Multi-Head Attention” and orange square in Fig. 2.8). A major difference between the two is that the recurrent architecture processes the sequence step by step, while the attention mechanism inputs the entire sequence at once, resulting in the loss of sequential order information in the Transformer’s input (Vaswani et al., 2017). In simpler terms, the Transformer architecture is unable to determine the relative positions of the meteorological forcing sequence in rainfall-runoff modeling. To address this issue, the Transformer employs positional encoding (denoted by the circles in Fig. 2.8), which injects information about the relative or absolute position of the steps into the sequence. The attention mechanism and positional encoding are vital components of the Transformer architecture and are essential for time series forecasting tasks like rainfall-runoff modeling.

Self attention mechanism

The attention mechanism employed by the Transformer is known as Scaled Dot-Product Attention and is depicted in Fig. 2.9.

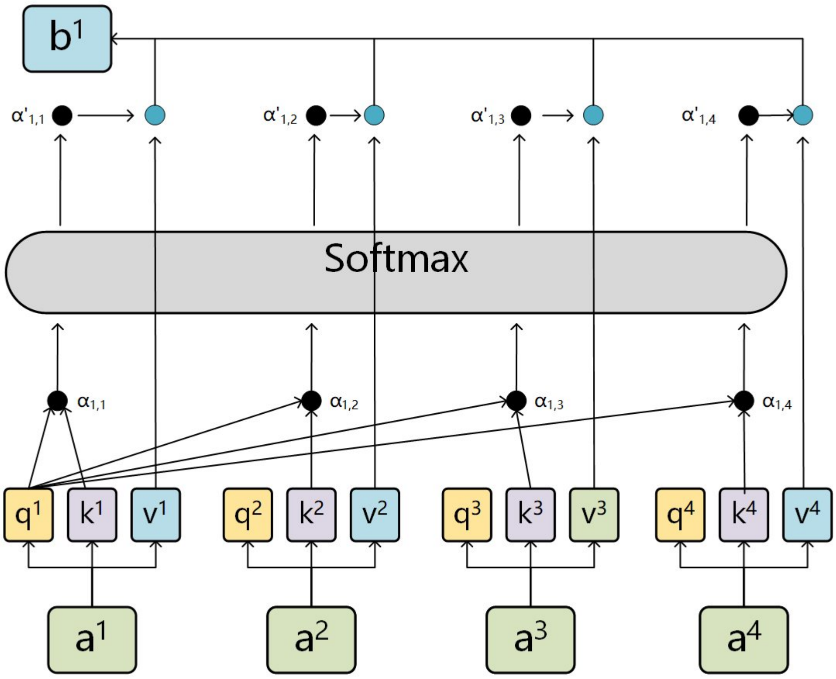


Figure 2.9: Illustration of the Scaled Dot-Product Attention mechanism in the Transformer architecture (Lee, 2019).

The Scaled Dot-Product Attention works as follows: given an input sequence $A\{a^1, a^2, a^3, a^4\}$, the attention layer maps each element a^n to its corresponding output b^n, $n \in \{1, 2, 3, 4\}$. First, trainable parameters W^Q, W^K, and W^k are used to generate corresponding query q^n, key k^n and value v^n for the input a^n, $n \in \{1, 2, 3, 4\}$. The query q^n represents the information that the attention layer is trying to output for a^n, the key k^n represents the information from a^n that the attention layer is attending to, and the value v^n holds the information that the attention layer uses to compute the output b^n. Taking the first element a^1 as an example, the attention scores $\alpha'_{1,n}$, $n \in \{1, 2, 3, 4\}$ of a^1 are calculated as the dot products of q^1 with k^n, $n \in \{1, 2, 3, 4\}$, and a softmax function (Bishop & Nasrabadi, 2006) is applied to these dot products, which are then used as weights on the values v^n, $n \in \{1, 2, 3, 4\}$, and the output of a^1 is b^1, computed as the weighted sum of values v^n. The self-attention mechanism in the Transformer allows the model to selectively attend to different parts of the input sequence and use this information to compute the output (Lee, 2019).

In fact, the attention layer in the Transformer processes a set of queries simultaneously by packing them into a matrix Q, and the keys and values are also packed into matrices K and V. This allows the Transformer to be significantly more parallelized. The Scaled Dot-Product Attention can be calculated as Eq. 2.3 (Vaswani et al., 2017).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.3}$$

where $\sqrt{d_k}$ is the scaling factor to prevent dot products grow large in magnitude.

It has been found that better performance can be achieved by projecting the queries, keys, and values h times in parallel using different sets of W^Q, W^K, and W^V (Vaswani et al., 2017). These h Scaled Dot-Product Attention layers will output results with the same dimensions, which will be concatenated and projected again.

Positional encoding

The absolute positional encoding was used in the Vanilla Transformer, which is also called sinusoidal positional encoding and can be summarized as Eq.2.4 (Vaswani et al., 2017).

$$
\begin{aligned}
PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \\
PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{\text{model}}}\right),
\end{aligned}
\tag{2.4}
$$

where pos is the position and i is the dimension. The positional encoding and the embedding results are combined to create the input for the Transformer. This allows the Transformer to include information about the position of the elements in the sequence in the input, while still maintaining the order of the sequence.

Transformer for rainfall-runoff modeling

Using Transformer-based architecture to model the relationship between rainfall and runoff is depicted in Fig. 2.10. Note that the traditional Transformer architecture takes in both a source sequence and a target sequence, which in the context of rainfall-runoff modeling represent the meteorological forcing data (concatenated with basin attributes) and discharge, respectively. However, when transferring the model to ungauged basins, the discharge data is not available, therefore, the Transformer-based rainfall-runoff modeling only utilizes the encoder part.

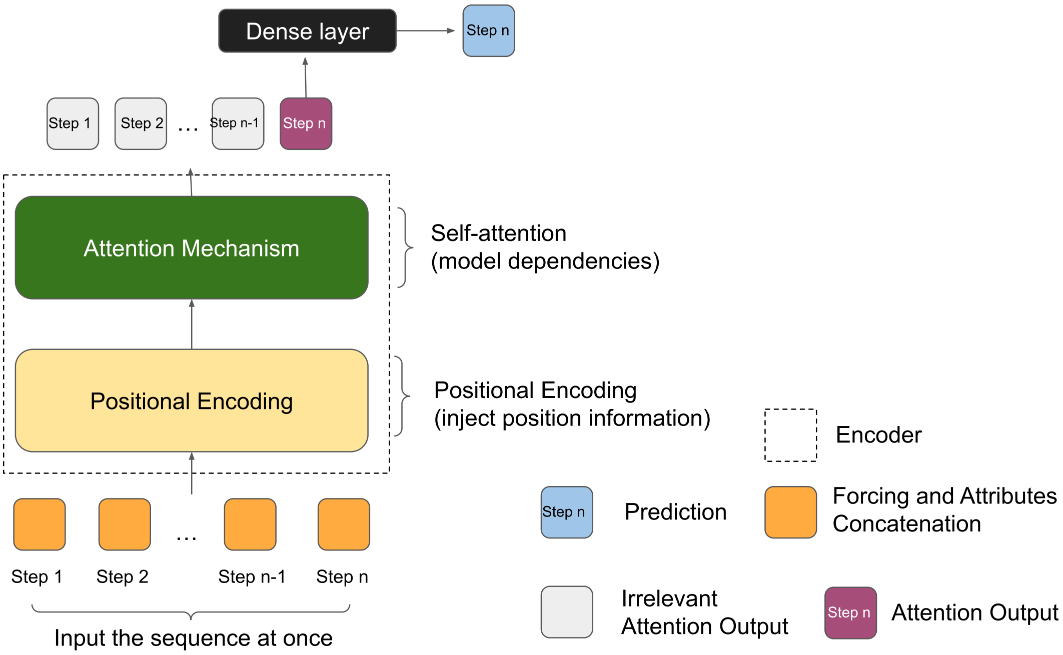


Figure 2.10: Illustration of Transformer architecture applied to rainfall-runoff modeling, where only the encoder part is utilized. The attention output at the final position (represented by the purple square) is connected to a dense layer and subsequently converted into discharge.

2.2.4. Transformer Variants

Recently, Transformer architectures have garnered a great deal of interest in time series modeling (Wen et al., 2022). However, the self-attention mechanism of the Transformer requires a large amount of quadratic computational resources for long input sequences, making it difficult to use for time series prediction (H. Zhou et al., 2021). For example, the Vanilla Transformer requires $\mathcal{O}\left(L^2\right)$ computational operations and memory in processing a length-L sequence. To address this issue, several Transformer variants with modified attention mechanisms have been proposed for time series forecasting problems, including Informer (H. Zhou et al., 2021), Reformer (Kitaev et al., 2020), FEDformer (T. Zhou et al., 2022), and Linformer (Wang et al., 2020).

Informer: is a variant of the Transformer specifically designed for time series forecasting. It aims to enhance the Transformer's prediction capacity on time-series forecasting through the incorporation of innovations including the ProbSparse self-attention mechanism, self-attention distilling operation, and a generative style decoder, in order to more efficiently model long-range dependencies in time series data.

Reformer: introduces the novel Locality-Sensitive Hashing attention to reduce the memory and time complexity into $\mathcal{O}(L \log L)$, where L is the length of the sequence. In addition, the use of reversible residual layers (Gomez et al., 2017) in the Reformer model also allows for faster and more memory-efficient computations.

FEDformer: stands for Frequency Enhanced Decomposed Transformer, it was proposed for analyzing time series data that includes a mixture of experts for seasonal-trend decomposition to better capture the global properties of the data. FEDformer is able to achieve linear $\mathcal{O}(L)$ computational complexity and memory cost by randomly selecting a fixed number of Fourier components.

Linformer: approximates the self-attention mechanism through a low-rank matrix decomposition, resulting in reduced space- and time-complexity of linear $\mathcal{O}(L)$, where L is the length of the sequence. This makes it suitable for time series modeling.

In the real world, time series modeling often involves the use of timestamps, such as calendar timestamps (e.g. second, minute, hour, week, month, year) or special timestamps (e.g. holidays, events). These timestamps can provide valuable information but are not effectively utilized by the vanilla Transformer's positional encoding method (as shown in Eq.2.4) (H. Zhou et al., 2021). To address this issue, the use of timestamps as positional encoding has been proposed.

The timestamp embedding consists of three parts, as shown in Fig 2.11 shows.

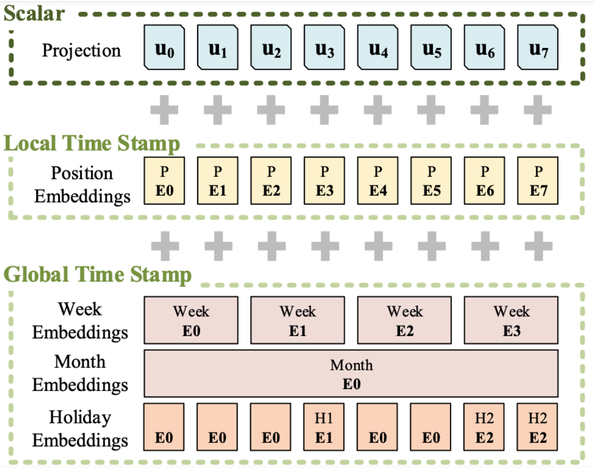

Scalar
Projection
u_0 u_1 u_2 u_3 u_4 u_5 u_6 u_7
Local Time Stamp
Position Embeddings
PE0 PE1 PE2 PE3 PE4 PE5 PE6 PE7
Global Time Stamp
Week Embeddings
Week E0 Week E1 Week E2 Week E3
Month Embeddings
Month E0
Holiday Embeddings
E0 E0 E0 H1 E1 E0 E0 H2 E2 H2 E2

Figure 2.11: Illustration of Timestamp embedding (H. Zhou et al., 2021).

The Local Time Stamp embedding is the same as the vanilla Transformer (as shown in Eq. 2.4), while the Global Time Stamp embedding will make the timestamp into [Day-Of-Week, Day-Of-Month, Day-Of-Year], whose values range from -0.5 to 0.5. For example, Tuesday is the $(\frac{1}{6} - 0.5)$ Day-Of-Week. The Positional embedding, Global Time Stamp embedding, and the original input sequences will be projected into the same dimension as the model and summed as the input of the attention mechanism.

The vanilla Transformer and its variants (i.e. Transformer Family), as well as timestamp positional encoding, will be tested for their effectiveness in rainfall-runoff modeling. Simply put, different attention mechanisms and positional encoding methods will be used to simulate discharge based on the architecture depicted in Fig. 2.10.

3 Methodology

This chapter provides a detailed overview of the data sets used in the research, and the methods to evaluate the performance of the model. Then how the experiments are designed to approach the research question will be shown.

3.1. Data set

3.1.1. CAMELS

The CAMELS dataset, which stands for ”Catchment Attributes for Large-Sample Studies,” consists of 671 catchment areas in the CONUS with minimal human disturbance. It includes catchment meteorological forcing data and daily streamflow observations starting in 1980 to 2010 for most catchments. There are three different resolution meteorological forcing products available in the dataset: Daymet (Newman et al., 2015), NLDAS (Xia et al., 2012), and Maurer (Maurer et al., 2002), which are local and generally reliable. A small proportion of the daily discharge measurements are missing for a few basins, but the meteorological forcing time series are all complete. For example, during the period 1990-2009, no more than 1% of the basins had more than 1% of their daily streamflow measurements missing (Addor et al., 2017).

The CAMELS dataset will be used to compare Transformer-based rainfall-runoff models to LSTMs. Many LSTM hydrological models have been developed using the CAMELS dataset, which has been shown to be reliable for developing deep learning-based rainfall-runoff models. Fig. 3.1 and Fig. 3.2 depict the daily mean precipitation and aridity of the CAMELS dataset, respectively.

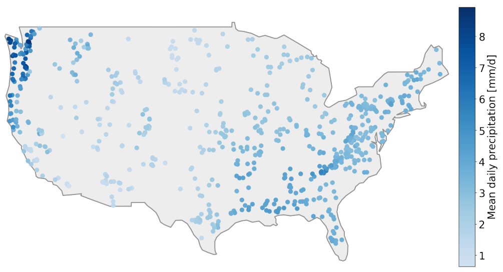


Figure 3.1: CAMELS basins mean daily precipitation, the points denote the centroid of the basins.

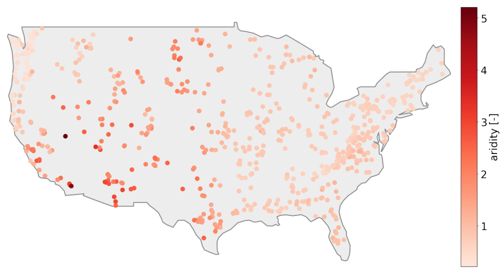


Figure 3.2: CAMELS basins aridity, the points denote the centroid of the basins.

3.1.2. Caravan

The Caravan dataset (Kratzert et al., 2022) consists of almost forty years (1981-2020) of daily meteorological forcing and discharge data, as well as basin attributes, for 2532 basins around the world. The meteorological forcing data was derived from the ERA5-Land product (Muñoz-Sabater et al., 2021), the basin attributes were taken from ERA5-Land and HydroATLAS (Linke et al., 2019), and the discharge data was sourced from seven open datasets like CAMELS and CAMELS-AUS. A list of Caravan basins and their corresponding discharge data sources can be found in 3.1.

Table 3.1: Overview of Caravan basins and sources of discharge data.

Sub-data	No. of basins	Location
CAMELS (US)	482	USA
CAMELS-AUS	150	Australia
CAMELS-BR	376	Brazil
CAMELS-CL	314	Chile
CAMELS-GB	408	Great Britain
HYSETS	323	Canada
LamaH-CE	479	Austrian territory and Danube catchment up to Bratislava

The Caravan dataset includes not only a large number of basins but also a long period of meteorological and discharge records, making it suitable for developing a global Transformer-based rainfall-runoff model. The distribution of basins in Caravan can be seen in Fig. 3.3.

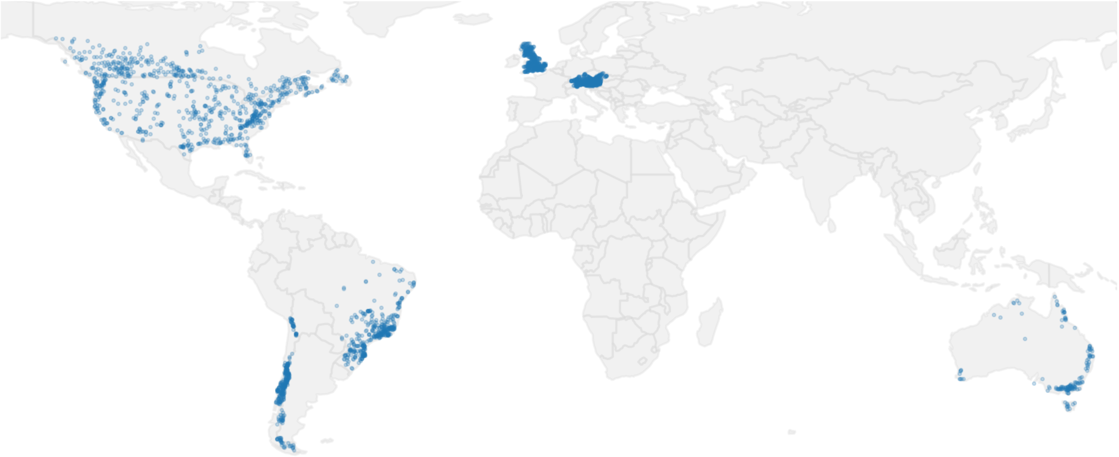


Figure 3.3: Caravan basins distribution (Kratzert et al., 2022)

3.2. Evaluation methods

The most commonly used evaluation metrics and methods for rainfall-runoff models include the Nash-Sutcliffe efficiency (NSE), peak flow bias, low flow bias, middle flow bias, the Kling-Gupta Efficiency (KGE), and the Budyko Framework, etc. These metrics and methods provide information on the accuracy and reliability of the model's ability to predict discharge and can be used to compare different models or to determine the best model for a particular application. These evaluation metrics and methods will be introduced in this section.

3.2.1. Metrics

There is no single metric that can fully evaluate the consistency, reliability, accuracy, and precision of a rainfall-runoff model, and the evaluation of hydrological models should be approached as a multi-objective problem (Efstratiadis & Koutsoyiannis, 2010). Therefore, it is necessary to estimate the model performance using multiple metrics (Gupta et al., 1998). More details about these metrics and hydrological signatures will be presented in this section. The following notations are chosen for all metric equations:

Q= discharge [mm/d]
X_o = observed X
X_m = modelled X
$\bar{Q}$ = average discharge [mm/d]
r = correlation
μ = mean
t = time step
T = number of total time step
σ = standard deviation
An overview of all the hydrological signatures and metrics can be seen in Table 3.2.

Table 3.2: Overview of the hydrological signatures and the metrics.

Metric	Abbr.	ranges	desirable value
Nash–Sutcliffe Efficiency	NSE	-$\infty - 1$	1
basin-averaged Nash–Sutcliffe efficiency	NSE*	-$\infty - 1$	1
alpha decomposition of NSE	alpha-NSE	0 - +∞	1
beta decomposition of NSE	beta-NSE	-$\infty - +\infty$	0
Kling-Gupta Efficiency	KGE	-$\infty - 1$	1
FDC High flow bias	FHV	-$\infty - +\infty$	0
FDC Midsegment slope bias	FMS	-$\infty - +\infty$	0
FDC Low flow bias	FLV	-$\infty - +\infty$	0

NSE

NSE (shown in Eq. 3.1) is used to assess the predictive skill of hydrological models. It's commonly used in evaluating the model prediction accuracy but often criticized for the overestimation of model skills in highly seasonal variables such as runoff in snowmelt-dominated basins (Gupta et al., 2009) and guides the model more focus on simulating high flow and ignoring errors in the low flow prediction.

$$NSE = 1 - \frac{\sum_{t=1}^{T}\left(Q_o^t - Q_m^t\right)^2}{\sum_{t=1}^{T}\left(Q_o^t - \bar{Q}_o\right)^2} \tag{3.1}$$

When NSE is applied in multiple basins, the NSE from a basin with a low average discharge is generally smaller than the NSE from a basin with a high average discharge. Therefore, the basin-averaged Nash–Sutcliffe efficiency (NSE*) was used, which does not overweight the basins with high average discharge (Kratzert, Klotz, Shalev, et al., 2019). NSE* can be described by Eq. 3.2.

$$\text{NSE}^* = \frac{1}{B}\sum_{b=1}^{B}\sum_{n=1}^{N}\frac{(Q_m - Q_o)^2}{(s(b)+\epsilon)^2}, \tag{3.2}$$

where B is the number of basins, N is the number of samples (days) per basin, and $s(b)$ is the standard deviation of the discharge in basin b. Note that NSE* serves only as a loss function, measuring the error between the prediction and observation from multiple basins in a batch rather than a metric to evaluate the model performance.

NSE Decomposition

NSE can be divided into two parts (see Eq. 3.3) α and β_n, α focuses on the evaluation of relative variability of the simulated and observed discharge values (Gupta et al., 2009).

$$NSE = 2 \cdot \alpha \cdot r - \alpha^2 - \beta_n^2 \\ \alpha = \sigma_m / \sigma_o \tag{3.3}$$

KGE

Kling-Gupta Efficiency (KGE) (Gupta et al., 2009) evaluates the hydrologic model performance like NSE does, and it was developed based on the limitation of NSE. KGE can be described by Eq. 3.4.

$$KGE = 1 - \sqrt{(r-1)^2 + (\alpha - 1)^2 + (\beta_n - 1)^2} \tag{3.4}$$

FLow Duration Curve

Flow Duration Curve (FDC) is a cumulative frequency curve that shows the percentage of discharge times equal to or exceeding the specified discharge times during a given period (Searcy, 1959) in a Hydrograph. 0–0.02 flow exceedance probabilities part of FDC is the high-flow segment, 0.2–0.7 flow exceedance probabilities part of FDC is the midsegment, 0.7–1.0 flow exceedance probabilities is the low-flow segment part. An example of a hydrograph and its flow duration curve can be seen in Fig. 3.4.

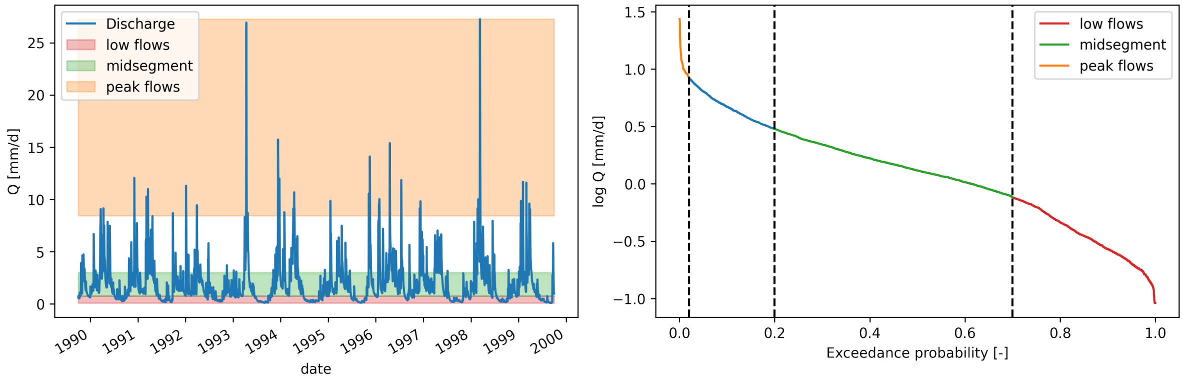


Figure 3.4: Hydrograph and flow duration curve (logarithmic y-axis).

High flow bias: To assess the hydrological model's skill in simulating exceedance percentage lower than 2% peak flow, FDC High-segment Volume (FHV) of FDC is used and can be seen in Eq 3.5 (Yilmaz et al., 2008).

$$\%\ \text{BiasFHV} \ = \frac{\sum_{h=1}^{H} (Q_{m,h} - Q_{o,h})}{\sum_{h=1}^{H} Q_{o_h}} \times 100, \tag{3.5}$$

where h = 1, 2,...H are the flow indices for flows with exceedance probabilities lower than 0.02.

Midsegment slope bias & Low flow bias: Similarly, the evaluation on model performance in mid-segment and low flow can be represented by the bias in FDC Midsegment Slope (FMS) and the bias in FDC Low-segment Volume (FLV), respectively. The two can be described by Eq. 3.6 and Eq. 3.7 (Yilmaz et al., 2008).

$$\%\ \text{BiasFMS} \ = \frac{[\log(Q_{m,m1}) - \log(Q_{m,m2})] - [\log(Q_{o,m1}) - \log(Q_{o,m2})]}{[\log(Q_{o,m1}) - \log(Q_{o,m2})]} \times 100, \tag{3.6}$$

where $m1$ and $m2$ are the lowest and highest flow exceedance probabilities (0.2 and 0.7, respectively).

$$\%\ \text{BiasFLV} \ = -1 \cdot \frac{\sum_{l=1}^{L} [\log(Q_{m,l}) - \log(Q_{m,L})] - \sum_{l=1}^{L} [\log(Q_{o,l}) - \log(Q_{o,L})]}{\sum_{l=1}^{L} [\log(Q_{o,l}) - \log(Q_{o,L})]} \times 100, \tag{3.7}$$

where l = 1,2,...L is the index of the flow value located within the low-flow segment (0.7–1.0 flow exceedance probabilities) of the flow duration curve, and L is the index of the minimum flow.

3.2.2. Paired Wilcoxon test

When comparing the performance of different models based on metrics in numerous catchments, for example, one model may have one NSE value for each of the 531 basins, resulting in 531 NSE values, so the use of mean or median values of the metrics may not accurately reflect the performance of each model. To more effectively compare the differences in the metrics obtained by different models in multiple catchments, the Wilcoxon signed-rank test (Wilcoxon, 1992) is used to assess the significance of the differences in the distribution of the metrics for each model in different catchments.

Wilcoxon signed-rank test is a nonparametric test used to compare two related samples or repeated measurements on a single sample. It is used when the assumptions of the parametric paired t-test are not met, such as when the data is not normally distributed (Wikipedia contributors, 2022).

3.2.3. Budyko Framework

Budyko framework was developed by Budyko (Budyko, 1963) to evaluate the connections and feedback on the water between climate forcing and land surface characteristics (Xu et al., 2013). The relationship between the actual evapotranspiration (AET) and runoff (Q), the change in catchment water storage (ΔS), and precipitation can be described by Eq. 3.8.

$$P = AET + Q + \Delta S \tag{3.8}$$

Budyko framework assumed that the change in storage can be neglected over long-term timescales (i.e. $\Delta S = 0$) in the long-term, and underlines that actual evapotranspiration (AET) is a function of the aridity index (ϕ) as Eq. 3.9 described. The aridity index (ϕ) is defined as the ratio between the potential evapotranspiration (PET) and the precipitation (P).

$$\frac{AET}{P} = \sqrt{\frac{PET}{P} * \tanh \frac{PET^{-1}}{P}} * \left(1 - e^{-\frac{PET}{P}}\right) \tag{3.9}$$

Budyko framework is limited by the energy and water limit (see Fig. 3.5). The energy constraint denotes there is not enough energy for more evaporation ($\overline{E_A} \leq \overline{E_P}$), and the water limit means no more water can be evaporated than has entered the catchment as precipitation ($\overline{E_A} \leq \bar{P}$).

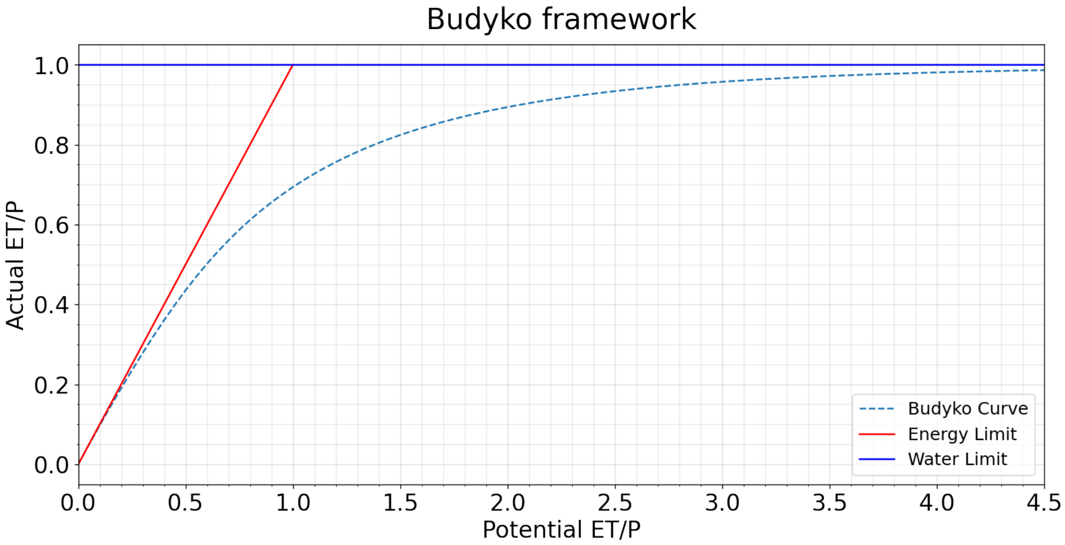


Figure 3.5: Budyko Framework.

3.3. Experiments design

Two experiments were designed to address the two sub-research questions, respectively. The first experiment, using the CAMELS (US) dataset, aims to compare the performance of Transformer models with LSTM models in rainfall-runoff modeling, particularly in snow-driven basins. This experiment will focus on regional rainfall-runoff modeling in the US. The second experiment, using the Caravan dataset, aims to evaluate the suitability of Transformer models for global rainfall-runoff modeling. This experiment will focus on global modeling.

3.3.1. Experiment 1: regional modeling

In the first experiment, the overall modeling performance of LSTM and the Transformer family will be compared on CAMELS basins. Additionally, the performance of LSTMs and Transformers in predicting storage effect basins (i.e., snow-driven basins) will be examined. By conducting this experiment, it will be possible to answer the first sub-research question: Can Transformer architecture outperform LSTM in rainfall-runoff modeling? In addition, the effectiveness of the Transformer's attention to all the precipitation events in modeling the storage effect will be evaluated.

LSTM Benchmark

In this experiment, the state-of-the-art daily LSTM-based rainfall-runoff model (Kratzert, Klotz, Shalev, et al., 2019) will be used as the benchmark. This benchmark has achieved a median NSE value of 0.73 on 513 basins in the US based on the CAMES data set and outperforms some conceptual hydrological models. The overview of the benchmark can be found in Table 3.3.

Table 3.3: Overview of the configuration for the LSTM benchmark model (Kratzert, Klotz, Shalev, et al., 2019).

	Configuration		Configuration
Model	LSTM	Loss function	NSE*
Hidden size	256	Input sequence length	270
Data	CAMELS	Output sequence length	1
Forcing product	Maurer	Training period	01/10/1999 - 30/09/2008
Forcing channel	precipitation; etc	Validation period	01/10/1980 - 30/09/1989
Attributes	see Appendix B	Test period	01/10/1989 - 30/09/1999

The LSTM benchmark model was trained using data from 531 basins in the CAMELS dataset and was then validated and tested on the same basins over different time periods (as shown in Table 3.3). The forcing channel refers to the types of dynamic meteorological forcing input used by the model, including (i) daily cumulative precipitation, (ii) daily minimum air temperature, (iii) daily maximum air temperature, and (iv) average short-wave radiation. In addition, 27 static basin attributes were used for model training, and more information can be found in the Appendix B. It is worth noting that the LSTM benchmark model was tested using different input sequence lengths (90, 180, 270, and 365 days), and achieved the highest NSE when using a 270-day input sequence. This length-270 input LSTM model will be used for comparison with Transformers in terms of general model performance.

To obtain a general comparison result in snow-driven basins, a benchmark model using a 365-day input sequence for the LSTM model was also trained.

Transformers

The Transformer family members (including vanilla Transformer, Informer, Reformer, Linformer, and FEDformer) were used for the regional rainfall-runoff modeling experiment. There are several hyperparameters that must be set prior to training, including the number of heads, number of encoder layers, number of embedding dimensions, learning rate, optimizer, etc. Due to the time-consuming nature of training Transformer models and the limited research time available, only different layers, heads, and positional encodings (including sinusoidal positional encoding and timestamp positional encoding) were experimented with for each member of the Transformer family. The input embedding dimension was fixed at 256 for the experiment, and the number of heads have to be divisible by this value, so 2, 4, 8, and 16 heads were tried. The number of encoder layers was also varied, with 2, 4, and 8 layers being tested. A summary of the hyperparameters can be found in Table 3.4.

Table 3.4: Overview of Transformer family configuration for regional rainfall-runoff modeling.

	Configuration		Configuration
Optimizer	Adam	Number of heads	2 / 4 / 8 / 16
Activation function	tanh	Input embedding dimention	256
Epoch	100	Learning rate	1e-4 ∼1e-3
Number of layers	2 / 4 / 8	Position encoding	Timestamp / Sinusoidal

Note: Learning rates change over epochs.

The input for the Transformer family models was the same as the LSTM benchmark, including the same basins, forcing channels, attributes, training, validation, and test period. A 365-day input sequence was used to capture at least the dynamics of a full annual cycle. The performance of the Transformers will be compared with the LSTM benchmarks using the previously mentioned evaluation methods.

This experiment can be illustrated in Fig. 3.6.

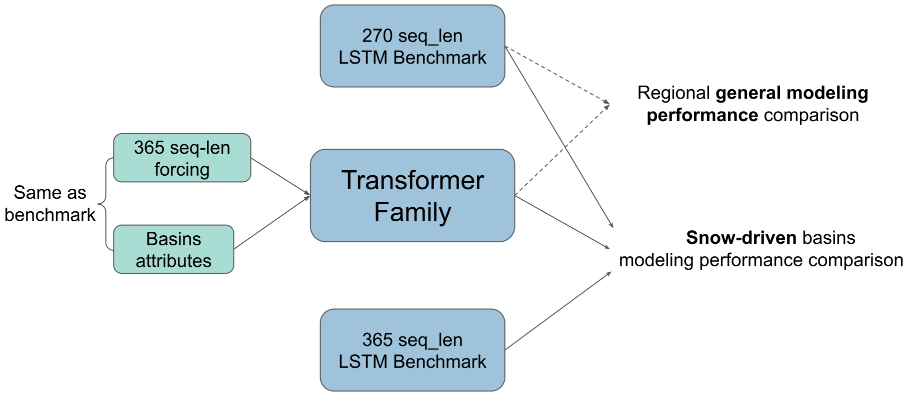


Figure 3.6: The flow chart illustrates the process of the first regional experiment.

3.3.2. Experiment 2: global modeling

The hypothesis for the second experiment is that a larger training data size is generally better for Transformer, and Transformer is able to learn a wide range of rainfall-runoff behaviors from various basins around the world, allowing it to be used as a global rainfall-runoff model to address PUB. The goal is to build a global Transformer-based rainfall-runoff model using the Caravan dataset. However, it is recognized that the Caravan dataset may contain uncertainty, and the sources of this uncertainty will also be examined in order to inform the modeling process.

LSTM Benchmark

Currently, there is no globally-trained DL-based rainfall-runoff model. However, the Transformer architecture is expected to achieve very good performance when trained on the global Caravan data. Therefore, another LSTM-based regional rainfall-runoff model (Gauch et al., 2021), developed on CAMELS (US) data with both hourly and daily forcing inputs and predicting discharge at a daily scale, was selected as the benchmark. This benchmark shows better prediction skills on the same 531 basins than the LSTM in the first experiment, with the only difference being the additional hourly meteorological forcing input. This benchmark will be compared with the US part of the global Transformer model. The configuration overview of the LSTM benchmark with multiple timescales input can be seen in Table 3.5.

Table 3.5: Overview of LSTM benchmark configuration in the global modeling experiment (Gauch et al., 2021).

	Configuration		Configuration
Model	LSTM	Loss function	NSE*
Data	CAMELS	Output sequence length	1
Forcing product	NLDAS	Training period	01/10/1990 - 30/09/2003
Forcing channel	precipitation; etc	Validation period	01/10/2003 - 30/09/2008
Attributes	see Table 3.7	Test period	01/10/2008- 30/09/2018

Transformer

The global Transformer-based rainfall-runoff model will be trained on the Caravan data set, which includes various types of meteorological forcing and attributes not present in CAMELS. In order to fairly compare the model to the LSTM benchmark, it will be trained with the same forcing (see Table 3.6) and attributes (see Table 3.7) as the benchmark whenever possible. The training, validation, and test periods will also be identical to those of the LSTM benchmark (see in Table 3.5). The performance of the US part of the Transformer global model will then be compared to that of the benchmark in order to determine whether Transformer can benefit from a larger dataset with diverse rainfall-runoff behaviors.

Table 3.6: The dynamic forcing used in Transformer global modeling.

Feature (ERA5-Land variable name)	Aggregation
total_precipitation_sum	Daily sum Precipitation
temperature_2m_mean	Daily mean air temperature
surface_pressure_mean	Daily mean surface pressure
surface_net_solar_radiation_mean	Daily mean shortwave radiation
surface_net_thermal_radiation_mean	Daily mean net thermal radiation at the surface
potential_evaporation_sum	Daily sum potential evaporation
u_component_of_wind_10m_mean	Eastward wind component daily mean
v_component_of_wind_10m_mean	Northward wind component daily mean

Table 3.7: The static basins attributes used in Transformer global modeling.

Variable Name in Caravan	Description
p_mean	Mean daily precipitation
pet_mean	Mean daily potential evaporation
aridity	Aridity index, ratio of mean PET and mean precipitation
seasonality	Moisture index seasonality in range [0, 2], where 0 indicates no changes in the water/energy budget throughout the year and 2 indicates a change from fully arid to fully humid.
frac_snow	Fraction of precipitation falling as snow
high_prec_freq	Frequency of high precipitation days, where precipitation ≥ 5 times mean daily precipitation
high_prec_dur	Average duration of high precipitation events (number of consecutive days where precipitation ≥ 5 times mean daily precipitation
low_prec_freq	Frequency of low precipitation days, where precipitation <1 mm/day
low_prec_dur	Average duration of low precipitation events (number of consecutive days where days precipitation <1 mm/day
ele_mt_sav	mean Elevation
slp_dg_sav	mean Terrain slope
for_pc_sse	mean Forest cover extent
swc_pc_syr	annual mean soil water content
snd_pc_sav	mean Sand fraction in soil
slt_pc_sav	mean Silt fraction in soil
cly_pc_sav	mean Clay fraction in soil

To further investigate the forcing uncertainty impact on the Transformer-based rainfall-runoff model, the global coverage ERA5-land forcing was compared with the local Maurer forcing. Additionally, the individual forcing channels of the Caravan dataset (precipitation, vapor pressure, and radiation) are replaced with the corresponding channels in Maurer forcing to identify which forcing channels are the most influential or uncertain for the modeling process.

The second global modeling experiment can be summarised by Fig. 3.7.

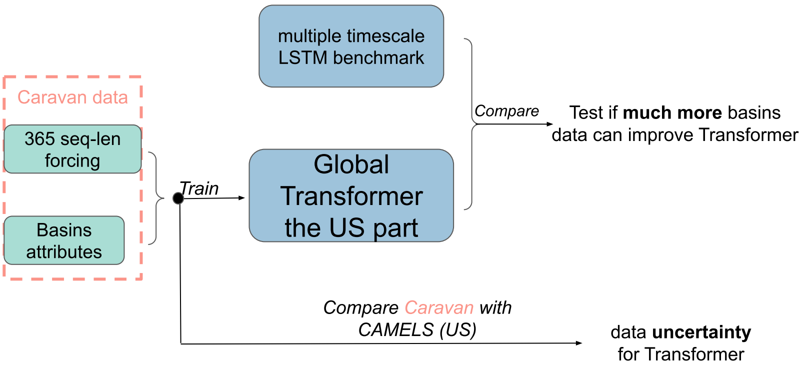

multiple timescale LSTM benchmark
Caravan data
365 seq-len forcing
Basins attributes
Train
Global Transformer the US part
Compare
Test if **much more** basins data can improve Transformer
Compare Caravan with CAMELS (US)
data **uncertainty** for Transformer

Figure 3.7: Second global experiment scheme.

3.4. Research equipments

The training of the Transformers requires significant computational resources, particularly a powerful GPU. As a result, the experiment had to be conducted using the newly launched supercomputer cluster DelftBlue ((DHPC), 2022), which provided powerful NVIDIA Tesla V100S GPUs. The NeuralHydrology (Kratzert, Herrnegger, et al., 2019) package was also utilized for data splitting, loading, training, and testing. A training configuration can be seen in Appendix A.

4
Results and discussion

This chapter is organized as follows:

1. Section 4.1 compares different Transformers and the LSTM benchmark in regional rainfall-runoff modeling in the US. The focus of this comparison is to determine if Transformer-based architecture can outperform LSTM in modeling rainfall-runoff behaviors, particularly in snow-driven basins in the US.
2. Section 4.2 presents the results of Transformer global modeling, including the impact of global ERA5-land and Maurer forcing on model performance, and identifies which forcing channel contains the most uncertainty for Transformer-based rainfall-runoff modeling.
3. Section 4.3 discusses the results of the experiments and highlights some of the limitations present in the experiment.

4.1. Regional modeling

The comparison between Transformers and LSTMs in the US regional rainfall-runoff modeling will be conducted in two parts. The first part will compare the general performance of Transformers and LSTM. The second part will compare the performance of the two models in snow-driven basins. This will provide a broad overview of the Transformer architecture's performance as a rainfall-runoff model in relation to LSTM, as well as an examination of whether the Transformer's attention on all precipitation events can improve its predictions in snow-driven basins.

4.1.1. General comparison

After attempting various configurations of the Transformer family members, including variations in the number of heads, encoder layers, and encoding methods (as shown in Table 3.4), the top-performing models for each member were selected based on median NSE during the test period, excluding some models that were unable to be trained due to too many parameters or stopped training after a few epochs because of the limited memory. Details about each family member's number of layers, heads, trainable parameters, and positional encoding methods can be seen see in Appendix D.1. Fig. 4.1 shows the Cumulative Density Functions (CDF) of NSE values for LSTM and the Transformer family across basins. The CDF shows that Reformer architecture demonstrates improved NSE in specific basins compared with LSTM. Some hydrographs that show the Transformer and LSTM simulation can be seen in Fig. 4.2, and more hydrographs can be seen in Appendix E.

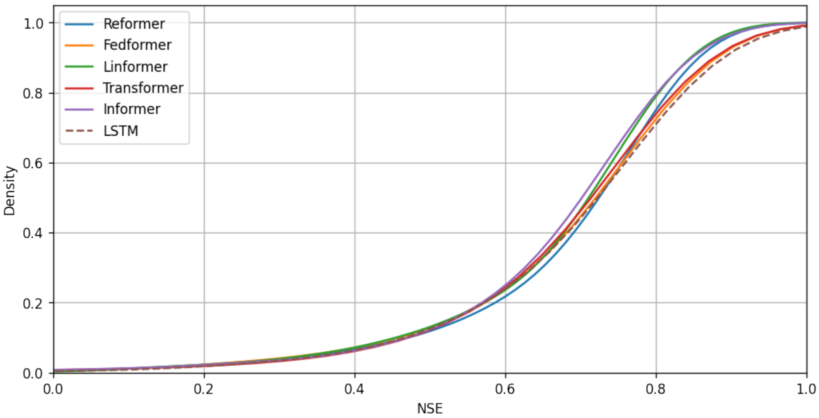


Figure 4.1: Cumulative density functions of NSE for Transformers and LSTM benchmark (Kratzert, Klotz, Shalev, et al., 2019) in 531 CAMELS basins. NSE is capped at 0 for better visualization.

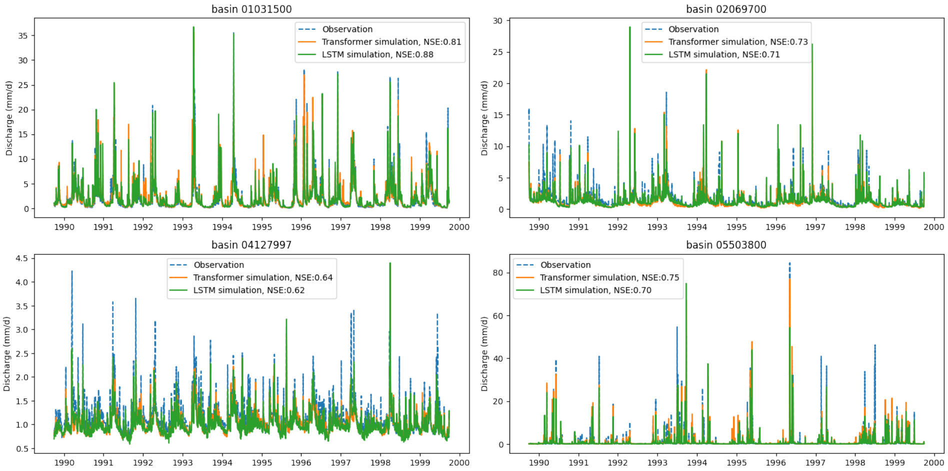


Figure 4.2: The hydrographs that show the LSTM and Transformer simulations vs the observations.

The median values of various metrics and hydrological signatures for the Transformer family, and LSTM have listed in Table 4.1, along with the average training time for each. The box plots that show each metric distribution can be seen in Appendix D. It can be observed that while none of the Transformers significantly outperform the LSTM benchmark across most metrics, they all require more training time than LSTM.

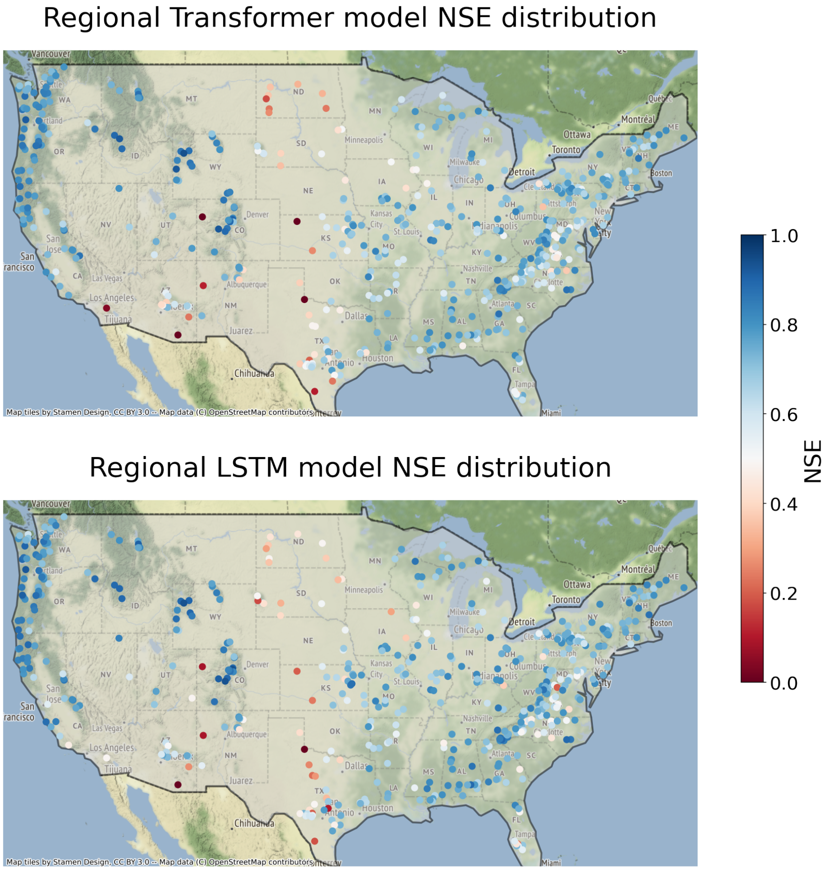
Table 4.1: Median metrics and hydrological signatures for the Transformer family and LSTM benchmark. Bolded values indicate results that are significantly different from the LSTM benchmark model in the respective metric or hydrological signature according to Wilcoxon signed-rank test (Wilcoxon, 1992) ($\alpha = 0.001$). The average training time is also included but is not related to the significance.

	Reformer	FEDformer	Linformer	Transformer	Informer	LSTM
NSE	0.728	0.732	0.715	0.725	0.709	0.732
Alpha-NSE	0.822	0.858	0.866	0.834	0.817	0.842
Beta-NSE	-0.050	**-0.011**	**-0.016**	**-0.006**	-0.032	-0.039
FHV	-17.376	-14.263	-13.368	-15.900	-17.717	-16.058
FLV	**-19.797**	**-1.806**	**-177.769**	24.667	3.661	28.927
FMS	**3.385**	-9.191	**-11.540**	-13.742	-7.671	-8.021
Training time (min/epoch)	93.536	63.786	15.900	64.307	16.034	7.693

In the experiment, Reformer, FEDformer, and Transformer took significantly longer to train compared to the other Transformers and LSTM. This is due to an inefficient coding method used. As previously mentioned, the experiment was carried out using the NeuralHydrology package, which does not allow for the option to load timestamps into the model. However, these Transformer variants require timestamp positional encoding methods to achieve higher median NSE. To accommodate for this, the code of NeuralHydrology was modified in an inefficient way, causing the timestamp data type to be transformed multiple times (from NumPy to tensor). A significant portion of the training time for these Transformers is a result of this inefficient timestamp data type transformation.

The 531 basins' NSE values spatial distribution from Transformer and LSTM can be seen in Fig. 4.3, the Transformer and LSTM NSE spatial distribution pattern are similar: in the eastern part of the United States, the basins have the highest NSE values, whereas the central US has the lowest NSE values.

The spatial distribution of NSE values for 531 basins from the Transformer and LSTM models can be seen in Fig. 4.3, and the difference of the NSE between the Transformer and the LSTM benchmark can be seen in Fig. 4.4. The patterns of the Transformer family and LSTM NSE distributions are similar according to Fig. 4.3, with the eastern part of the United States having the highest NSE values and the central US having the lowest NSE values. The difference map of NSE shows that Transformer can only achieve better NSE than LSTM in a small number of basins, and these basins are mainly concentrated in the central region of the United States.



Regional Transformer model NSE distribution

Regional LSTM model NSE distribution

1.0
0.8
0.6
0.4
0.2
0.0
NSE
Map tiles by Stamen Design, CC BY 3.0 -- Map data (C) OpenStreetMap contributors

Figure 4.3: Spatial distribution of NSE values for Transformer and LSTM in 531 basins, with locations marked on the map. The color maps are limited to a range of [0, 1] for better visualization. The NSE with a value over 0.5 is considered acceptable.

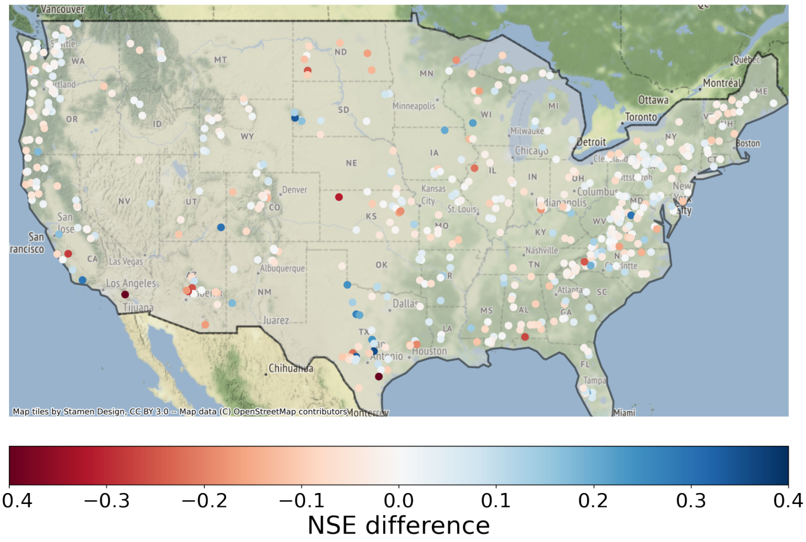

Map tiles by Stamen Design, CC BY 3.0 — Map data (C) OpenStreetMap contributors
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
NSE difference

Figure 4.4: The difference of the NSE between the Transformer and the LSTM benchmark model, blue colors (>0) indicate that the Transformer performs better than the LSTM benchmark model, red (<0) the other way around). The color map is limited to [−0.4, 0.4] for better visualization.

The NSE values distribution patterns of the Transformer and LSTM under the Budyko framework can also be seen in Fig. 4.5.

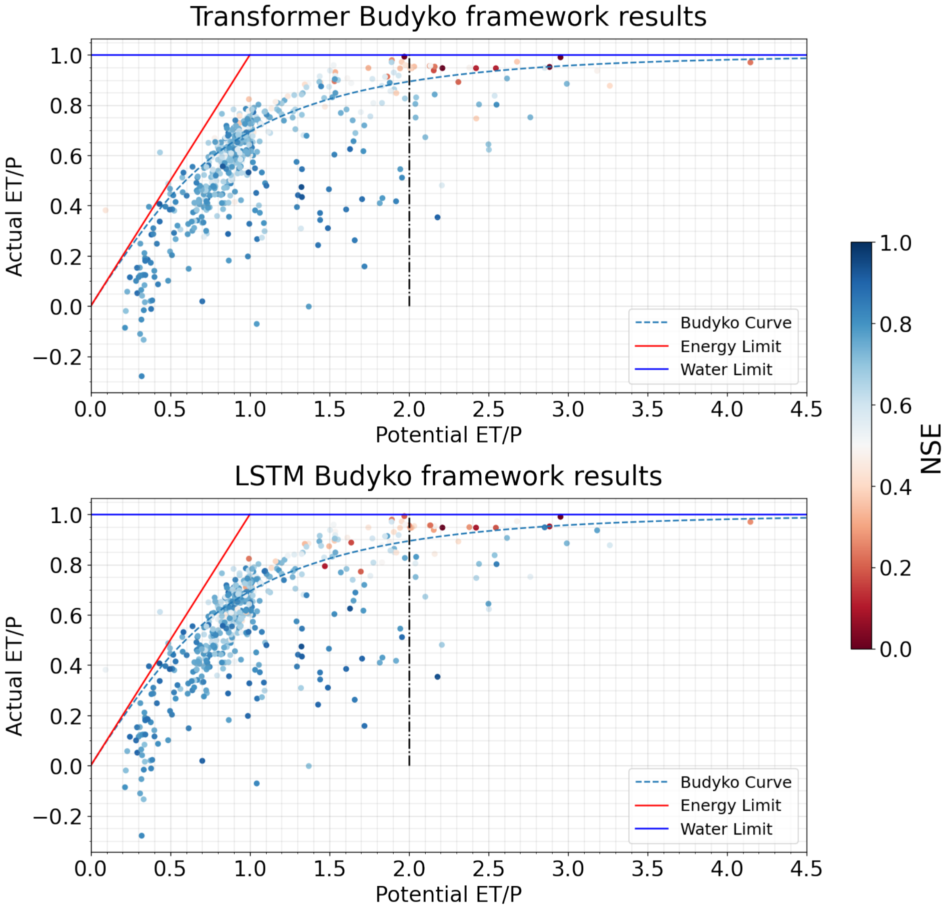


Figure 4.5: Transformer and LSTM Budyko Framework, the color maps are limited to [0, 1] for better visualization. The NSE with a value over 0.5 is considered acceptable.

Based on Fig. 4.5, it appears that both Transformer and LSTM have difficulty with discharge forecasting in (semi-) arid basins where the aridity index (Potential ET/P) and Evaporative index (Actual ET/P) are higher. This is particularly evident when the aridity index exceeds 2 (as indicated by the dotted line). The poor performance of the two models may be due to uncertainty in discharge measurement. An example of Transformer and LSTM discharge simulation in an arid basin can be seen in Fig. 4.6. It is common for flow observations to be small but may suddenly increase to very large values, as shown in Fig. 4.6. The discharge of CAMELS (US) is not measured directly, but rather stages are measured and discharge is decided using a rating curve which shows the relationship between stage and flow rate (as Fig. 4.7shows). The rating curve is created from multiple observations of stage and discharge, but due to the low frequency of high flow, errors in discharge decisions by rating curve are often larger. As a result, LSTM and Transformer perform poorly in arid basins.

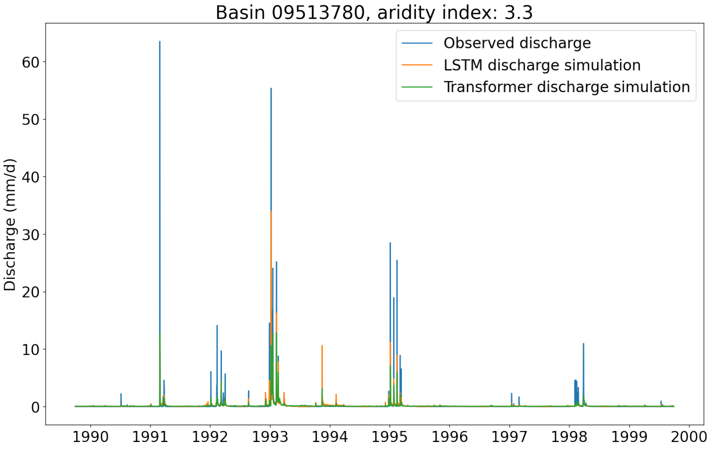


Figure 4.6: Transformer and LSTM discharge simulation in an arid basin.

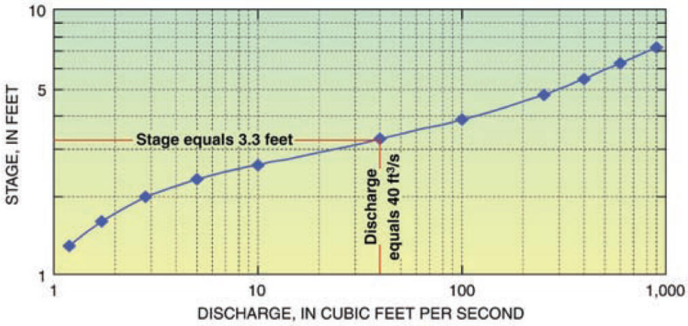


Figure 4.7: Illustration of a rating curve (USGS, 2018).

An NSE value above 0.7 is considered good model performance while a value below 0.7 indicates poor performance. After using the Transformer, the NSE values of 32 basins in the LSTM benchmark model results improved from below 0.7 to above 0.7, indicating that the Transformer transforms them from poor to good. Conversely, LSTM also improved the modeling performance of 52 basins in the Transformer model results, transforming them from poor to good. Therefore, the possibility of combining the two models for improved prediction was also explored by feeding both models the same input data and having them predict discharge together, with the hope that a dense layer could ”learn” how to utilize the combined information and make more accurate predictions. This approach is shown in Fig. 4.8, with both Transformer and LSTM receiving a 365-length input.

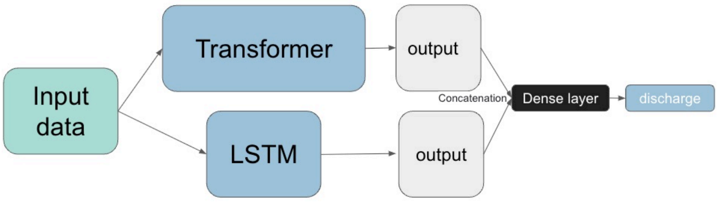


Figure 4.8: The illustration of the combination model using both Transformer and LSTM. The two models are fed the same input data and their outputs are concatenated and passed through a dense layer for prediction.

Table 4.2 shows the median metrics and their standard deviation for the combined and single LSTM models. It is evident that, with the exception of FLV, most of the median metrics are improved by the combined model compared to a single LSTM. This is similar to the findings of the previous study (Kratzert, Klotz, Shalev, et al., 2019), where feeding the same data into 8 LSTMs and taking the mean of the predictions resulted in better median NSE than a single LSTM, but also led to worse performance in other median metrics. However, the Transformer and LSTM combination used in this study was able to improve both median NSE and other median metrics.

Table 4.2: Overview of median metrics and hydrological signatures and the standard deviation for the combined model and single LSTM. Bolded values indicate results that are significantly different from the LSTM benchmark model in the respective metric or hydrological signature according to Wilcoxon signed-rank test ($\alpha = 0.001$).

	Combine Transformer & LSTM		LSTM	
	Median	Standard deviation	Median	Standard deviation
NSE	0.747	0.163	0.732	0.143
Alpha-NSE	0.864	0.163	0.842	0.163
Beta-NSE	**-0.012**	0.067	-0.039	0.068
FHV	**-12.816**	15.212	-16.058	15.538
FMS	-8.875	4.805+07	-8.021	3.781e+06
FLV	**-228.202**	4.842e+10	28.927	1.234e+11
Training time (min/epoch)	40.031		7.992	

4.1.2. Snow-driven basins comparison

Snow-driven basins, which tend to have slower discharge responses to precipitation compared with arid basins and therefore require longer periods of historical meteorological data as input for modeling the rainfall-runoff relationship, will be used to evaluate the performance of the Transformer versus LSTM. Transformer's self-attention mechanism, which allows it to consider all precipitation events, may give it an advantage in modeling discharge in these types of basins. The snow-driven basins are defined as the basin with high frac_snow (>0.5) attribute in CAMELS, which means the fraction of precipitation falling as snow is over 50%. The distribution of the snow-driven basins can be seen in Fig. 4.9. The overview of the basins can be seen in Table 4.3.

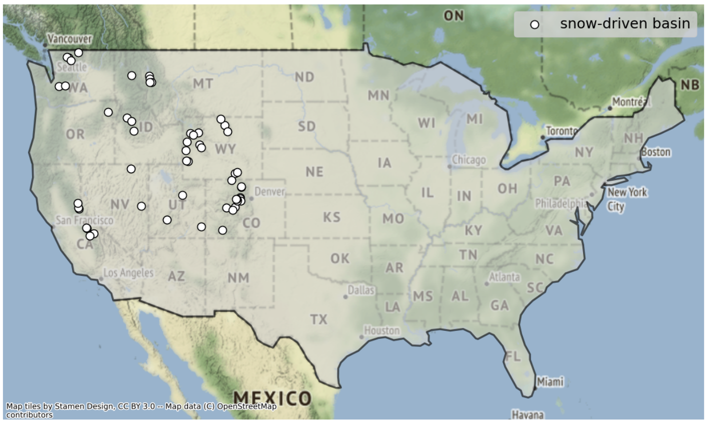


Figure 4.9: CAMELS snow-driven basins distribution

Table 4.3: Overview of snow-driven basins.

	Number	Catchments mean elevation [m above sea level]
Snow-driven basins	55	2489.7

Snow-driven basins tend to experience a significant peak flow each year due to the melting of the snow that is prevalent in these areas from June to August, as shown in Fig. 4.10.

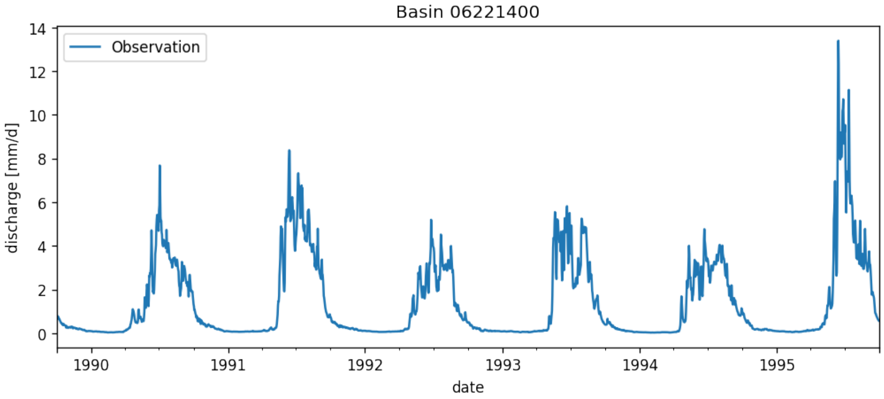


Figure 4.10: Hydrograph of snow-driven basin 06221400 in CAMELS.

To allow for a more general conclusion, an LSTM trained on input sequences of length 365 was used as a benchmark because the snow-driven basins' discharges are influenced by the very earlier precipitation events. The difference between the two LSTMs is the input sequence length. The comparison is focused on the FDC signatures (FHV, FMS, and FLV) and alpha-NSE rather than NSE because these metrics can show the model skills in predicting the peak, middle, low flow, and flow relative variability, respectively.

Among all the Transformers and the variants, the Reformer model (hyperparameters can be seen in Table: D.1) with timestamp positional encoding usually achieved better performance. The median metrics of the Reformer and the two LSTM in snow-driven basins can be seen in Table 4.4. The Reformer has better median metrics than the two LSTM models except for NSE. As mentioned before, LSTM with length-270 achieved the best median NSE over all the basins, which is consistent in the snow-driven basins (i.e. length-270 input LSTM still has the highest median NSE), and all metrics medians of LSTM (270 seq-len) got better in the snow basin than in all basins (compared with Table

4.1). Besides, it seems the longer input sequence helps LSTMs achieve better median metrics, except for NSE, by comparing the two LSTMs.

Table 4.4: Overview of the median metrics and hydrological signatures of two LSTM and Reformer regional models in snow-driven basins. The LSTM (270 seq-len) metrics are from (Kratzert, Klotz, Shalev, et al., 2019), while LSTM (365 seq-len) was trained ourselves. Bolded values indicate results that are significantly different from the Reformer model in the respective metric or hydrological signature according to Wilcoxon signed-rank test ($\alpha = 0.001$).

	Reformer	LSTM (270 seq-len)	LSTM (365 seq-len)
NSE	0.804	**0.830**	0.821
Alpha-NSE	0.965	**0.874**	**0.892**
FHV	-2.872	**-13.801**	**-11.354**
FMS	-3.244	-6.714	-6.008
FLV	12.376	46.989	-13.917

The Reformer was compared with the two LSTM models basin by basin. The results can be seen in Fig. 4.11 and Fig. 4.12. It is discernible that Reformer performs better than LSTM in capturing the top 2% peak flow, top 30% low flow, and the relative flow variability than both LSTMs in most basins.

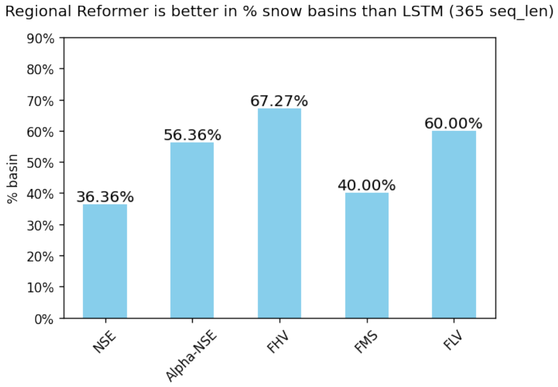


Figure 4.11: Basin-by-basin metrics comparison in snow-driven basins: Reformer vs LSTM (365 seq-len), the numbers above each bar indicate the proportion on all snow basins that Reformer is better than LSTM on the x-axis metric.

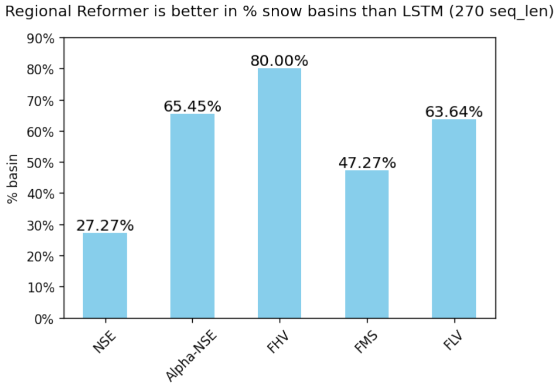


Figure 4.12: Basin-by-basin metrics comparison in snow-driven basins: Reformer vs LSTM (270 seq-len), the numbers above each bar indicate the proportion on all snow basins that Reformer is better than LSTM on the x-axis metric.

To get a general result, the two LSTMs and the Reformer on the 55 snow-driven basins were retrained on the 55 snow-driven basins. The results of the metrics and hydrological signatures of the three models can be seen in Table 4.5.

Table 4.5: Overview of the median metrics and hydrological signatures of two retrained LSTM and the retrained Reformer models in snow-driven basins. Bolded values indicate results that are significantly different from the Reformer model in the respective metric or hydrological signature according to Wilcoxon signed-rank test ($\alpha = 0.001$).

	Reformer	LSTM (Retrained 270 seq-len)	LSTM (Retrained 365 seq-len)
NSE	0.823	0.840	0.836
Alpha-NSE	0.950	**0.903**	**0.916**
FHV	-9.271	-11.379	-10.998
FMS	9.415	**-7.034**	**-3.592**
FLV	28.862	1.943	**-45.137**
Training time (min/epoch)	14.329	1.047	1.230

The comparison between Reformer and LSTMs can be seen on the basin-by-basin basis in Fig.

4.13 and Fig. 4.14, with all models being retrained on the 55 snow-driven basins.

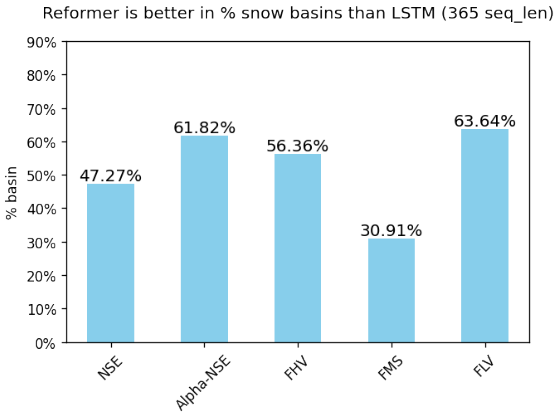


Figure 4.13: Basin-by-basin metrics comparison in snow-driven basins: Retained Reformer vs Retrained LSTM (365 seq-len), the numbers above each bar indicate the proportion on all snow basins that retrained Reformer is better than the retrained LSTM on the x-axis metric.

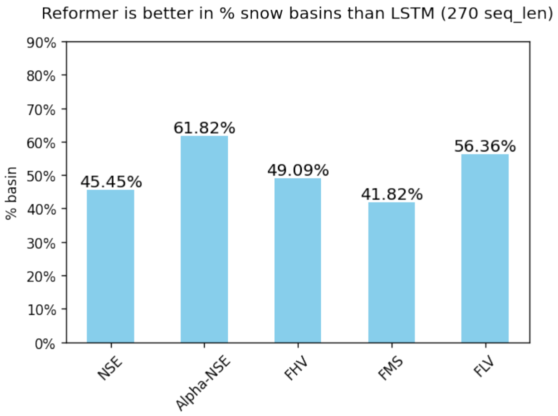


Figure 4.14: Basin-by-basin metrics comparison in snow-driven basins: Retained Reformer vs Retained LSTM (270 seq-len), the numbers above each bar indicate the proportion on all snow basins that retrained Reformer is better than the retrained LSTM on the x-axis metric.

The comparison of the Reformer model with two LSTM models showed that Reformer usually performs better in alpha-NSE, FHV, and FLV. However, when retraining the models on only snow-driven basins, the Reformer's performance relative to the LSTMs was not as significant as when it was trained on all 531 basins. This may be due to the smaller size of the training dataset (i.e. only 55 training basins were used to train the Reformer).

4.2. Global modeling

To answer the second sub-research question: Can Transformer be a global rainfall-runoff model based on the Caravan data set? The second experiment was conducted using the Transformer architecture to determine if it could learn more rainfall-runoff behaviors from larger data sets, potentially enabling it to function as a global rainfall-runoff model. In this experiment, the Transformer was trained using more than two thousand basins of global LSH data from Caravan. For comparison, an LSTM model was also trained with the same configuration. Due to the length of time required to train a global Transformer-based rainfall-runoff model (nearly one week), and the fact that the Reformer model, which performed well in the first experiment, had run out of memory in the global modeling, only the vanilla Transformer with the sinusoidal positional encoding was trained to be a global model. As a result, only two global rainfall-runoff models (Transformer and LSTM) were obtained, and the medians of their metrics and hydrological signatures are shown in Table 4.6.

Table 4.6: Overview of the two global rainfall-runoff model median metrics.

	Transformer	LSTM
NSE	0.511	0.539
KGE	0.544	0.565
Alpha-NSE	0.721	0.732
Beta-NSE	-0.046	-0.008
Pearson-r	0.767	0.778
FHV	-27.609	-26.850
FMS	-18.425	-21.003
FLV	26.965	30.419
Training time (min/epoch)	95.130	55.464

The overview performance of the two models can be seen in Fig. 4.15 and Fig. 4.16. The Caravan sub-areas median NSE comparison between the two global models can be seen in Fig 4.17.

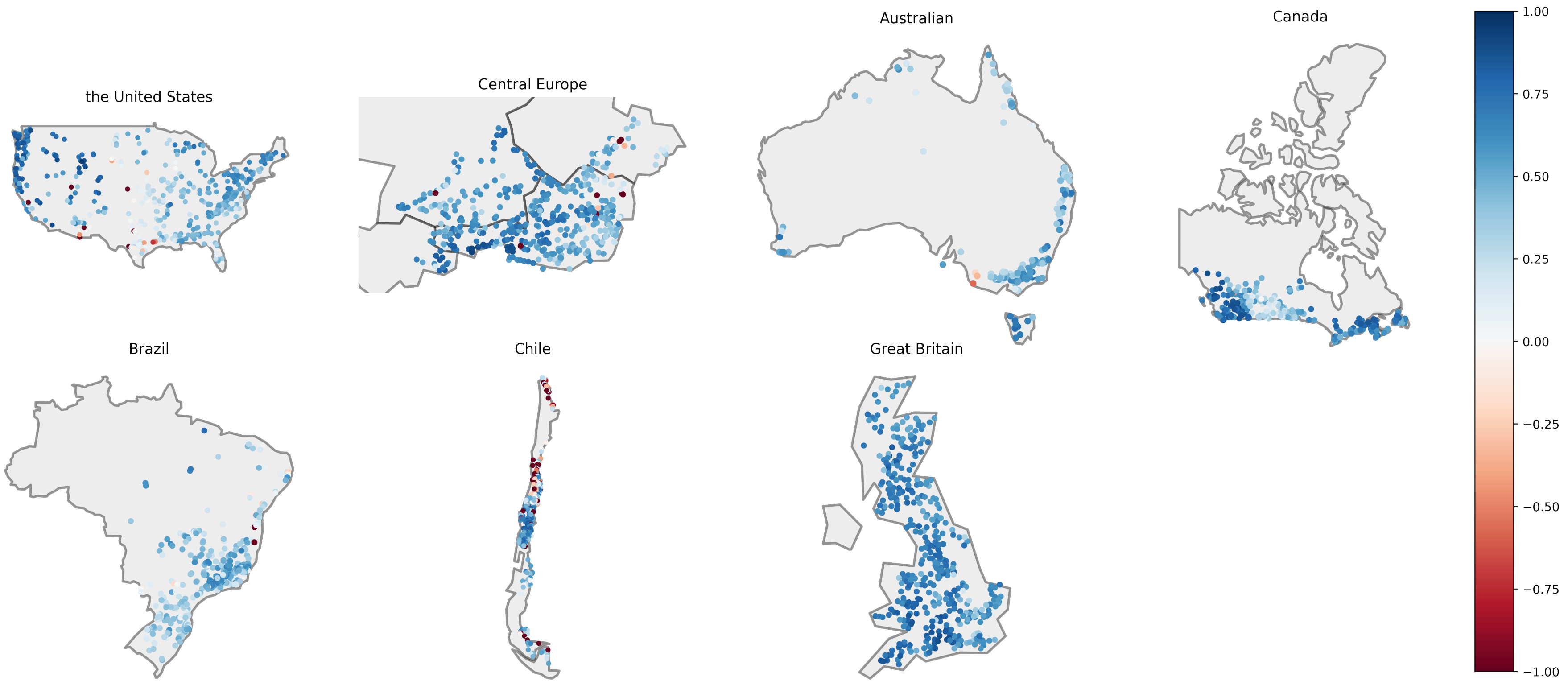


Figure 4.15: The NSE distribution of sub-regions of Caravan as simulated by the global LSTM model, the color maps are limited to [-1, 1] for better visualization. The NSE with a value over 0.5 is considered acceptable.

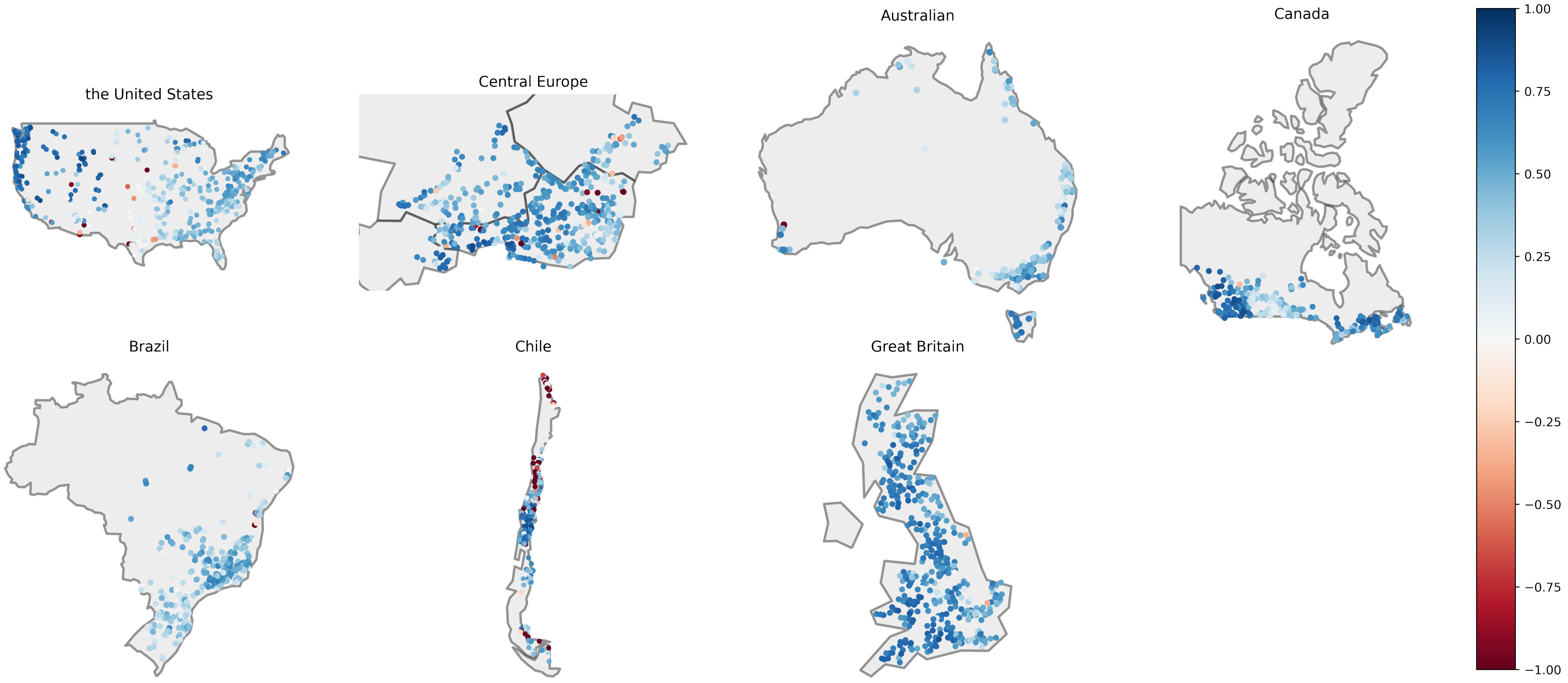


Figure 4.16: The NSE distribution of sub-regions of Caravan as simulated by the global Transformer model, the color maps are limited to [-1, 1] for better visualization. The NSE with a value over 0.5 is considered acceptable.

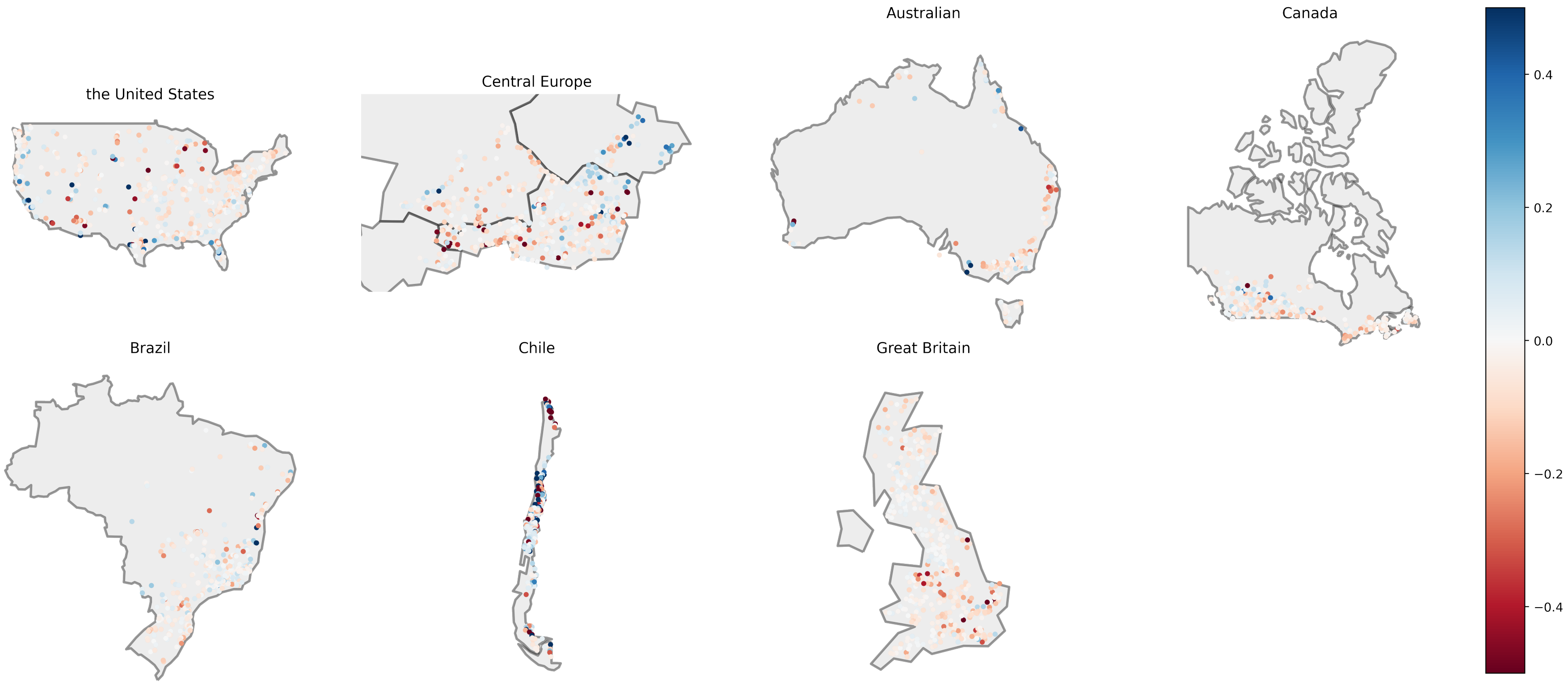


Figure 4.17: The difference of the NSE between the global Transformer and the global LSTM model, blue colors (>0) indicate that the Transformer performs better than the LSTM benchmark model, red (<0) the other way around). The color map is limited to [−0.4, 0.4] for better visualization.

As can be seen from Fig. 4.17, in the United Kingdom, Canada, and Australia regions, the results of LSTM are better than the Transformer (light red dots clearly dominates). However, Transformer appears to have better modeling results in the northeastern part of Central Europe than LSTM.

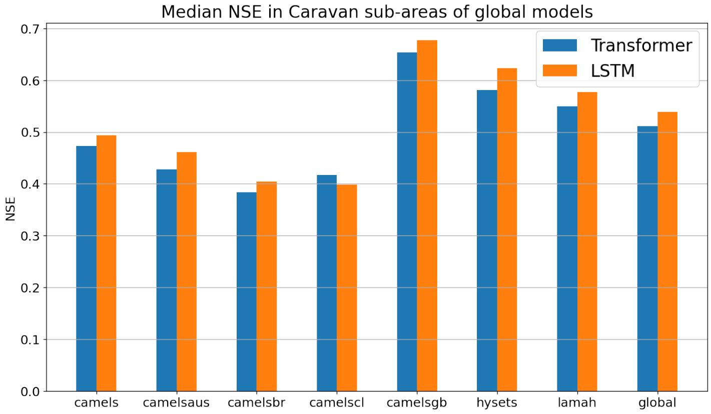


Figure 4.18: The Caravan sub-areas meidan NSE comparison.

Upon closer examination of the results of the Transformer global model for the United States (as shown in Fig. 4.19), it was found that, although the pattern is similar to the previous first experiment (with higher NSE in the western basins), the median NSE of the Transformer model in the US region is significantly lower than the multiple timescale LSTM benchmark (with a median NSE of 0.75). This suggests that the Caravan data set may not be of as high quality as CAMELS. As a result, no comparison with the benchmark was made. Note that the US portion of the Caravan data used for the Transformer-based global rainfall-runoff model was derived from the CAMELS data set, but the basin attributes and meteorological forcing were sourced from the global ERA5-Land data in Caravan. This change in input (from local to global) may have caused a degradation in modeling performance.

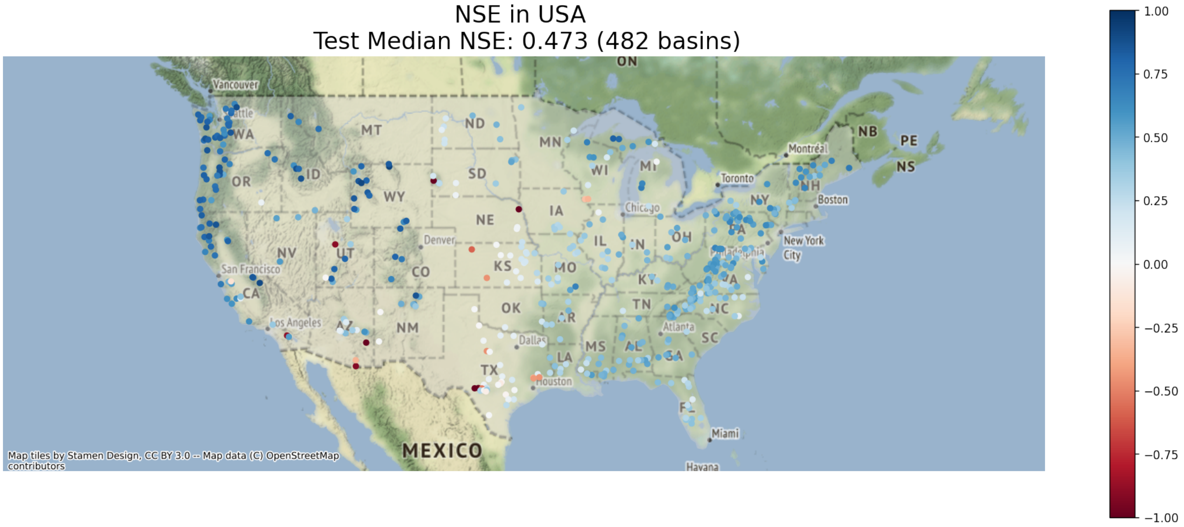


Figure 4.19: Global Transformer-based rainfall-runoff model the US part NSE distribution, the color map is limited to [-1, 1] for the NSE differences for better visualization. The NSE with a value over 0.5 is considered acceptable.

To verify the conjecture that the large difference in median NSE between the global Transformer model and the LSTM benchmark in the US is caused by the uncertainty of the Caravan data set, an experiment was conducted with the following three steps. First, the US portion of the Caravan data was trained alone to eliminate the possibility that the poor performance of the Transformer global model in the US region was due to data from other areas in the Caravan data set. Next, the Caravan catchment attributes were replaced with the CAMELS data (i.e., the basin attributes were changed from global to local Maurer product) to determine if the degradation in modeling performance was caused by the global coverage of the basin attributes. Finally, the Caravan meteorological forcing was replaced with CAMELS Maurer forcing based on the second step, to determine if the performance degradation was due to the meteorological forcing. A summary of the three steps of the above experiment and the results can be seen in Table 4.7.

Table 4.7: The overview of test the how the local and global Meteorological Forcing and basin attributes impact Transformer global modeling.

Step	Model	Forcing	Attributes	median NSE
-	Global Model (trained on 2532 basins)	Global (ERA5-land)	Global (ERA5-land & HydroATLAS)	0.515
0	the US part of Global model (482 basins)	Global (ERA5-land)	Global (ERA5-land & HydroATLAS)	0.473
1	Regional Model (retrain on 482 basins)	Global (ERA5-land)	Global (ERA5-land & HydroATLAS)	0.431
2	Regional Model (retrain on 482 basins)	Global (ERA5-land)	Local (CAMELS)	0.462
3	Regional Model (retrain on 482 basins)	Local (maurer)	Local (CAMELS)	0.749

As can be seen from Table 4.7, neither the local nor global basin attribute significantly improves the median NSE of the model by comparing step 0, 1 and 2, but the median NSE of the model is significantly increased (from 0.462 to 0.749) based on step 2 and 3 comparison when the model input changed from global to local. This shows that the quality of the meteorological forcing data from the Caravan is not high compared to the local Maurer product, at least for these 482 basins in the US.

From Table 4.7, it can be seen that local meteorological forcing can significantly improve the median NSE of the Transformer compared to using the global forcing. To find out which channel in the meteorological forcing in Caravan impacted the model performance most, some forcing channels in Caravan were replaced with Maurer data based on step 1 in Table 4.7, and only one channel was replaced at a time, keeping the remaining unchanged, and trained the Transformer model on the same 482 basins again. There are 5 channels in Maurer data, namely, daily cumulative precipitation, daily minimum air temperature, daily maximum air temperature, average short-wave radiation, and vapor pressure.

In the forcing data of Caravan, three similar channels can be found to correspond to the Maurer forcing, namely daily total precipitation, shortwave radiation, and surface pressure. After replacing these three data one by one, the results can be obtained as shown in Table 4.8.

Table 4.8: The results of replacing ERA5-land forcing channels.

Model	Forcing	Replaced Channel (in ERA5-land)	Replaced by (in Maurer)	median NSE
Regional Model (retrain on 482 basins)	Global	-	-	0.431
Regional Model (retrain on 482 basins)	Global	Surface pressure	vapor pressure	0.450
		Shortwave radiation	average short-wave radiation	0.433
		Daily total precipitation	daily cumulative precipitation	0.687

As can be seen in the summary Table 4.8, a great improvement in median NSE can be achieved by replacing the precipitation data from global to local (from 0.473 to 0.687), which shows that the quality of rainfall data is very important for hydrological modeling, and precipitation data of the Caravan contains the significant uncertainty for global rainfall-runoff modeling.

4.3. Discussion

The main objective of this study is to investigate the potential of the Transformer architecture in rain-runoff modeling. In this research, some regional rainfall-runoff models based on the Transformer architectures were trained using CAMELS (US) data, and a global rainfall-runoff model based on the Transformer architecture was trained using Caravan data. All models were evaluated using various hydrological signatures and metrics. In regional rainfall-runoff modeling, the Transformer-based rainfall-runoff model was not able to fully replace LSTM, but the Reformer model performed well in snow-driven basins. In global modeling, the Transformer-based rainfall-runoff model did not perform well, mainly due to the uncertainty in the global Caravan data.

The existing results of using the Transformer architecture in hydrological research (Amanambu et al., 2022; Castangia et al., 2023; Yin et al., 2022) generally indicate that the Transformer-based hydrological models can outperform the LSTM benchmarks in discharge prediction such as floods and droughts forecasting, which is different from the results of this study. The main differences between these existing studies and this study are that the existing studies used the discharge (or water level) as input and performed long-term forecasting, while this study only predicted 1 day. The lack of using the target sequence (i.e. discharge) as input and short-term forecasting may be the reason why the Transformer in this study did not outperform LSTM.

There is still a work (Zeng et al., 2022) doubts about the effectiveness of using Transformer architecture for time series prediction due to Transformers do not preserve temporal order well. There is an experiment that was not mentioned in this study, in which different lengths (i.e. 10, 30, 50, 70, 100, 150, 270, 365, 548, and 730 days) of meteorological input were used to train the Transformer and LSTM using CAMELS (US) 531 basins. The results show that the two architectures have different responses to the various input length. As shown in Fig. 4.20, the CDF curves of LSTM almost overlap when the input length is 150, 270, 365, 548, or 730. However, for longer inputs such as 548 or 730, the CDF curves of the Transformer showed a slight shift to left, indicating a decline in the Transformer model performance (especially obvious when the input length is 730). These two different responses can be explained as LSTM is not able to effectively focus on earlier precipitation events in meteorological forcing, so longer meteorological input does not change the performance of the rainfall-runoff model based on LSTM, significantly. However, Transformer is able to attend very early precipitation events in meteorological input, but because it cannot preserve temporal information, longer input introduces more noise, causing the performance of the Transformer-based rainfall-runoff model to decline.

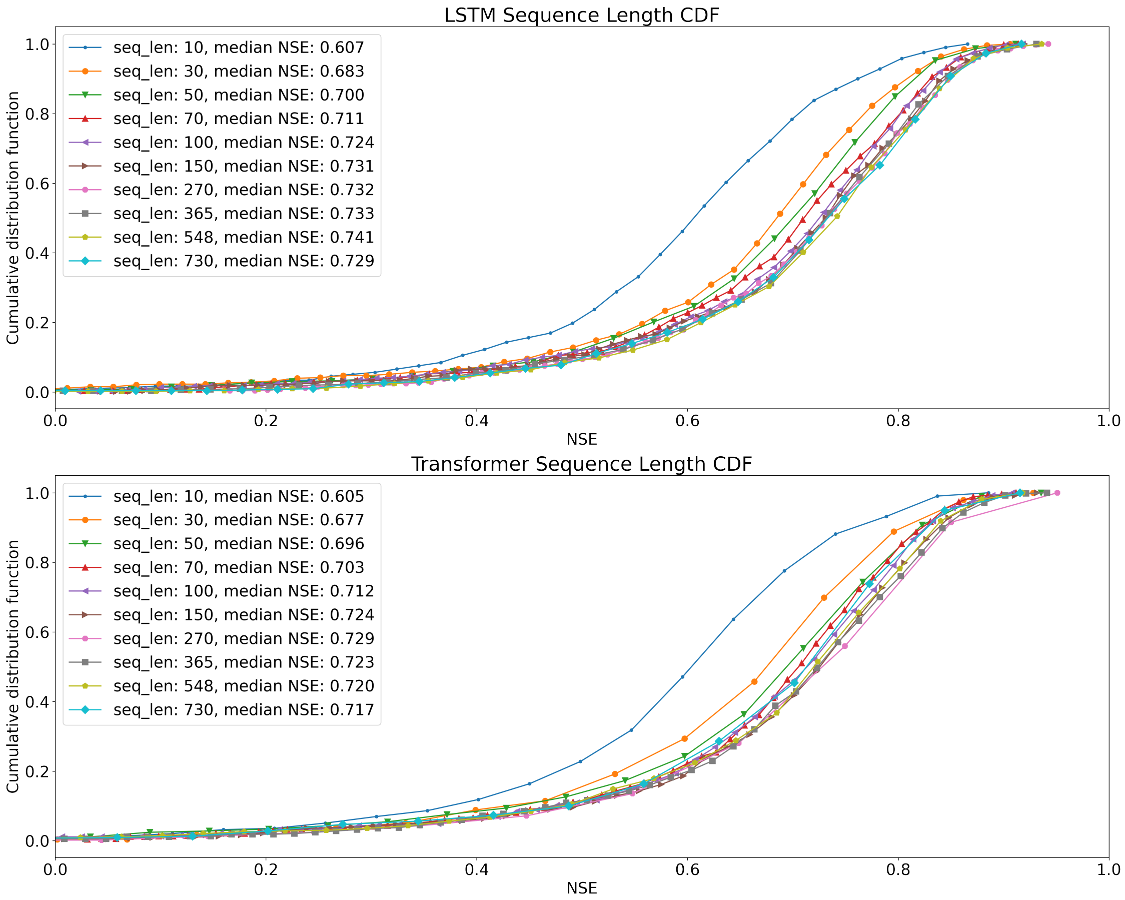


Figure 4.20: The effect of input length on the NSE CDF, Transformer's NSE CDF curve shifts left when really long forcing was fed, which is different from LSTM.

Additionally, there are still some limitations in this study, such as model parameter set selection and positional encoding methods.

Model selection In each experiment, each model was trained for 100 epochs, resulting in one hundred sets of parameters (weights) obtained. Then, the ”best” set of parameters was selected using the validation data based on the median NSE. This method ensures that the model converges well but it should be noted that the ”best” parameter set does not necessarily indicate superior performance compared to the other 99 sets. The distribution (variance) of NSE for the model should also be taken into account. Additionally, a higher NSE could indicate that the model excels at predicting peak flow, but may perform poorly in predicting low flow.

Timestamp positional encoding In this study, the timestamp positional encoding method was used, which included [Day-Of-Week, Day-Of-Month, Day-Of-Year] information. However, [Day-Of-Week] information has no obvious significance for hydrological modeling and may introduce more noise. Similarly, [Day-Of-Year] information may also have a negative impact on global-scale rainfall-runoff modeling because the seasons in the northern and southern hemispheres are opposite. Therefore, although many Transformer models designed specifically for time series prediction use this positional encoding method, it does not have clear applicability in hydrological modeling.

Data uncertainty In this study, the uncertainty of CAMELS (US) and Caravan data were not evaluated prior to training the model, which may have led to inaccurate predictions from the model after training.

5 Conclusion and future work

This chapter will draw conclusions by answering the two sub-questions of this study and provide recommendations for using the Transformer architecture for modeling rainfall-runoff modeling, while also suggesting some future work.

5.1. Conclusion and recommendations

Can Transformer architectures outperform LSTM in rainfall-runoff?

In the regional rainfall-runoff modeling experiment, based on metrics or hydrological signatures such as NSE and FDC, no Transformer architecture was found to significantly outperform LSTM, but all Transformer architectures took much longer time to train than the LSTM benchmark. In snow-driven basins, the Reformer architecture was found to be superior to LSTM in simulating peak and low flow, and relative variability in flows, and this still holds true for retraining LSTM and Reformer only on the snow-driven basins.

Therefore, the Reformer architecture may be an option for simulating rainfall-runoff in snow-driven basins. However, it should be noted that this conclusion was only drawn from 55 snow-driven basins in the US.

Can Transformer be a global rainfall-runoff model based on the Caravan data set?

No, according to the results of the second experiment, it appears that the Caravan data quality is not sufficient to build a global rainfall-runoff model. Whether training on the entire Caravan dataset or just its US portion, the model was unable to achieve good performance. The main barrier to global rainfall-runoff modeling appears to be the uncertainty in precipitation data.

Therefore, as stated above, the Transformer architecture cannot replace the LSTM architecture for rainfall-runoff simulation, but when simulating rainfall-runoff in a snow-driven basin, particularly when focusing on the peak and low flow, the Reformer architecture is worth considering and trying. Additionally, it is not feasible to establish a global rain-runoff model based on the Caravan data set, as it contains uncertainty for modeling, especially in the precipitation data. Besides, it is still unclear whether the Transformer can become a global rain-runoff model and learn a diverse range of rain-runoff behaviors.

5.2. Future work

The Reformer architecture has shown good results in snow-driven basins. Both LSTM and Transformer architectures-based hydrological models' performance can be explained by LSTM's cell states (Kratzert et al., 2018) and attention scores from different heads (Castangia et al., 2023), respectively. Therefore, some attempts can be made to interpret the Reformer's good performance in snow-driven basins.

As discussed in the discussion section, the Transformer architecture seems to have difficulty in preserving temporal information well (Zeng et al., 2022), which is detrimental for time series forecasting. This issue may also be present in Transformer-based rainfall-runoff modeling, and if this issue could be mitigated, the Transformer architecture may have the potential to outperform LSTM in tasks such as rainfall-runoff modeling.

It remains unsure if Transformer-based rainfall-runoff models would improve with larger training data sizes due to failure in global modeling. One of the main obstacles to global modeling is uncertainty in precipitation data. However, other global forcing data, such as Tropical Rainfall Measuring Mission (TRMM) (Kummerow et al., 1998), may be utilized as a global precipitation source to train a global model. An alternative approach to consider would be to train the Transformer using multiple CAMELS datasets, such as a combination of CAMELS (US) (Addor et al., 2017) and CAMELS-GB (Coxon et al., 2020). Some experimental results (Kratzert, Klotz, Shalev, et al., 2019; Lees et al., 2021) have shown that the LSTM architecture can achieve median NSE values of over 0.7 and 0.8 on these two datasets, respectively.

One of the important starting points of this study is to address the PUB problem, but the Transformer-based rainfall-runoff model in this study has not yet been tested in ungauged basins, and this is one of the possible future works.

References

Addor, N., Do, H. X., Alvarez-Garreton, C., Coxon, G., Fowler, K., & Mendoza, P. A. (2020). Large-sample hydrology: Recent progress, guidelines for new datasets and grand challenges. *Hydrological Sciences Journal*, *65*(5), 712–725.

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.

Amanambu, A. C., Mossa, J., & Chen, Y.-H. (2022). Hydrological drought forecasting using a deep transformer model. *Water*, *14*(22), 3611.

Arsenault, R., Brissette, F., Martel, J.-L., Troin, M., Lévesque, G., Davidson-Chaput, J., Gonzalez, M. C., Ameli, A., & Poulin, A. (2020). A comprehensive, multisource database for hydrometeorological modeling of 14,425 north american watersheds. *Scientific Data*, *7*(1), 1–12.

Babovic, V., & Keijzer, M. (2000). Forecasting of river discharges in the presence of chaos and noise. In *Flood issues in contemporary water management* (pp. 405–419). Springer.

Bardossy, A., Bogardi, I., & Duckstein, L. (1990). Fuzzy regression in hydrology. *Water Resources Research*, *26*(7), 1497–1508.

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer.

Blöschl, G., Bierkens, M. F., Chambel, A., Cudennec, C., Destouni, G., Fiori, A., Kirchner, J. W., McDonnell, J. J., Savenije, H. H., Sivapalan, M., et al. (2019). Twenty-three unsolved problems in hydrology (uph)–a community perspective. *Hydrological sciences journal*, *64*(10), 1141–1158.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Brath, A., Montanari, A., & Toth, E. (2002). Neural networks and non-parametric methods for improving real-time flood forecasting through conceptual hydrological models. *Hydrology and Earth System Sciences*, *6*(4), 627–639.

Bray, M., & Han, D. (2004). Identification of support vector machines for runoff modelling. *Journal of Hydroinformatics*, *6*(4), 265–280.

Budyko, M. I. (1963). Evaporation under natural conditions=.

Calzone, O. (2022). An intuitive explanation of lstm.

Castangia, M., Grajales, L. M. M., Aliberti, A., Rossi, C., Macii, A., Macii, E., & Patti, E. (2023). Transformer neural networks for interpretable flood forecasting. *Environmental Modelling & Software*, *160*, 105581.

Chagas, V. B., Chaffe, P. L., Addor, N., Fan, F. M., Fleischmann, A. S., Paiva, R. C., & Siqueira, V. A. (2020). Camels-br: Hydrometeorological time series and landscape attributes for 897 catchments in brazil. *Earth System Science Data*, *12*(3), 2075–2096.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Coxon, G., Addor, N., Bloomfield, J. P., Freer, J., Fry, M., Hannaford, J., Howden, N. J., Lane, R., Lewis, M., Robinson, E. L., et al. (2020). Camels-gb: Hydrometeorological time series and landscape attributes for 671 catchments in great britain. *Earth System Science Data*, *12*(4), 2459–2483.

Daniell, T. (1991). Neural networks. applications in hydrology and water resources engineering. *National Conference Publication- Institute of Engineers. Australia*.

Dertat, A. (2017). Applied deep learning - part 1: Artificial neural networks.

(DHPC), D. H. P. C. C. (2022). DelftBlue Supercomputer (Phase 1).

Edith, D. M., et al. (2021). The development of the planet impacted by climate change. *Annals-Economy Series*, *5*, 78–85.

Efstratiadis, A., & Koutsoyiannis, D. (2010). One decade of multi-objective calibration approaches in hydrological modelling: A review. *Hydrological Sciences Journal–Journal Des Sciences Hydrologiques*, *55*(1), 58–78.

Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2012). Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1915–1929.

Fowler, K. J., Acharya, S. C., Addor, N., Chou, C., & Peel, M. C. (2021). Camels-aus: Hydrometeorological time series and landscape attributes for 222 catchments in australia. *Earth System Science Data*, *13*(8), 3847–3867.

Frame, J. M., Kratzert, F., Klotz, D., Gauch, M., Shelev, G., Gilon, O., Qualls, L. M., Gupta, H. V., & Nearing, G. S. (2022). Deep learning rainfall–runoff predictions of extreme events. *Hydrology and Earth System Sciences*, *26*(13), 3377–3392.

Frame, J. M., Kratzert, F., Raney, A., Rahman, M., Salas, F. R., & Nearing, G. S. (2021). Post-processing the national water model with long short-term memory networks for streamflow predictions and model diagnostics. *JAWRA Journal of the American Water Resources Association*, *57*(6), 885–905.

Gauch, M., Kratzert, F., Klotz, D., Nearing, G., Lin, J., & Hochreiter, S. (2021). Rainfall–runoff prediction at multiple timescales with a single long short-term memory network. *Hydrology and Earth System Sciences*, *25*(4), 2045–2062.

Gomez, A. N., Ren, M., Urtasun, R., & Grosse, R. B. (2017). The reversible residual network: Backpropagation without storing activations. *Advances in neural information processing systems*, *30*.

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.

Gupta, H. V., Sorooshian, S., & Yapo, P. O. (1998). Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resources Research*, *34*(4), 751–763.

Halff, A. H., Halff, H. M., & Azmoodeh, M. (1993). Predicting runoff from rainfall using neural networks. *Engineering hydrology*, 760–765.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Vol. 2). Springer.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, *29*(6), 82–97.

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al. (2001). Gradient flow in recurrent nets: The difficulty of learning long-term dependencies.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Khandelwal, U., He, H., Qi, P., & Jurafsky, D. (2018). Sharp nearby, fuzzy far away: How neural language models use context. *arXiv preprint arXiv:1805.04623*.

Kitaev, N., Kaiser, Ł., & Levskaya, A. (2020). Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

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.

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

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.

Kratzert, F., Nearing, G., Addor, N., Erickson, T., Gauch, M., Gilon, O., Gudmundsson, L., Hassidim, A., Klotz, D., Nevo, S., et al. (2022). Caravan-a global community dataset for large-sample hydrology.

Kummerow, C., Barnes, W., Kozu, T., Shiue, J., & Simpson, J. (1998). The tropical rainfall measuring mission (trmm) sensor package. *Journal of atmospheric and oceanic technology*, *15*(3), 809–817.

Lee, H.-y. (2019). Transformer.

Lees, T., Buechel, M., Anderson, B., Slater, L., Reece, S., Coxon, G., & Dadson, S. J. (2021). Benchmarking data-driven rainfall–runoff models in great britain: A comparison of long short-term memory (lstm)-based models with four lumped conceptual models. *Hydrology and Earth System Sciences*, *25*(10), 5517–5534.

Linke, S., Lehner, B., Ouellet Dallaire, C., Ariwi, J., Grill, G., Anand, M., Beames, P., Burchard-Levine, V., Maxwell, S., Moidu, H., et al. (2019). Global hydro-environmental sub-basin and river reach characteristics at high spatial resolution. *Scientific data*, *6*(1), 1–15.

Liu, C., Liu, D., & Mu, L. (2022). Improved transformer model for enhanced monthly streamflow predictions of the yangtze river. *IEEE Access*.

Ma, K., Feng, D., Lawson, K., Tsai, W.-P., Liang, C., Huang, X., Sharma, A., & Shen, C. (2021). Transferring hydrologic data across continents–leveraging data-rich regions to improve hydrologic prediction in data-sparse regions. *Water Resources Research*, *57*(5), e2020WR028600.

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.

McMillan, H. K., Westerberg, I. K., & Krueger, T. (2018). Hydrological data uncertainty and its implications. *Wiley Interdisciplinary Reviews: Water*, *5*(6), e1319.

Muñoz-Sabater, J., Dutra, E., Agustí-Panareda, A., Albergel, C., Arduini, G., Balsamo, G., Boussetta, S., Choulga, M., Harrigan, S., Hersbach, H., et al. (2021). Era5-land: A state-of-the-art global reanalysis dataset for land applications. *Earth System Science Data*, *13*(9), 4349–4383.

Nash, J. E., & Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part i—a discussion of principles. *Journal of hydrology*, *10*(3), 282–290.

Newman, A., Clark, M., Sampson, K., Wood, A., Hay, L., Bock, A., Viger, R., Blodgett, D., Brekke, L., Arnold, J., 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.

Pinos, J., & Quesada-Román, A. (2021). Flood risk-related research trends in latin america and the caribbean. *Water*, *14*(1), 10.

Popel, M., & Bojar, O. (2018). Training tips for the transformer model. *arXiv preprint arXiv:1804.00247*.

Razavi, T., & Coulibaly, P. (2013). Streamflow prediction in ungauged basins: Review of regionalization methods. *Journal of hydrologic engineering*, *18*(8), 958–975.

Ropelewski, C. F., & Halpert, M. S. (1986). North american precipitation and temperature patterns associated with the el niño/southern oscillation (enso). *Monthly Weather Review*, *114*(12), 2352–2362.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85–117.

Searcy, J. K. (1959). *Flow-duration curves*. US Government Printing Office.

Shrestha, D. L., & Solomatine, D. P. (2008). Data-driven approaches for estimating uncertainty in rainfall-runoff modelling. *International Journal of River Basin Management*, *6*(2), 109–122.

Solomatine, D. P., & Dulal, K. N. (2003). Model trees as an alternative to neural networks in rainfall—runoff modelling. *Hydrological Sciences Journal*, *48*(3), 399–411.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, *27*.

USGS. (2018). How streamflow is measured.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wagener, T., & Wheater, H. S. (2006). Parameter estimation and regionalization for continuous rainfall-runoff models including uncertainty. *Journal of hydrology*, *320*(1-2), 132–154.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

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*.

Wikipedia contributors. (2022). Wilcoxon signed-rank test — Wikipedia, the free encyclopedia [[Online; accessed 20-January-2023]]. https://en.wikipedia.org/w/index.php?title=Wilcoxon_signed-rank_test&oldid=1129790782

Wilcoxon, F. (1992). Individual comparisons by ranking methods. In *Breakthroughs in statistics* (pp. 196–202). Springer.

Xia, Y., Mitchell, K., Ek, M., Sheffield, J., Cosgrove, B., Wood, E., Luo, L., Alonge, C., Wei, H., Meng, J., et al. (2012). Continental-scale water and energy flux analysis and validation for the north american land data assimilation system project phase 2 (nldas-2): 1. intercomparison and application of model products. *Journal of Geophysical Research: Atmospheres*, *117*(D3).

Xiang, Z., Yan, J., & Demir, I. (2020). A rainfall-runoff model with lstm-based sequence-to-sequence learning. *Water resources research*, *56*(1), e2019WR025326.

Xu, X., Liu, W., Scanlon, B. R., Zhang, L., & Pan, M. (2013). Local and global factors controlling water-energy balances within the budyko framework. *Geophysical Research Letters*, *40*(23), 6123–6129.

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).

Yin, H., Guo, Z., Zhang, X., Chen, J., & Zhang, Y. (2022). Rr-former: Rainfall-runoff modeling based on transformer. *Journal of Hydrology*, *609*, 127781.

Yin, H., Zhang, X., Wang, F., Zhang, Y., Xia, R., & Jin, J. (2021). Rainfall-runoff modeling using lstm-based multi-state-vector sequence-to-sequence model. *Journal of Hydrology*, *598*, 126378.

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of physics: Conference series*, *1168*(2), 022022.

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2022). Are transformers effective for time series forecasting? *arXiv preprint arXiv:2205.13504*.

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.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. *arXiv preprint arXiv:2201.12740*.

A

A Training Configuration Example

```
additional_feature_files: None
batch_size: 256
cache_validation_data: true
checkpoint_path: None
clip_gradient_norm: 1
clip_targets_to_zero:
- QObs(mm/d)
commit_hash: a4c284b
data_dir: /scratch/kangminmao/data/CAMELS_US
dataset: camels_us
device: cuda:0
dynamic_inputs:
- prcp(mm/day)
- srad(W/m2)
- tmax(C)
- tmin(C)
- vp(Pa)
dynamics_embedding:
  type: fc
  hiddens:
  - 30
  - 20
  - 64
  activation: tanh
  dropout: 0.0
epochs: 100
evolving_attributes:
experiment_name: Transformer_seq365
forcings:
- maurer_extended
head: regression
hidden_size: 256
img_log_dir: /scratch/kangminmao/exp/runs/Transformer_seq365_1810_000943/img_log
initial_forget_bias: 3
learning_rate:
  0: 0.001
  10: 0.0005
  20: 0.0001
log_interval: 5
log_n_figures: 2
log_tensorboard: true
loss: NSE
metrics:
```

```
- NSE
- MSE
- RMSE
- KGE
- Alpha-NSE
- Pearson-r
- Beta-KGE
- Beta-NSE
- FHV
- FMS
- FLV
- Peak-Timing
model: transformer
num_workers: 16
number_of_basins: 531
ode_method: euler
ode_num_unfolds: 4
ode_random_freq_lower_bound: 6D
optimizer: Adam
output_activation: linear
output_dropout: 0.4
package_version: 1.3.0
per_basin_test_periods_file: None
per_basin_train_periods_file: None
per_basin_validation_periods_file: None
predict_last_n: 1
regularization:
run_dir: /scratch/kangminmao/exp/runs/Transformer_seq365_1810_000943
save_train_data: false
save_validation_results: true
save_weights_every: 1
seed: 502918
seq_length: 365
shared_mtslstm: false
static_attributes:
- p_mean
- pet_mean
- aridity
- p_seasonality
- high_prec_freq
- high_prec_dur
- low_prec_freq
- low_prec_dur
- elev_mean
- slope_mean
- area_gages2
- lai_max
- lai_diff
- gvf_max
- gvf_diff
- soil_depth_pelletier
- soil_depth_statsgo
- soil_porosity
- soil_conductivity
- max_water_content
- sand_frac
- silt_frac
- clay_frac
- geol_permeability
statics_embedding:
  type: fc
  hiddens:
  - 30
  - 20
  - 64
  activation: tanh
  dropout: 0.0
target_noise_std:
target_variables:
- QObs(mm/d)
test_basin_file: /scratch/kangminmao/exp/basin/531_basin_list.txt
test_end_date: 30/09/1999
test_start_date: 01/10/1989
train_basin_file: /scratch/kangminmao/exp/basin/531_basin_list.txt
train_data_file: None
train_dir: /scratch/kangminmao/exp/runs/Transformer_seq365_1810_000943/train_data
train_end_date: 30/09/2008
train_start_date: 01/10/1999
transfer_mtslstm_states:
  h: identity
  c: identity
transformer_dim_feedforward: 32
transformer_dropout: 0
transformer_nheads: 4
transformer_nlayers: 4
transformer_positional_dropout: 0.0
transformer_positional_encoding_type: sum
use_basin_id_encoding: false
validate_every: 1
validate_n_random_basins: 3000
validation_basin_file: /scratch/kangminmao/exp/basin/531_basin_list.txt
validation_end_date: 30/09/1989
validation_start_date: 01/10/1980
```

B

Regional modeling static basins attributes

Table B.1: Table of catchment attributes used in the regional modeling

p_mean	Mean daily precipitation.
pet_mean	Mean daily potential evapotranspiration.
aridity	Ratio of mean PET to mean precipitation.
p_seasonality	Seasonality and timing of precipitation. Estimated by representing annual precipitation and temperature as sine waves. Positive (negative) values indicate precipitation peaks during the summer (winter). Values of approx. 0 indicate uniform precipitation throughout the year.
frac_snow_daily	Fraction of precipitation falling on days with temperatures below 0 °C.
high_prec_freq	Frequency of high-precipitation days (≥5 times mean daily precipitation).
high_prec_dur	Average duration of high-precipitation events (number of consecutive days with ≥5 times mean daily precipitation).
low_prec_freq	Frequency of dry days (<1 mm d−1).
low_prec_dur	Average duration of dry periods (number of consecutive days with precipitation <1 mm d−1).
elev_mean	Catchment mean elevation.
slope_mean	Catchment mean slope.
area_gages2	Catchment area.
forest_frac	Forest fraction.
lai_max	Maximum monthly mean of leaf area index.
lai_diff	Difference between the max. and min. mean of the leaf area index.
gvf_max	Maximum monthly mean of green vegetation fraction.
gvf_diff	Difference between the maximum and minimum monthly mean of the green vegetation fraction.
soil_depth_pelletier	Depth to bedrock (maximum 50 m).
soil_depth_statsgo	Soil depth (maximum 1.5 m).
soil_porosity	Volumetric porosity.
soil_conductivity	Saturated hydraulic conductivity.
max_water_content	Maximum water content of the soil.
sand_frac	Fraction of sand in the soil.
silt_frac	Fraction of silt in the soil.
clay_frac	Fraction of clay in the soil.
carb_rocks_frac	Fraction of the catchment area characterized as “Carbonate sedimentary rocks”.
geol_permeability	Surface permeability (log10).

Table B.1 lists the catchment attributes used by the LSTM benchmark in regional modeling. The Transformer-based models also used these attributes except:

- frac_snow_daily,
- forest_frac,
- and carb_rocks_frac,

because some of the 531 basins miss three attributes, but there are still 24 attributes that can be used for model training. One of our experiments: a length-270 model based on the same forcing as (Kratzert, Klotz, Shalev, et al., 2019) and the rest 24 attributes, can achieve the close performance as the benchmark (Kratzert, Klotz, Shalev, et al., 2019), which shows that missing these 3 attributes won't really affect the model performance, significantly.

C

LSTM & Transformer Global Models Results

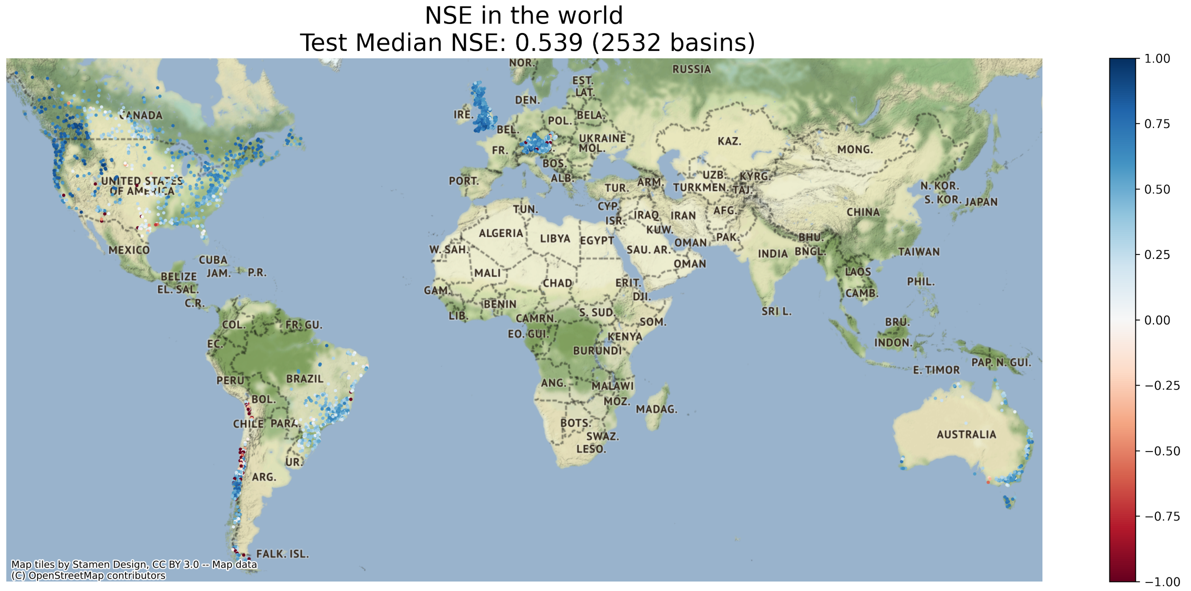


Figure C.1: Global LSTM-based rainfall-runoff model NSE, the color map is limited to [-1, 1] for the NSE differences for better visualization

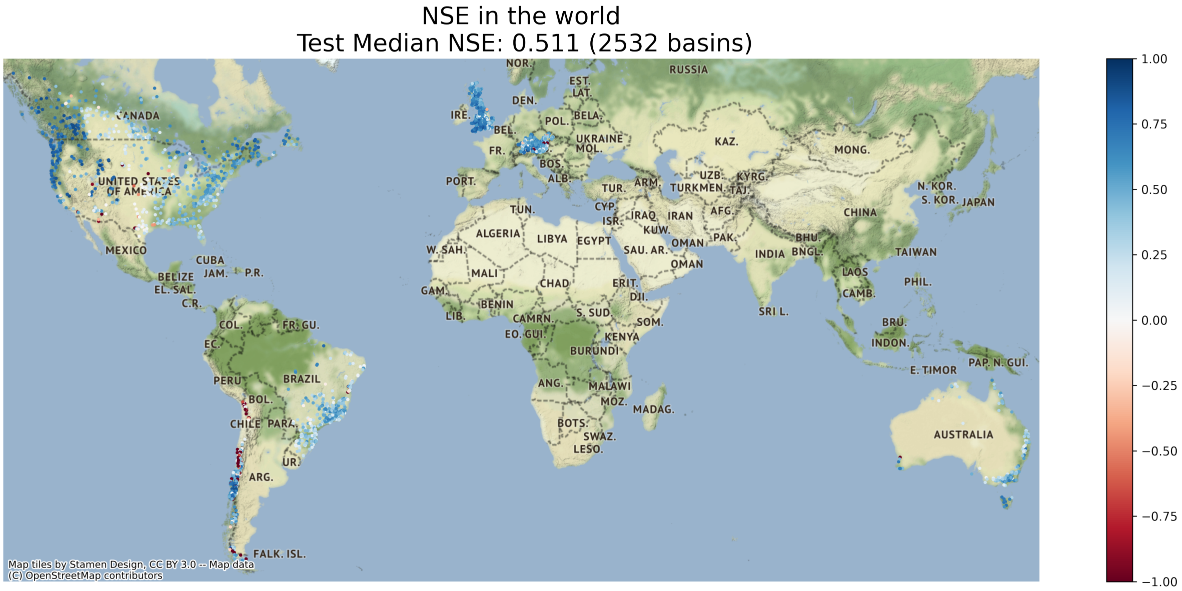

NSE in the world
Test Median NSE: 0.511 (2532 basins)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Map tiles by Stamen Design, CC BY 3.0 -- Map data
(C) OpenStreetMap contributors

Figure C.2: Global Transformer-based rainfall-runoff model NSE, the color map is limited to [-1, 1] for the NSE differences for better visualization

D

Transformer family regional modeling

Table D.1: Transformer family configuration

Model	Layer	Head	Embedding	Seed	Number of Trainable Parameters
Reformer	2	8	timestamp	483593	1157005
FEDformer	2	8	timestamp	710046	100796373
Linformer	2	16	sinusoidal	936305	669881
Transformer	4	4	timestamp	733480	402573
Informer	2	8	sinusoidal	192899	450573

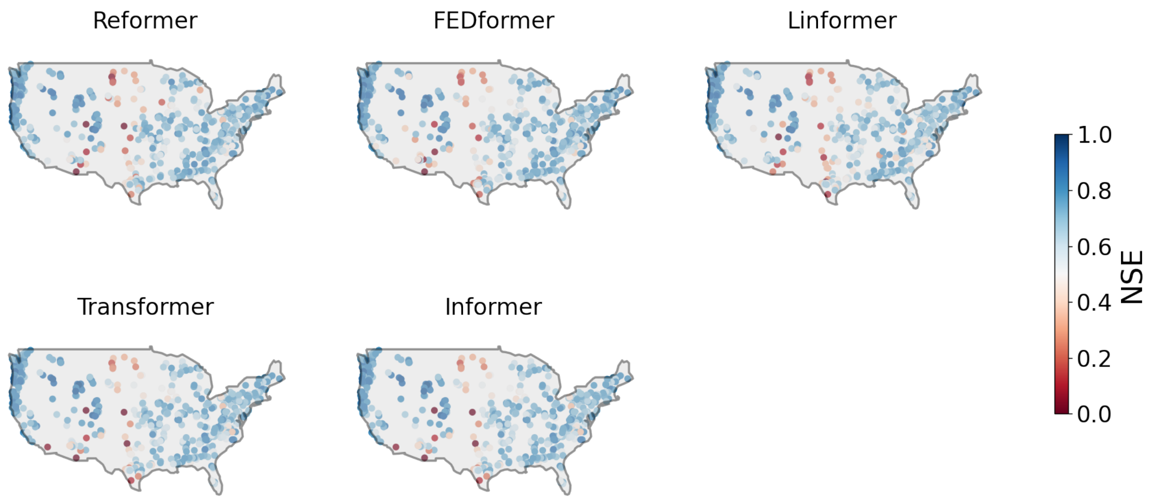


Figure D.1: Transformer family regional modeling NSE distribution, the color map is limited to [0, 1] for the NSE differences for better visualization

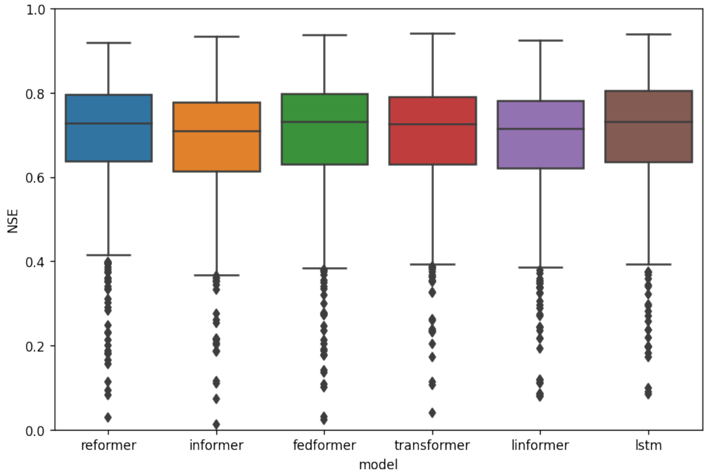


Figure D.2: Transformer family NSE boxplot, y-axis is limit to [0, 1] for a visualization

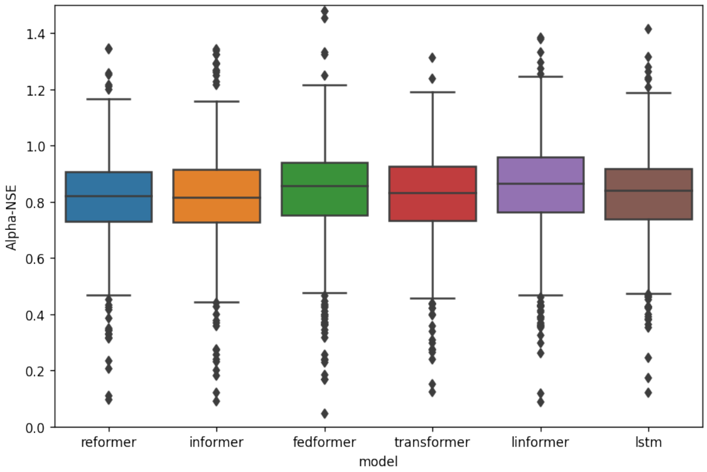


Figure D.3: Transformer family Alpha-NSE boxplot, y-axis is limit to [0, 1.5] for a visualization

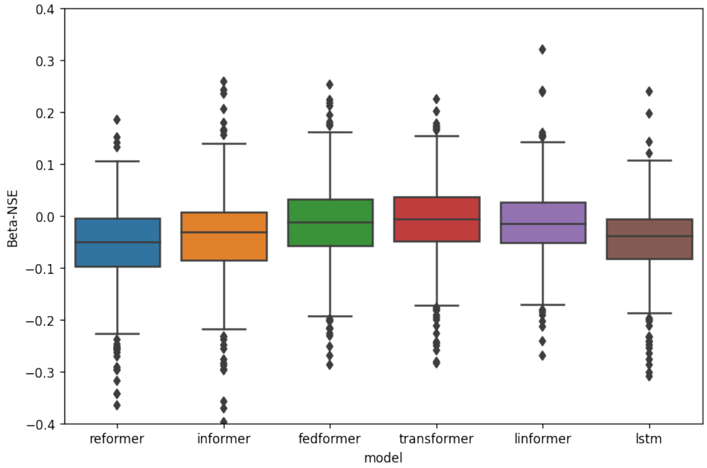


Figure D.4: Transformer family Beta-NSE boxplot, y-axis is limit to [-0.4, 0.4] for a visualization

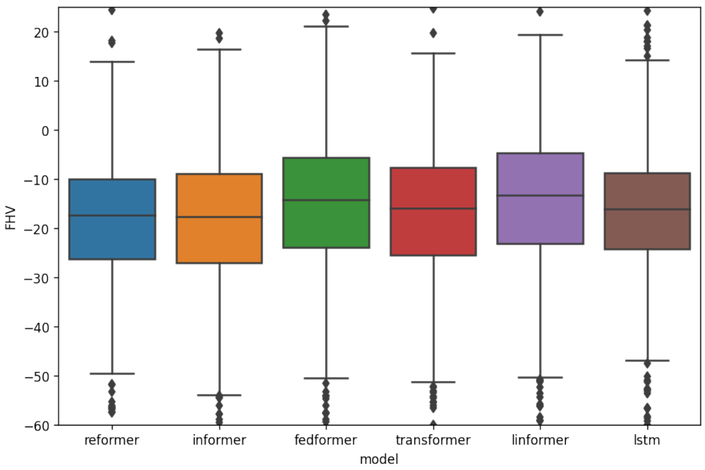


Figure D.5: Transformer family FHV boxplot, y-axis is limit to [-60, 25] for a visualization

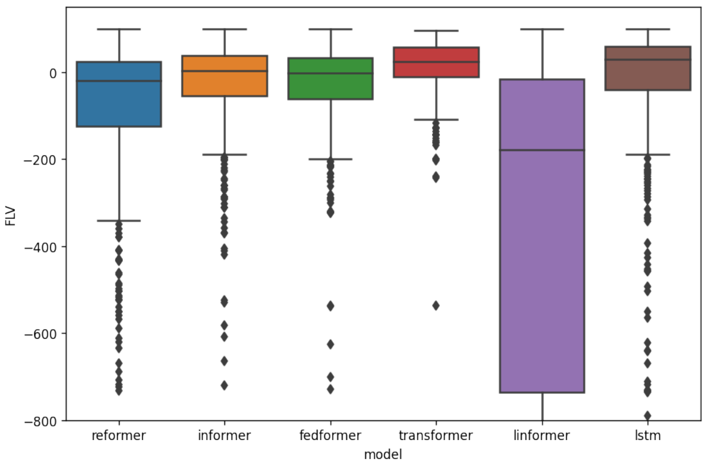


Figure D.6: Transformer family FLV boxplot, y-axis is limit to [-800, 150] for a visualization

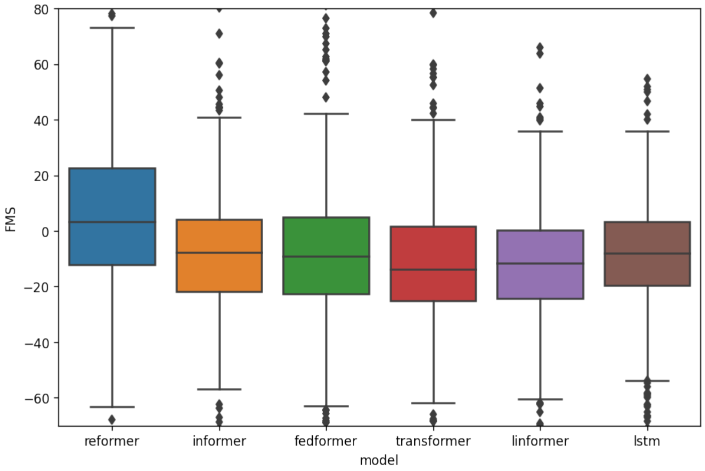


Figure D.7: Transformer family FMS boxplot, y-axis is limit to [-70, 80] for a visualization

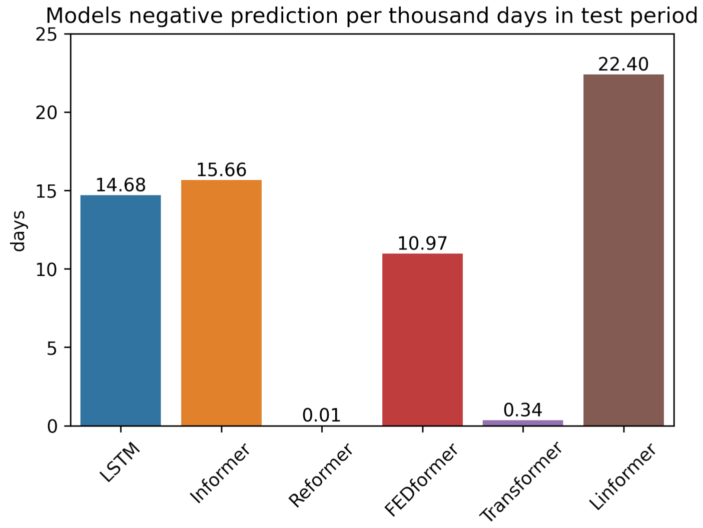


Figure D.8: Models negative prediction per thousand days in the test period.

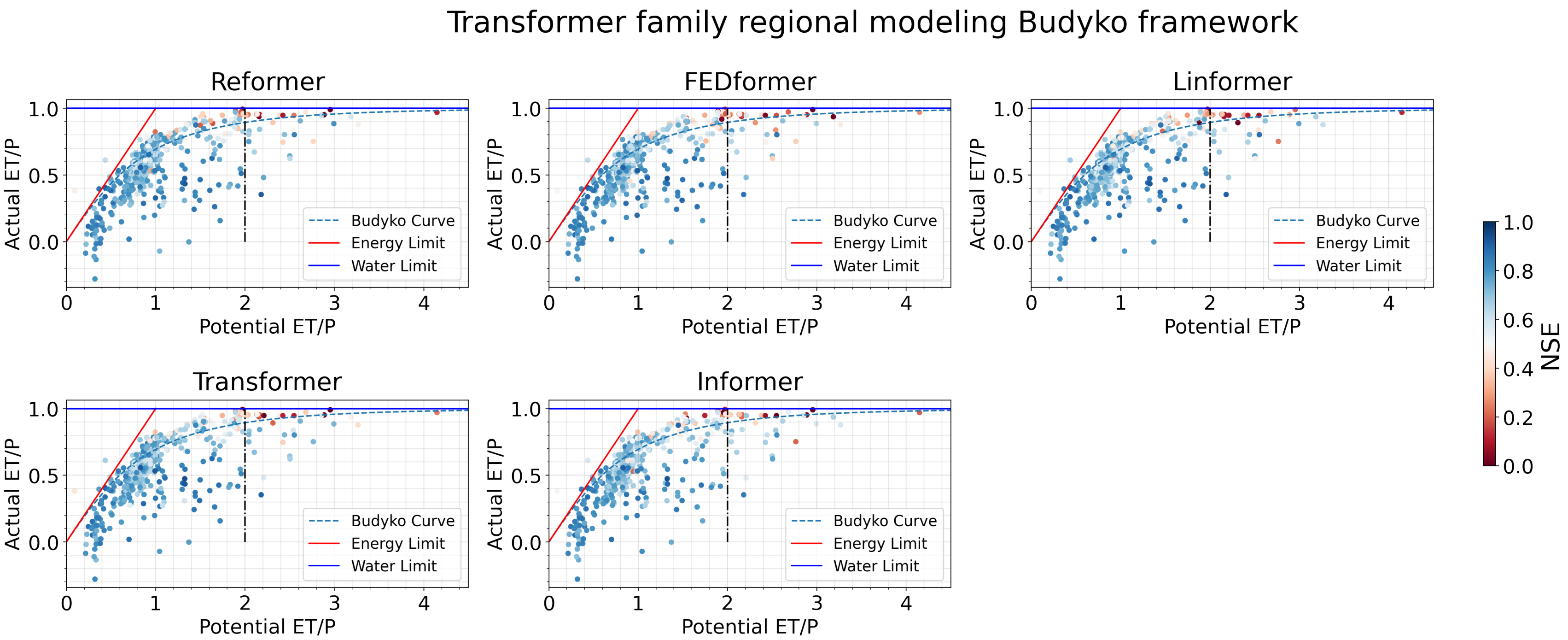


Figure D.9: Transformer family Budyko frame, the color maps are limited to [0, 1] for better visualization

E

Regional modeling hydrographs

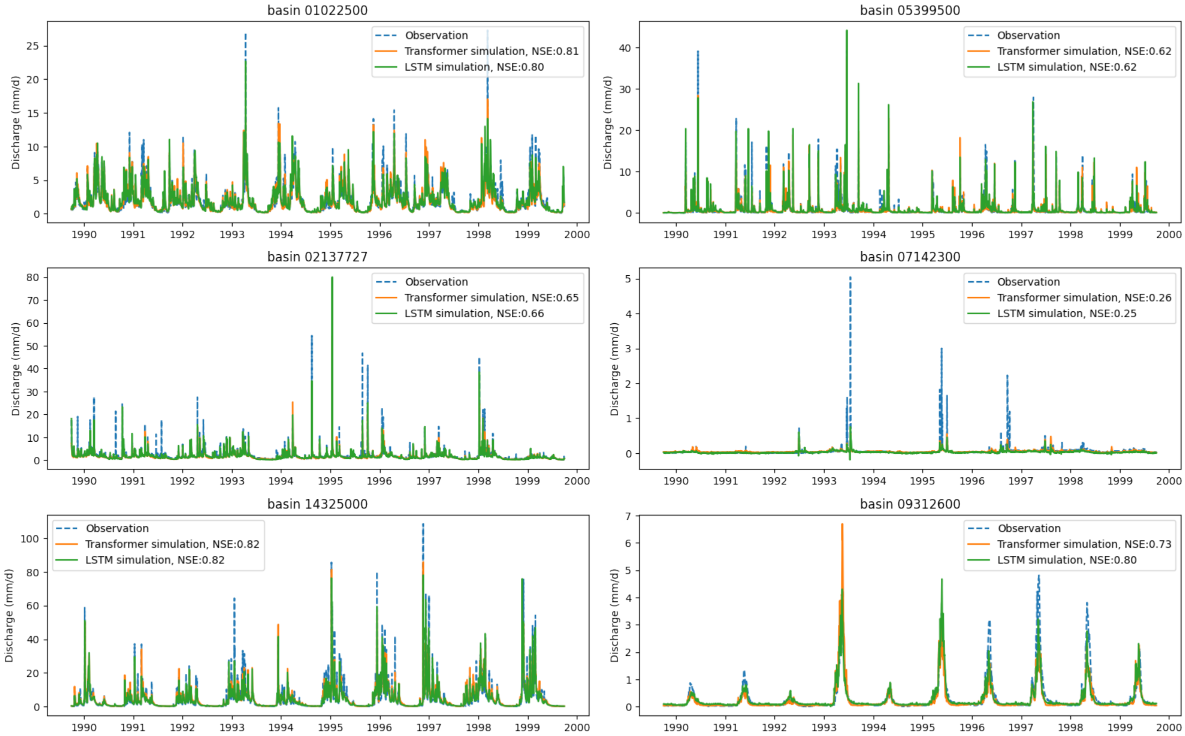


Figure E.1: Hydrographs

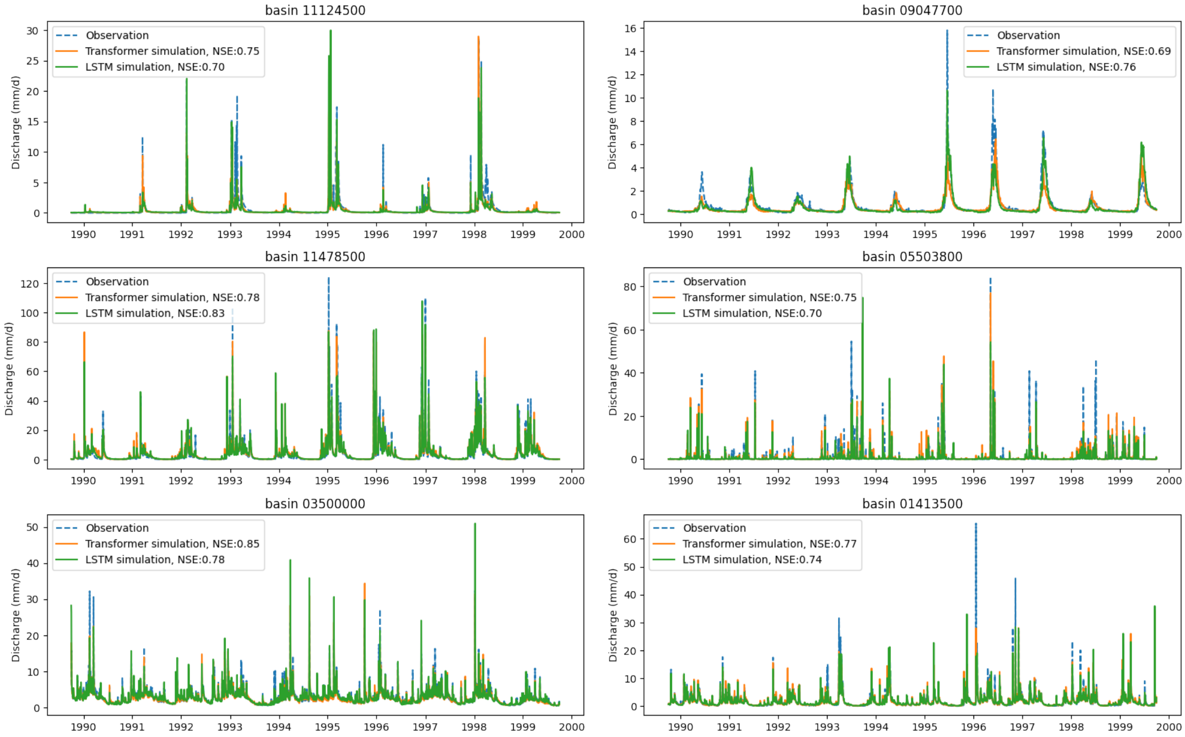


Figure E.2: Hydrographs

F

Positional encoding difference

The Transformers are thought the positional information is needed, but in our regional modeling, we can find that: the model performance will not decrease a lot when the positional encoding layer were dropped.

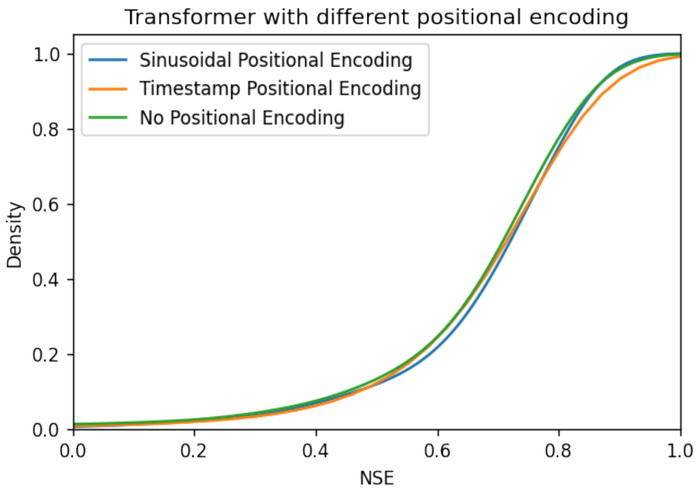


Figure F.1: Transformer with different (and no) positional encoding CDF

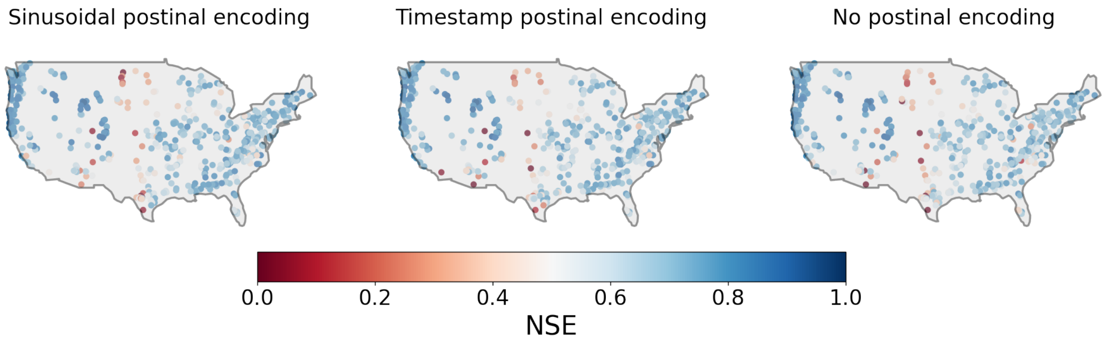


Figure F.2: Transformer with different NSE distribution

Table F.1: The median metrics of Transformer with different (and no) positional encoding

	Sinusoidal	Timestamp	No
NSE	0.723	0.725	0.715
Alpha-NSE	0.848	0.834	0.839
FHV	-14.982	-15.900	-15.628
FMS	-8.817	-13.742	-12.101
FLV	19.006	24.667	-30.635
Training time (min/epoch)	19.327	64.307	20.115